Vivante GCNanoUltraV Series V2.0.x Small Footprint Vector Graphics IP

Hardware Features

Document Revision 0.95
16 January 2026

This document is compatible with Vivante GCNanoUltraV hardware versions 2.0.x

Legal Notices

Preface

This document is the primary feature description for Vivante GCNanoUltraV Series vector graphics processing unit IP core.

Audience

This document was prepared for both hardware and software integrators who are familiar with system-on-a-chip (SoC) digital design and related support devices. Those who would benefit from this technical document are:

- Engineers and managers who are evaluating this core for use in a system
- Engineers who are designing this core into a system

Conventions Used in This Document

AHB	Advanced High-Performance Bus
APB	Advanced Peripheral Bus
AXI	Advanced eXtensible Interface
DDR	Double Data Rate
DMA	Dynamic Memory Access
FE	Graphics Pipeline Front End/Fetch Engine
GPU	Graphics Processing Unit
GUI	Graphical User Interfaces
HI	Host Interface
IM	Imaging Engine
IOT	Internet of Things
MC	Memory Controller
MCU	Micro Controller Unit
MPU	Micro Processor Unit
PE	Pixel Engine
PM	Power Management
RA	Rasterizer
RTL	Resistor Transistor Logic
SoC	System on Chip
TS	Tessellation Engine
VG	Vector Graphics

The word *assert* means to drive a signal true or active. Signals that are active LOW end in an "n."
Hexadecimal numbers are indicated by the prefix "0x" —for example, 0x32CF.
Binary numbers are indicated by the prefix "0b" —for example, 0b0011.0010.1100.1111
Code snippets are given in `Consolas` typeset.

Table of Contents

LEGAL NOTICES **2**
PREFACE **3**
TABLE OF CONTENTS **4**
LIST OF FIGURES **5**
LIST OF TABLES **5**
1 INTRODUCTION **6**
1.1 GCCORE Design Description 7
1.2 GCCORE API Support 8
1.3 Core Variants 9
2 GPU CORE MODULE FEATURES **10**
2.1 GPU Host and Memory Interface Features 10
2.2 GPU Power Management Features 10
2.3 GPU Command Processor Features 11
2.4 GCNanoUltraV Series Graphics Hardware Features 11
2.5 Format Support with GCNanoUltraV IP 12
2.6 Alignment Requirement with GCNanoUltraV IP 14
2.6.1 Image Buffer Alignment Requirement 14
2.6.2 Destination Buffer Alignment Requirement 16
3 ENCODE AND DECODE HARDWARE FEATURES **17**
3.1 DECNano Lossy Data Compression Overview 17
4 SOFTWARE API **18**
4.1 Vivante VGLite Graphics API 18
4.1.1 Reference 18
4.1.2 Organization of the VGLite Graphics API 18
DOCUMENT REVISION HISTORY **19**

List of Figures

Figure 1. Typical SoC with Vivante GCNanoUltraV IP 6
Figure 2. GCNanoUltraV Block Diagram with Vector Graphics Pipeline and AXI+AHB 7
Figure 3. GCNanoUltraV Block Diagram with Vector Graphics Pipeline with 3xAHB 7

List of Tables

Table 1. GCNanoUltraV API Support 8
Table 2. Typical RTL Key Features and Customization Summary 9
Table 3. Host and Memory Interface Features 10
Table 4. Power Management Features 10
Table 5. Command Processor Features 11
Table 6. Graphics Hardware Features 11
Table 7. Graphics Formats Available 12
Table 8. Image Buffer Alignment Summary 14
Table 9. Destination Buffer Alignment Summary 16
Table 10. Encode Decode Hardware Features 17
Table 11. Compression Ratios Per Compression Mode 17
Table 12. Blit and Draw Controls in VGLite API 18

1 Introduction

This document provides a summary of the functional feature set for the Vivante GCNanoUltraV Series small footprint vector graphics processing unit IP core (GCNanoUltraV GPU IP). The Vivante GCNanoUltraV Series IP defines a next generation high-performance/area UI graphics core designed for hardware acceleration of Vivante's proprietary VGLite™ Graphics API for vector graphics display on a variety of consumer devices.

Architectural enhancements bring the look and feel of consumer product GUI rendering and menu displays to embedded and Internet-of-Things (IoT) devices that need ultra-low power, minimal silicon area and zero DDR memory. The Vivante GCNanoUltraV Series cores has a simple, lightweight application programming interface (API) and software architecture that makes customization easy and minimizes memory footprint.

The Vivante GCNanoUltraV Series processor cores are designed specifically for MCU and MPU applications that need hardware accelerated UI displays and effects. Built upon Vivante's successful mass market proven ScalarMorphic™ architecture powering some of the leading smartphones, TVs, tablets, automobiles, and embedded devices, the embedded UI architecture of the GCNanoUltraV Series brings high-end graphics capabilities to MCU/MPU designs.

