Section 4 Indy Graphics

The Indy Graphics board shown in block diagram in Figure 10 comes standard on the Indy. It contains the following subsystems:

- The Raster Engine (REX3) ASIC, which converts geometric data processed by the CPU into pixel and line data that it then writes into the framebuffer.
- The framebuffer (VRAM), which contains the pixel color and overlay data for the 1280 x 1024 display, and the CID planes for arbitrary window clipping operations.
- The VC2 (video controller) ASIC, XMAP9 ASICs, and CMAP which, together with a 24-bit DAC, generate the RGB video signals sent to the monitor.

The GIO32-bis Expansion Subsystem is also integrated onto the Indy Graphics board.

FIGURE 10 The Indy Graphics Board. The shaded boxes show Silicon Graphics-designed ASICs.

The main bus on the Indy Graphics board provides a communication path between the elements of the graphics subsystem. The Indy Graphics board is connected to the GIO64 bus via the REX3 ASIC, which can access data from the CPU at a burst rate of 267 MBytes per second.

4.1 Graphics Processing

Indy Graphics makes extensive use of the power of the R4400/R4600 CPU by implementing the IRIS Graphics Library(tm) application programming interface as a set of highly efficient software algorithms. Because of the flexibility inherent in the software-based IRIS GL(tm) implementation, new graphics functions can be added as new software releases, without the need for costly hardware upgrades.

The CPU uses the IRIS GL to transform three-dimensional polygons to 2D screen

coordinates, decompose them into triangles, and finally decompose the triangles into spans by exact point sampling of the triangle's interior. These spans are then passed on to the REX3 ASIC, which interpolates them and renders the results into the framebuffer.

Figure 11 shows how 3D polygons are processed by the CPU and the REX3 ASIC. All transformation, clipping, and lighting calculations for points, lines, and polygons are performed in software by the CPU, including Z-buffer calculations, which are implemented in 32-bit resolution without the use of dedicated Z-buffer bitplanes.

> **FIGURE 11 Graphics processing on Indy. The majority of Indy Graphics computation is performed in software by the CPU.**

The Indy Graphics implementation offers several advantages in speed and flexibility over hardware-based graphics systems:

- IRIS Advanced Graphics functionality such as antialiased lines and polygons and exact point sampling is provided.
- Character (font) rendering is performed by the CPU, unencumbered by unnecessary filtering through graphics hardware.
- 2D rendering is optimized in software with algorithms that speed computation by eliminating the z coordinate parameters from calculations performed by the CPU, resulting in very high performance for 2D graphics applications.
- Because Indy Graphics performance is directly proportional to CPU speed, upgrades to faster CPUs automatically increase application performance.

4.2 The Raster Engine

The Raster Engine (REX3) is a pipelined, pixel-filling, and line-drawing ASIC which provides pixel shade, pattern, logicop, and blend operations for point, span, block, and integer or subpixel positioned line addressing modes. The REX3 ASIC achieves outstanding performance by the following techniques:

- buffering multiple graphics drawing requests in an on-chip input FIFO
- overlapping memory accesses, command executions, and command setups to achieve maximum pipeline throughput
- using page-mode accesses in the VRAMs when appropriate, because pixels in the same page can be accessed more quickly
- buffering multiple display bus (VC2, XMAP9, CMAP, RAMDAC, etc.) requests in a second, on-chip FIFO

The REX3 has five basic drawing primitives: point, line, span, block, and screen-to-screen copy. Draw commands can modify either pixel, overlay, or CID planes. The REX3 hardware recognizes a total X11 pixel address space of 64K x 64K pixels or a GL pixel address space of 4096 x 4096 pixels including 64 x 1K pixels of offscreen memory. REX3 can read pixel, overlay, pop-up, or CID information from both onscreen and offscreen memory.

REX3 supports both flat and linear (Gouraud) shading in both color index and RGB

formats with optional dithering. The peak fill rates for the REX3 are:

- 20 Mpixels/second for lines
- 400 Mpixels/second for clear (flat, undithered, and unstippled)
- 100 Mpixels/second for flat spans (dithered)
- 50 Mpixels/second for shaded spans

Dithering is window-relative in REX3, in contrast to the screen-relative dithering supplied in other GL(tm) implementations. This has the benefit of allowing a window to be moved while it is rendering, without creating artifacts. REX3 also includes the following features:

- It provides a window offset register, which holds the window origin and allows pixel addresses within a window to be stored relative to the origin.
- It maintains five screen masks. One of the screen masks is mapped into the user address space and can be set and used by calls in the IRIS Graphics Library application programming interface (API). The other four are available to the window manager to facilitate the clipping of overlapping windows. The addition of CID planes allows non-rectangular window clipping.
- It implements a set of sixteen logic operations on writes into the framebuffer. Window managers can use these functions to highlight selected areas and objects, to remove an object without removing objects in front or back of it, etc.
- It uses a screen to screen block move with logic op function, making movement of windows a quick and straightforward process.
- It considers the upper left corner of the screen to be the viewable origin, which supports X11 protocol screen addressing. A programmable register mode can be used to define the lower left corner as the viewable origin for GL.

