



# GoForce 3D 3D/2D Graphics Core

## Advance Information for GDC Attendees

### 3D ENGINE

- Geometry transform engine (floating point and fixed point)
- 16-bpp (RGB565) color with 16-bit Z (high quality dithering)
- 40-bit color pipeline with signed non-integer color (over bright)
- 8 surfaces (color, Z, texture 1-6)
- Programmable pixel shader
- Fog, perspective correction
- Mip-mapping
- Bilinear/trilinear filtered texturing
- 4/8-bit palettized texture, 16-bit (1555ARGB, 565RGB, 4444ARGB)
- Multi-texture support (up to 6 simultaneous textures)
- Super-sampled antialiasing
- Compressed texture support (DXT)

### 2D ENGINE

- 8-bpp (palettized), 16-bpp and 32-bpp
- Rectangle draw and BitBLT with 3-operand raster operation (ROP)
- All-angle Bresenham line drawing with sub-pixel resolution and ROP
- Mono (text) to color expansion
- Mono pattern or mono source transparency
- Source or destination color transparency
- Clipping
- Drawing synchronization with LCD display controller / 3D unit

### POWER MANAGEMENT

- Automatic power-down of unused pipelines (nPower)
- Normal, standby, and sleep modes
- Architectural-level power management
- Gated-clock power-management
- RTL/gate-level power-management
- Circuit-level power-management
- Low-voltage operation

### STANDARDS

- OpenGL ES 1.0 and 1.1 supported
- Mobile D3D Supported
- JSR184 Supported

### INTRODUCTION

The NVIDIA GoForce 3D is an ultra low power 2D/3D graphics core designed from ground up to bring truly realistic 3D graphics and blazing gaming performance to handsets and other mobile devices. Stunning, lifelike rich media effects can now be experienced anytime anywhere on handheld devices. The GoForce 3Dcore is based on traditional OpenGL architecture and all of the geometry and pixel processing are executed in hardware. GoForce 3D is compliant with OpenGL ES 1.0 and future versions of OpenGL ES as well as Mobile D3D API from Microsoft.

The GoForce 3D core incorporates NVIDIA's patent-pending nPower technology, which saves precious battery time, enabling over ten times the energy efficiency as compared to equivalent software-based solutions.

### 3D ENGINE

The 3D engine is based on a traditional OpenGL architecture. Vertex data is written into the vertex cache and textures are stored in display/texture memory. Vertex data is either IEEE floating point, or 16.16 fixed data. Texture data can be in the format of 4bit/8bit palettized textures, 16 bit 1555 or 565 color, or compressed textures.

The transform unit takes vertex information (X, Y, Z, W plus color and/or texture attributes), applies a user defined transform to calculate screen space coordinates for each triangle which is then sent to the rasterizer unit to draw the given triangle.

The rasterizer calculates new color and texture attributes (stored in pixel packets) as it walks every pixel of the given triangle.

The data fetch block reads both the existing pixel data or texel data from frame buffer memory.

The shader unit performs most of the 3D algorithm work such as texture combine, fog, alpha blend, etc.

### 2D ENGINE

The 2D engine consists of a BitBLT engine and a line engine. The BitBLT engine is a rectangle drawing engine with three-operand (Pattern, Source, Destination) raster operation (ROP), clipping and transparency support. The line engine supports is an all angle Bresenham line drawing engine with clipping and transparency support.