GCNanoUltraV Series is designed from the ground up to work within the limitations of high memory and compute resource constrained MCU/MPU solutions including embedded flash memory configurations. These features are absolutely essential for building efficient embedded systems targeting the next generation of displays to meet rising consumer expectations for graphics intensive man-machine interaction.

Our robust embedded UI graphics solution includes easy-to-use software, a VGLite Graphics API to enable the GPU acceleration through customized applications on Linux.

Vivante GCNanoUltraV Series IP is designed for easy integration onto the SoC, providing powerful graphics at low power consumption and the smallest of silicon footprints. The core is delivered as synthesizable RTL. It is technology independent and can be synthesized using a variety of libraries. Dynamic power consumption is minimized by extensive use of localized clock gating.

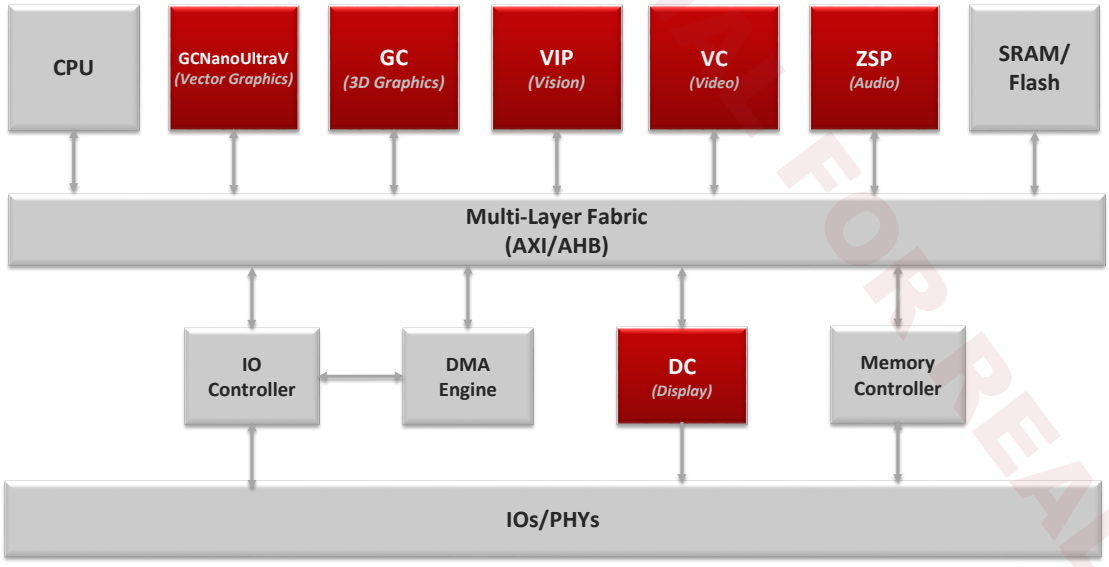

Figure 1. Typical SoC with Vivante GCNanoUltraV IP

1.1 GCCORE Design Description

The main functional blocks of the GCCORE are described here, and a block diagram is shown below.

Host Interface	Allows the GCCORE to communicate with external memory and the CPU through AXI and/or AHB/APB buses. In this block data crosses clock domain boundaries.
Memory Controller	Internal memory management unit that controls the block-to-host memory request interface.
Graphics Pipeline Front End	Inserts high level primitives and commands into the graphics pipeline.
Tessellation Engine	Transforms vertices and control points. Tessellates lines, quadratic and cubic Bezier curves.
Vector Graphics Engine	Rasterizer that converts primitives to pixels.
Imaging Engine	Paint and image generator that colors each pixel.
Pixel Engine	Renderer that combines different sources into the final pixel value.

