

# 3D graphics core targets mobile devices

By Ashley Stevens

Solutions Marketing Manager

ARM Ltd

E-mail: Ashley.Stevens@arm.com

Today's mobile and consumer devices feature complex GUIs, and incorporate gaming capabilities. As a result, capability and performance requirements as well as demands for low power consumption and cost continue to increase.

Traditionally, 3D graphics accelerators require very high memory bandwidth. ARM's Mali family minimizes memory bandwidth by using a tile-based rendering scheme. Tile-based schemes break the screen into tiles and render one tile at a time to an on-chip tile-memory. This means that many memory accesses that would go off-chip with a traditional renderer remain on-chip using Mali cores. Since off-chip memory accesses are roughly an order of magnitude higher-power than on-chip memory accesses, this also helps minimize power consumption, which is key for a mobile consumer device requiring good battery life.

Cost is one drawback of tile-based renderers. Its greater complexity as compared with simple

"immediate-mode" renderers results in high gate counts, and thus, additional silicon area and cost. The Mali family combines tile-based and traditional "immediate-mode" rendering, thus minimizing memory bandwidth, cost and area, and making Mali cores significantly smaller than competing mobile tile-based renderers.

The Mali family consists of the Mali55 core, which implements the OpenGL ES 1.1 fixed-function rendering API, and the new Mali200 and MaliGP2 cores, which implement both the OpenGL ES 1.1 API and the OpenGL ES 2.0 programmable API. The OpenGL ES 2.0 API provides flexibility and high quality via a programmable rendering pipeline that replaces the fixed-function pipeline in OpenGL ES 1.1. This enables very high quality with low numbers of polygons, minimizing memory bandwidth while maximizing image quality.

The Mali55 core implements the OpenGL ES 1.1 pipeline from the rasterizer onwards. The Mali200 core also implements the OpenGL ES 2.0 pipeline stages from the rasterizer onwards. The MaliGP2 implements the Vertex Shader part of the graphics

pipeline. The Mali cores can also implement other APIs such as Microsoft DirectX Mobile.

To make the most of the small displays of cellphones, PDAs and mobile GPS devices, Mali cores offer full scene anti-aliasing (FSAA). FSAA reduces jaggies and improves image quality. They use a patent-pending technique that enables 4x anti-aliasing at virtually no cost to memory bandwidth and frame-rate, while 16x anti-aliasing can be performed, albeit with some increase in bandwidth and reduction in frame rate achieved.

The Mali graphics cores are optimized for a unified memory architecture where a single memory subsystem is shared between the main processor and the graphics accelerator. The Mali core's primary system interface is an AMBA 3 AXI master interface. An AMBA 2 APB interface provides for programming of on-chip registers. Accesses to Mali registers are minimized. This is because all graphics-processing commands for each frame are assembled in memory with little more than the base address of the commands in memory needing to be programmed into registers.

Graphic objects to be displayed are broken down into lists of triangles. The triangles are made up of three vertices, with the vertices being points in the X, Y and Z spaces. The vertices and commands for the MaliGP2 are written by the ARM processor to memory. The MaliGP2 processes the commands and vertices, and writes the transformed vertices back to main memory. In the case of the Mali55, this is handled in optimized software.

The Mali200 then rasterizes the triangles. It reads vertices and commands from main memory, and renders a tile at a time. The rasterized (or rendered) tiles are then written to memory to be read and displayed by an LCD or VGA controller. Since Mali cores are deferred renderers, the pipeline between the ARM core, MaliGP2, and between the MaliGP2 and the Mali200 is on a frame basis. A complete frame of data is generated and stored in the main memory between each component in the pipeline flow.

The Mali graphics cores require less memory bandwidth than other solutions, but memory system bandwidth is still important. The Mali cores are designed to work with SDRAM and share SDRAM with the main processor. Today's SDRAM devices have high clock speeds and high theoretical maximum bandwidth, but long latency.

The transactions from Mali cores are mostly bursts of four or eight words. 3D graphics engines differ from GPPs in that there are few dependencies between individual memory accesses. A memory transaction for another pixel can be issued before data is returned for the previous pixel, as there is no dependency between them. Thus, the 3D core can issue multiple outstanding transactions.

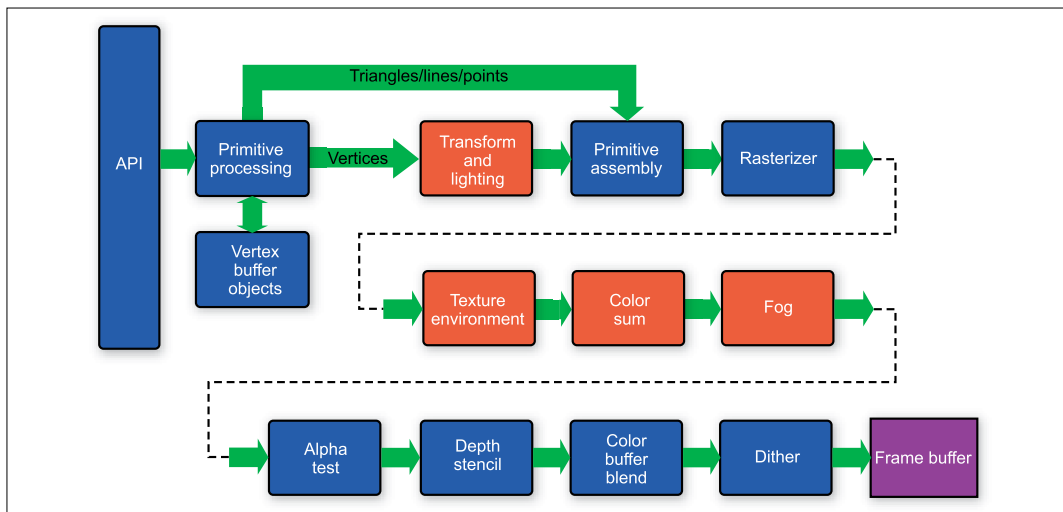


Figure: The OpenGL ES 2.0 API provides flexibility and high quality via a programmable rendering pipeline that replaces the fixed-function pipeline in OpenGL ES 1.1.

The ARM PrimeCell PL340 SDRAM controller supports SDR and DDR memory, including mobile and low-power variants of each over 16bit, 32bit and 64bit buses, with clock rates up to 166MHz. A typical mobile application would use 16bit or 32bit mobile DDR clocked at between 100-166MHz (200-333MHz data

rate). For higher performance systems, the ARM PrimeCell PL341 DDR2 SDRAM controller supports DDR2 clock speeds up to 400MHz (800MHz data rate).

Mali200 and Mali55 render graphics to a frame buffer in memory to be displayed by an LCD or video controller. The ARM PrimeCell PL111 Color LCD

controller supports monitors and LCD panels with up to 1,024 x 1,024 resolution. In addition to active TFT LCDs, it also supports older single and dual-panel STN LCD displays.

Mali cores are supplied with drivers for OpenGL ES2.0, ES1.1 (for compatibility with the fixed-function API) and OpenVG Vector

Graphics APIs. Licensees receive drivers for one OS of their choice (chosen from Linux, Symbian and Windows Mobile) and may optionally license (at additional cost) the driver-sets for additional operating systems. Besides OpenGL ES, the Microsoft DirectX Mobile API will be supported on Microsoft Windows Mobile.