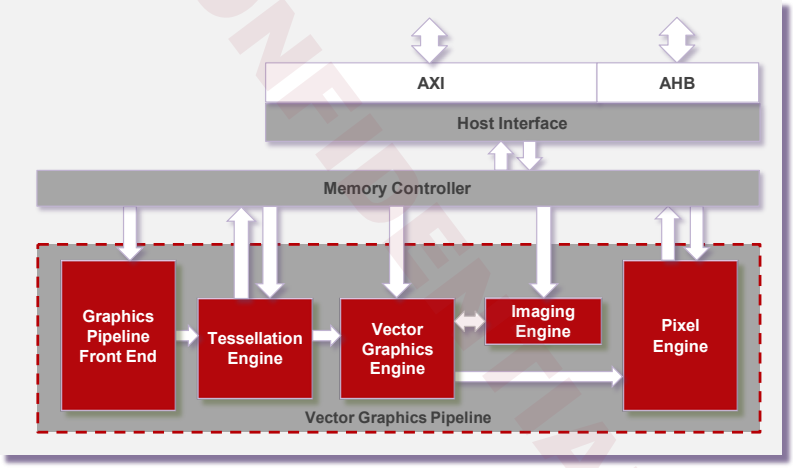

Figure 2. GCNanoUltraV Block Diagram with Vector Graphics Pipeline and AXI+AHB

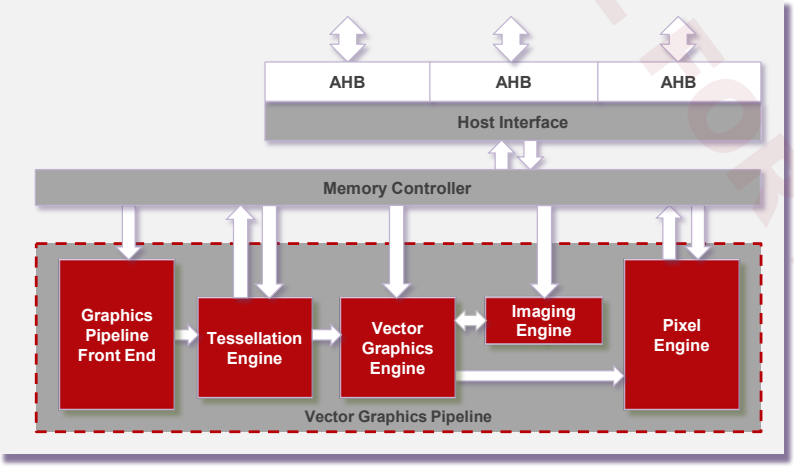

Figure 3. GCNanoUltraV Block Diagram with Vector Graphics Pipeline with 3xAHB

1.2 GCCORE API Support

The following table describes the API support available for the Vivante GCNanoUltraV Series small footprint vector graphics IP.

Table 1. GCNanoUltraV API Support

Feature	GPU Support
Primary API	Vivante's proprietary VGLite® Graphics API
Drivers	Vivante GCNanoUltraV Driver
Operating systems	VGLite Graphics API is platform independent, typical use is Embedded Linux

1.3 Core Variants

The GCNanoUltraV V2.0x Series Vector Graphics is designed in the following primary variant:

GCNanoUltraV This is the default design. GCNanoUltraV Series has a small area. Product revisions may vary, especially in their RGB and YUV format support levels.

Because of its target small footprint design, few options are configurable for this IP. The following table summarizes some key features and options. Typical product features which can be customized by Vivante prior to RTL delivery are listed in the table below.

Table 2. Typical RTL Key Features and Customization Summary

Feature	GCNanoUltraV Default	Alternate Options
Interfaces	1 64-bit AXI + 1 32-bit AHB	3 32-bit AHB
32-bit bus interface	AHB	APB option
Clock source	clk1x	
Image Source Read	YES	
Vivante VGLite API Support	YES	
Transformation Support	3x3 perspective and affine	
Linear Paint	YES	
INDEX1/2/4/8 Input Support	YES	
YUY2 Input Support	YES	
YUV Tiled Input Support	YES	option
ETC2 Compression	YES	option
DECNano Compression/Decompression	YES	option Compress on Output, Decompress Input and Output

For pixel format support, refer to Section 2.5.

This Vivante design is customizable, including customizations for RGB color formats and YUV formats. Please note that all variations presented in this document may not be immediately available “off the shelf” as Ready-IP. Check with your Vivante personnel for schedule and availability.

2 GPU Core Module Features

2.1 GPU Host and Memory Interface Features

Table 3. Host and Memory Interface Features

Feature	GPU Support
Interfaces	1 64-bit AXI + 1 32-bit AHB (default) or 3x32-bit AHB + FLEXA® interface. APB alternate option for AHB.
Code and data memory location restrictions	Unrestricted; arbitrary memory reads and writes
Physical address	32 bits
Read write request size support	Memory controller supports 8, 16, 32, 64 byte read and write requests
Resource locks with CPU	Semaphore lock
Clock domains	3 domains: • core clock (from clk1x), • register access AHB (from HCLK pin) or alternately APB (from PCLK pin), and • memory data access from AXI (ACLK pin) or alternately AHB (from ACLK pin)

Note: FLEXA interface is a set of hardware interfaces defined by VeriSilicon. To share and transfer data between hardware IPs by using low-level hardware and software mechanisms, FLEXA enables creating multi-IP subsystem solutions. If this feature is not enabled, the FLEXA input on the interface could be set to 0, and the FLEXA* output could be left floating.

FLEXA mode does not support rotation, translation, scaling, filter, gradient paint, 3-planars image input, DECNano, orientation, or image input. Output frame sizes are inconsistent (if only the output buffer is working in FLEXA mode, there is no above restriction except for DECNano and orientation). When the input buffer enables two consumers, data from two IP addresses is not supported.

2.2 GPU Power Management Features

Table 4. Power Management Features

Feature	GPU Support
Low power CMOS technology compatible	Yes
Automatic clock gating of flip flops and rams	Yes
Global clock gating of unused macro blocks	Yes
Active (ON), Idle, Standby and Sleep (OFF) Programmable Power Modes	Yes

2.3 GPU Command Processor Features

Table 5. Command Processor Features

Feature	GPU Support
Command list structure	Call memory buffer
GPU register access	AHB access to selected GPU registers
GPU-CPU synchronization	Synchronization occurs via Interrupt queues.
Command OPCODES for GCNanoUltraV	The Command set is different than that of most other GCCORES: 00 END 04 DATA 01 SEMAPHORE 06 CALL 02 STALL 07 RETURN 03 STATE 08 NOP Additional Command OPCODES are provided for Path Data control.

2.4 GCNanoUltraV Series Graphics Hardware Features

Table 6. Graphics Hardware Features

Feature	Description
Area	Feature set optimized for small footprint
Blending	8 Porter Duff blending modes
Rotation	To any angle
Fill rules	Odd/even and non-zero
Paint paths	Linear and curved paint paths
Compression	ETC2
Coordinate systems and transformations	Image drawing uses a 3x3 perspective transformation matrix
Paints	Solid and linear paint
Image filters and interpolation	Point, Linear and Bi-linear blends
Pixel formats for image interpolation	Pixel formats for image interpolation. *See following Section 2.5 for detail.*
Perspective Transformation	Perspective transformation

2.5 Format Support with GCNanoUltraV IP

Table 7. Graphics Formats Available

Feature	GPU Support
Image formats The graphics engine supports these formats for source image and destination render targets. Notes: • Your design variant may be customized and may not include support for all possible formats.	With hardware V 2.0.x

Format	Bit Depth	Supported for Source IMAGE	Supported for Destination
INDEX1	1	Yes	
INDEX1_TILED	1	Yes	
INDEX2	2	Yes	
INDEX2_TILED	2	Yes	
INDEX4	4	Yes	
INDEX4_TILED	4	Yes	
INDEX8	8	Yes	
INDEX8_TILED	8	Yes	
A1	1	Yes	Yes
A1_TILED	1	Yes	Yes
A2	2	Yes	Yes
A2_TILED	2	Yes	Yes
A4	4	Yes	Yes
A4_TILED	4	Yes	Yes
A8	8	Yes	Yes
A8_TILED	8	Yes	Yes
L4	4	Yes	Yes
L4_TILED	4	Yes	Yes
L8	8	Yes	Yes
L8_TILED	8	Yes	Yes
A8L8	16	Yes	Yes
A8L8_TILED	16	Yes	Yes
ARGB2222	8	Yes	Yes
ARGB2222_TILED	8	Yes	Yes
RGB565	16	Yes	Yes
RGB565_TILED	16	Yes	Yes
ARGB1555	16	Yes	Yes
ARGB1555_TILED	16	Yes	Yes
ARGB4444	16	Yes	Yes
ARGB4444_TILED	16	Yes	Yes
ARGB8888	32	Yes	Yes
ARGB8888_TILED	32	Yes	Yes
XRGB8888	32	Yes	Yes
XRGB8888_TILED	32	Yes	Yes
ARGB8565	24	Yes	Yes
ARGB8565_TILED	24	Yes	Yes
RGB888	24	Yes	Yes
RGB888_TILED	24	Yes	Yes
ARGB8565_PLANAR	24	Yes	Yes
ARGB8565_PLANAR_TILED	24	Yes	Yes
YUY2/UYVY	16	Yes	
YUY2/UYVY_TILED	16	Yes	
NV12	12	Yes	
NV12_TILED	12	Yes	
YV12	12	Yes	

<table>
<tr><th>Feature</th><th>GPU Support</th></tr>
<tr><td></td><td>
<table>
<tr><td>YV12_TILED</td><td>12</td><td>Yes</td><td></td></tr>
<tr><td>NV16</td><td>16</td><td>Yes</td><td></td></tr>
<tr><td>YV16</td><td>16</td><td>Yes</td><td></td></tr>
<tr><td>NV24</td><td>24</td><td>Yes</td><td></td></tr>
<tr><td>NV24_TILED</td><td>24</td><td>Yes</td><td></td></tr>
<tr><td>YV24</td><td>24</td><td>Yes</td><td></td></tr>
<tr><td>YV24_TILED</td><td>24</td><td>Yes</td><td></td></tr>
<tr><td>ETC2_ARGB8888</td><td>8</td><td>Yes</td><td></td></tr>
<tr><td>ETC2_RGB888</td><td>4</td><td>Yes</td><td></td></tr>
</table>
</td></tr>
<tr><td>Color component swizzle support</td><td>Supported color component swizzles are:
SWIZZLE_ABGR, SWIZZLE_ARGB, SWIZZLE_BGRA, SWIZZLE_RGBA</td></tr>
<tr><td>DECNano format compression support (optional)</td><td>
<table>
<tr><th>Bit Depth</th><th>Format</th><th>Supported for Source</th><th>Supported for Destination</th></tr>
<tr><td>32</td><td>A8R8G8B8</td><td>Optional</td><td>Optional</td></tr>
<tr><td>32</td><td>X8R8G8B8</td><td>Optional</td><td>Optional</td></tr>
</table>
Note: DECNano can be bypassed by software configuration.
</td></tr>
<tr><td>ETC2 format compression support (optional)</td><td>
<table>
<tr><th>Bit Depth</th><th>Format</th><th>Supported for Source</th><th>Supported for Destination</th></tr>
<tr><td>32</td><td>A8R8G8B8</td><td>Yes</td><td>NO</td></tr>
<tr><td>24</td><td>R8G8B8</td><td>Yes</td><td>NO</td></tr>
</table>
</td></tr>
</table>

2.6 Alignment Requirement with GCNanoUltraV IP

2.6.1 Image Buffer Alignment Requirement

Table 8. Image Buffer Alignment Summary

Image Format	Bits per pixel	Source Tile Mode	Start Address Alignment Requirement in Bytes	Stride Alignment Requirement in Bytes	Buffer Height Alignment Requirement	AXI Burst Length in Bytes (Read)	Supported for Source Image	Supported for Destination
VG_LITE_INDEX1	1	Linear	8B	1B	1	8*	Yes	
	1	Tile	8B	1B	4	8*	Yes	
VG_LITE_INDEX2	2	Linear	8B	1B	1	8*	Yes	
	2	Tile	8B	1B	4	8*	Yes	
VG_LITE_INDEX4	4	Linear	8B	1B	1	8	Yes	
	4	Tile	8B	2B	4	8	Yes	
VG_LITE_INDEX8	8	Linear	8B	1B	1	16	Yes	
	8	Tile	8B	4B	4	16	Yes	
VG_LITE_A4	4	Linear	8B	1B	1	8	Yes	
	4	Tile	8B	2B	4	8	Yes	
VG_LITE_A8	8	Linear	8B	1B	1	16	Yes	Yes
	8	Tile	8B	4B	4	16	Yes	Yes
VG_LITE_L8	8	Linear	8B	1B	1	16	Yes	Yes
	8	Tile	8B	4B	4	16	Yes	Yes
VG_LITE_ARGB2222	8	Linear	8B	1B	1	16	Yes	Yes
	8	Tile	8B	4B	4	16	Yes	Yes
VG_LITE_RGB565	16	Linear	8B	2B	1	32	Yes	Yes
	16	Tile	8B	8B	4	32	Yes	Yes
VG_LITE_ARGB1555	16	Linear	8B	2B	1	32	Yes	Yes
	16	Tile	8B	8B	4	32	Yes	Yes
VG_LITE_ARGB4444	16	Linear	8B	2B	1	32	Yes	Yes
	16	Tile	8B	8B	4	32	Yes	Yes
VG_LITE_ARGB8888	32	Linear	8B	4B	1	64	Yes	Yes
	32	Tile	8B	16B	4	64	Yes	Yes
VG_LITE_XRGB8888	32	Linear	8B	4B	1	64	Yes	Yes
	32	Tile	8B	16B	4	64	Yes	Yes
VG_LITE_ARGB8565	24	Linear	8B	3B*	1	48	Yes	Yes
	24	Tile	8B	12B*	4	48	Yes	Yes
VG_LITE_RGB888	24	Linear	8B	3B*	1	48	Yes	Yes
	24	Tile	8B	12B*	4	48	Yes	Yes
ARGB8565_PLANAR	24	Linear	A: 8B RGB: 8B	A: 1B RGB: 2B	1	A: 16 RGB: 32	NO (option)	NO (option)
	24	Tile	A: 8B RGB: 8B	A: 4B RGB: 8B	4	A: 16 RGB: 32	NO (option)	NO (option)
VG_LITE_YUY2/UYVY	16	Linear	8B	4B	1	32	Yes	
	16	Tile	8B	8B	4	32	Yes	
VG_LITE_NV12	12	Linear	Y: 8B UV: 8B	Y: 2B UV: 2B	1	Y: 16 UV: 16	No (option)	
	12	Tile	Y: 8B UV: 8B	Y: 8B UV: 8B	4	Y: 16 UV: 8	No (option)	

Image Format	Bits per pixel	Source Tile Mode	Start Address Alignment Requirement in Bytes	Stride Alignment Requirement in Bytes	Buffer Height Alignment Requirement	AXI Burst Length in Bytes (Read)	Supported for Source Image	Supported for Destination
VG_LITE_YV12	12	Linear	Y: 8B U: 8B V: 8B	Y: 2B U: 1B V: 1B	1	Y: 16 U: 8 V: 8	No (option)	
	12	Tile	Y: 8B U: 8B V: 8B	Y: 8B U: 4B V: 4B	4	Y: 16 U: 8* V: 8*	No (option)	
VG_LITE_NV16	16	Linear	Y: 8B UV: 8B	Y: 2B UV: 2B	1	Y: 16 UV: 16	No (option)	
VG_LITE_YV16	16	Linear	Y: 8B U: 8B V: 8B	Y: 2B U: 1B V: 1B	1	Y: 16 U: 8 V: 8	No (option)	
VG_LITE_NV24	24	Linear	Y: 8B UV: 8B	Y: 1B UV: 2B	1	Y: 16 UV: 32	No (option)	
	24	Tile	Y: 8B UV: 8B	Y: 4B UV: 8B	4	Y: 16 UV: 32	No (option)	
VG_LITE_YV24	24	Linear	Y: 8B U: 8B V: 8B	Y: 1B U: 1B V: 1B	1	Y: 16 U: 16 V: 16	No (option)	
	24	Tile	Y: 8B U: 8B V: 8B	Y: 4B U: 4B V: 4B	4	Y: 16 U: 16 V: 16	No (option)	
VG_LITE_ETC2_ARGB8888	8	Tile	8B	4B	4	16	Yes	

*Note:

1. Values in the Stride Alignment Requirement in Bytes column reflect the alignment requirements of the data in memory. The stride of ARGB8565/RGB888 is seen as 4Byte per pixel when configuring hardware.
2. AXI Burst Size in Bytes (Read) is at least 8B. For index1 format, only 2B of the 8B read data is available. For index2 format, only 4B of the 8B read data is available. For YV12U and V plane, only 4B of the 8B read data is available.

2.6.2 Destination Buffer Alignment Requirement

Table 9. Destination Buffer Alignment Summary

Target Format	Bits per pixel	Target Tile Mode	Start Address Alignment Requirement in Bytes	Stride Alignment Requirement in Bytes	Buffer Height Alignment Requirement	AXI Burst Length in Bytes (Read)	AXI Burst Length in Bytes (Write)	Supported for Source Image	Supported for Destination
VG_LITE_A8	8	Linear	4B	1B	1	64	64*	Yes	Yes
		Tile	4B	4B	4	64	64*	Yes	Yes
VG_LITE_L8	8	Linear	4B	1B	1	64	64*	Yes	Yes
		Tile	4B	4B	4	64	64*	Yes	Yes
VG_LITE_ARGB2222	8	Linear	4B	1B	1	64	64*	Yes	Yes
		Tile	4B	4B	4	64	64*	Yes	Yes
VG_LITE_RGB565	16	Linear	4B	2B	1	64	64*	Yes	Yes
		Tile	4B	8B	4	64	64*	Yes	Yes
VG_LITE_ARGB1555	16	Linear	4B	2B	1	64	64*	Yes	Yes
		Tile	4B	8B	4	64	64*	Yes	Yes
VG_LITE_ARGB4444	16	Linear	4B	2B	1	64	64*	Yes	Yes
		Tile	4B	8B	4	64	64*	Yes	Yes
VG_LITE_ARGB8888	32	Linear	4B	4B	1	64	64*	Yes	Yes
		Tile	4B	16B	4	64	64*	Yes	Yes
VG_LITE_XRGB8888	32	Linear	4B	4B	1	64	64*	Yes	Yes
		Tile	4B	16B	4	64	64*	Yes	Yes
VG_LITE_ARGB8565	24	Linear	64B	3B*	1	48	48*	Yes	Yes
		Tile	64B	12B*	4	48	48*	Yes	Yes
VG_LITE_RGB888	24	Linear	64B	3B*	1	48	48*	Yes	Yes
		Tile	64B	12B*	4	48	48*	Yes	Yes
ARGB8565_PLANAR	24	Linear	A: 64B RGB: 64B	A: 1B RGB: 2B	1	A:16 RGB: 32	A:16* RGB:32*	NO (option)	NO (option)
		Tile	A: 64B RGB: 64B	A: 4B RGB: 8B	4	A:16 RGB: 32	A:16* RGB:32*	NO (option)	NO (option)

*Note:

1. The values in the table reflect the alignment requirements of pixel data in memory. The stride of ARGB8888/ARGB8565 is seen as 4 Bytes per pixel when configuring hardware.
2. For tile mode, buffer stride is still the byte size of a row of pixels instead of 4 rows of pixels.
3. For the PE clear function, the clear size must align to 48 Bytes for RGB888 or ARGB8565 format.
4. For the PE clear function with DECNano enabled, the clear size must align to 48 Bytes for RGB888 and must align to 64 Bytes for ARGB8888 or XRGB8888.
5. If DECNano function is enabled for the buffer, the target buffer start address must align to 64 Bytes.
6. If DECNano function is enabled for the buffer, the total buffer size must align to a 64 Byte*compression rate for ARGB8888 or XRGB8888 format and must align to a 48 Byte*compression rate for RGB888 format.
7. AXI Burst Size in Byte (Write) is always in a unit of single cacheline, which is 64B. But PE has a mechanism where it can write back only dirty bytes among 64B to avoid dummy write. It supports 8, 16, 24, 32, 40, 48, 56, 64 Bytes. So, in the table, 64* means at most 64 Bytes.
8. AXI Burst Size in Bytes (Write) for 24bpp and ARGB8565PLANAR formats is shown as 48*. This is because when 24bpp is in use and the write back unit is a single cacheline, the available data is at most 48 Bytes.
9. For ARGB8565PLANAR format, the write requests include write Alpha request and write RGB request. For Alpha request it is marked as 16* (at most 16 Bytes); for RGB request it is marked as 32* (at most 32 Bytes).
10. PE clear write burst size is 48B for 24bpp, 64B for other formats including ARGB8565PLANAR.

3 Encode and Decode Hardware Features

Key features of the DEC compression unit are shown in the following table.

Table 10. Encode Decode Hardware Features

Feature	DEC Support
Input and Output Pixel Formats	See Data Formats Section
Compression Scheme	Lossy Compression and decompression
Compression Modes	3: No sample; Horizontal only; Horizontal and vertical
Compression De-compression Ratio	1.6 to 3.0 dependent on color format and compression mode
Pixel Data Organization Support (in pixels)	Tile (4x4) and Raster Scan (16x1)
User controls*	Compression is controlled by the GPU
Compression Units	64 Bytes
Burst Size	4 Bytes
By-pass Scheme	Uncompressed data travels through the GPU pipeline directly bypassing the DEC compression/decompression logic then travels across the AXI interface through standard memory-to-the memory display path. DECNano bypass can be configured via software.
Codec Pixel Data Rate	1 pixel per cycle encode/decode for 32bpp
Bandwidth Reduction	2-3X with lossy compression
Maximum Concurrent Channels	DECNano: 1 read path and 1 write path DECNanoE: 1 write path DECNanoD: 1 read path
Tile Status Support	No
Error Notification	Yes

*Note: The DECNano does not have any user-controllable features. Functions such as stream support, stream identification, flush and outstanding requests should be implemented outside the DECNano if necessary.

3.1 DECNano Lossy Data Compression Overview

The DECNano Series supports the following formats, tile modes and Compression modes.

- Formats are ARGB8 and XRGB8 and RGB8, specified in signal **format**. A/X is in the upper bits and B in the lower bits.
- Tile mode is either linear 16x1 pixels or tiled 4x4 pixels per tile. Each tile occupies 64 Bytes or 48Bytes.
- Four compression modes are supported:
  - None
  - Horizontal sample
  - Horizontal and vertical sample. Note this mode is not supported for linear scanlines.
  - High quality mode

Table 11. Compression Ratios Per Compression Mode

Color Subsample Compression Mode	ARGB	XRGB	RGB
None	2:1	2.6:1	2:1
Horizontal Only (raster or tile)	2.6:1	4:1	3:1
Horizontal and Vertical (tile only)	4:1	N/A	N/A
High quality mode (raster or tile)	1.6:1	2:1	1.5:1

4 Software API

The Vivante VGLite® Graphics API is designed specifically for use with Vivante GCNanoUltraV Series vector graphics processing unit IP. It provides mechanisms for hardware accelerated two-dimensional vector and raster graphics, thereby allowing the user to implement customized applications and drivers for their hardware accelerated graphical user interface.

4.1 Vivante VGLite Graphics API

Vivante's platform independent VGLite Graphics API (Application Programming Interface) is designed to support menu driven user interfaces optimized for a system's overall resource requirements. Its goal is to provide maximum performance, while keeping the memory footprint to a minimum. The typical environment is Embedded Linux. The Vivante VGLite Graphics API allows fine granularity in memory usage and is appropriate for use in cases where only one of the available rendering classes is used.

4.1.1 Reference

The document **Vivante Programming: VGLite Vector Graphics API** is the current reference document for the VGLite API. It provides a set of functions for menu UIs with Vivante vector graphics GCNanoUltraV Series cores.

4.1.2 Organization of the VGLite Graphics API

The VGLite API is partitioned to provide controls for three main areas of functionality:

- **Initialization** for hardware and software initialization
- **Blit** for raster rendering
- **Draw** for draw operations

Hardware supported Blit and/or Draw capabilities available for software control through the API include:

Table 12. Blit and Draw Controls in VGLite API

Capability	GCNanoUltraV GPU with VGLite API Support
Fill	Non-zero or Even-Odd
Path coordinate formats	S8, S16, S32, FP32
HW Anti-alias levels	Low, Medium, High
Blending	SrcOver, DstOver, SrcIn, DstIn, Screen, Multiply, Additive, Subtractive
Data Formats	Linear: INDEX1/2/4/8, ARGB2/4/8, XRGB8, RGB8, A8RGB565, A4/8, L8, YUY2 and BRGA/X variants. Tiled YUV formats if supported by hardware. (Refer to VGLite API documentation for full list of formats)
Path definitions	Bounding box, path quality, coordinate format, buffer pointer, path length, path data

Document Revision History

This section describes top level differences in the versions of this document.

Document Revision	Doc Date	Compatible Hardware	Description
0.95	2026-01-16	GCNanoUltraV	Section 1.3, Table 2 Typical RTL Key Features and Customization Summary: updated default GCNanoUltraV default support for YUV Tiles Input Support, ETC2 Compression and DECNano Compression/Decompression. Section 2.5, updated Table 7 Graphics Formats Available.
0.94	2026-01-13	GCNanoUltraV	Section 2.1, Added FLEXA restriction.
0.93	2025-12-18	GCNanoUltraV	Section 2.1, Table 2 Host and Memory Interface Features: added FLEXA interface to Interface feature and note.
0.92	2025-10-28	GCNanoUltraV	Section 3.1 DECNano Lossy Data Compression Overview: added RGB8 to list of supported formats; added 48 Bytes to supported tile sizes; added High quality mode (raster or tile) to list of supported compression modes; added RGB column to Table 11 Compression Ratios Per Compression Mode.
0.91	2025-02-12	GCNanoUltraV	Added Section 2.6 Alignment Requirement with GCNanoUltraV IP.
0.90	2024-11-06	GCNanoUltraV	Change to single watermark. Minor formatting enhancements. Section 2.4, Table 6 Graphics Hardware Features: added Perspective transformation feature. Section 2.5, Table 7 Graphics Formats Available: updated Image Formats table. Added Section 2.6 Alignment Requirement with GCNanoUltraV IP.
0.89	2023-02-24	GCNanoUltraV	Update document template. Section 1.3: Add DECNano option Section 2.5: Add DECNano format support. Section 3: Inserted for DECNano
0.88	2022-09-02	GCNanoUltraV	Update document template. Miscellaneous format and text refinements. Section 3.1.2 Table 8: add YUY2.
0.87	2022-05-30	GCNanoUltraV	Update document template. Miscellaneous refinements. Section 2.5 Table 7: Remove SWIZZLE_UV, SWIZZLE_VU.
0.86	2022-04-12	GCNanoUltraV	Section 2.4, Table 6: Add row for image filters
0.85	2022-02-02	GCNanoUltraV	Legal Notices, General: Update branding layout to include VeriSilicon. Miscellaneous format refinements.

Document Revision	Doc Date	Compatible Hardware	Description
0.84	2022-01-11	GCNanoUltraV	Section 1.1: Add block diagram for AXI+AHB. Section 1.3: Refine Interface options. Update INDEX as supported by default. Remove ARGB2 (also default, not a separate option). Section 2.1and various: Update available interface options. Section 2.3: remove WAIT. Section 2.4: update format description to reference next table. Section 2.5 Table 7: remove DI references and update support levels.
0.83	2021-12-10	GCNanoUltraV	Section 1.3: Add format options for INDEX and ARGB2. Section 2.3: remove WAIT. Section 2.4: update format description to reference next table. Section 2.5 Table 7: add optional formats. Add note for YUV targets.
0.82	2021-11-12	GCNanoUltraV	Section 1.3 and various: Remove variant F. Section 1.3 Table 1. Add more key features.
0.81	2021-10-01	GCNanoUltraV	Section 2.5 Table 7 and Section 3.1.2 Table 8: Add tiled formats.
0.80	2021-07-13	GCNanoUltraV	Initial Edition