In addition to features that support windowing, REX3 also has features to support IRIS GL and X11 library functions. These include:

- Bresenham's line drawing (including fractional-positioned, endpoint filtered antialiased lines).
- Two 32-bit recirculating pattern/stipple registers, which hold values that either prohibit the filling of a pixel (transparent stippling) or choose between the foreground and background color (opaque stippling). One register is of programmable length for X11 graphics and allows a repeat count for GL graphics; the other is 32-bit fixed length, and is useful for Z-buffering when in transparent mode. Both registers can be selectively enabled, allowing them to be used alone or in tandem.
- Burst-mode GIO64 bus transfers. When the GIO64 bus is operating in burst-mode, 8 bytes are transferred every cycle for a peak rate of 267 MBytes per second. REX3 can accept data into its FIFO at this rate, although processing the data takes longer. REX3 reads or writes pixel data over the GIO64 bus into a rectangular area at 43 million pixels per second. Both REX3 and MC1 support a robust rectangular DMA function that allows pixel data to be aligned to any arbitrary byte address and to have an arbitrary byte stride between scanlines.
- Commands for drawing IRIS GL color index and RGB antialiased lines. For color

index lines, REX3 can perform a color comparison test for each pixel for full support of the IRIS GL API's zsource (ZSRCCOLOR) mode. This technique is useful when depthcueing or antialiasing because it can prevent darker pixels from drawing over brighter pixels, such as in the case of a dark background. For RGB lines using the GL blendfunction to enable alpha-blending, REX3 computes alpha based on line coverage.

- 20-bit iterators for each of red, green, and blue to guarantee accurate color interpolation. This assures that accumulated color interpolation error is not visible on the screen. In addition, one of the 20 bits is used as an overflow/underflow bit to clamp invalid colors to either all zeros or all ones. Color index mode is supported by a 24-bit iterator in order to maintain accuracy for 12-bit pixels in color index mode.

4.3 Framebuffer and Clipping Planes (8-bit Graphics)

There are three kinds of bitplanes in the 8-bit Indy Graphics system: pixel, pop-ups, and clipping ID (CID) planes. The three bitplanes are implemented in Indy 8-bit graphics as a framebuffer in a 4x256Kx16 bank of VRAM.

Pixel planes hold 8 bits of information for each pixel of the display. In single buffer mode, the framebuffer contains an 8-bit color index or 8-bit RGB (3 bits each for red and green, 2 bits for blue). Four-bit color indices or 4-bit RGB (1 bit each for red and blue, 2 bits for green) are stored in the framebuffer in double buffer mode. While this resolution may seem low, 8-bit Indy Graphics makes up for the lack of per-pixel color definition by representing color values in main memory as 20-bits each of red, green, and blue, and then using dithering techniques (implemented in the REX3 ASIC) to create on-screen image definition of near full-color quality.

Pop-up planes, holding 2 bits per pixel, are used for pop-up menus, dialog boxes, and other components of a graphical user interface that temporarily obscure the image.

Clipping ID planes, also holding 2 bits per pixel, store information about which parts of windows are visible on the display. Multiple windows can overlap each other, totally or partially obscuring their contents. Individual pixels can be masked against a window boundary, thereby supporting non-rectangular windows without imposing additional overhead on the window system. The clipping ID (CID) planes are in the framebuffer VRAM.

The framebuffer is connected to the REX3 ASIC via the RB2s. Full use of the VRAM bandwidth is realized through a 128 pixel wide data path that connects the framebuffer to the RB2 ASIC. The framebuffer displays to a 1024 x 768 or 1280 x 1024 pixel monitor.

4.4 Framebuffer and Clipping Planes (24-bit Graphics)

24-bit Indy Graphics adds a fourth bitplane---overlay--- to the other three bitplanes in the 8-bit Indy Graphics system (pixel, pop-ups, and clipping ID (CID) planes). The four bitplanes planes are implemented as a framebuffer in a 12x256Kx16 bank of VRAM.

In 24-bit Indy Graphics, pixel planes hold 24 bits of information for each pixel of the

display. In single buffer mode, the framebuffer contains an 12-bit color index or 24-bit RGB (8 bits each for red, green, and blue).

As in the 8-bit graphic system, pop-up planes, holding 2 bits per pixel, are used for pop-up menus, dialog boxes, and other components of a graphical user interface that temporarily obscure the image. Clipping ID planes, also holding 2 bits per pixel, store information about which parts of windows are visible on the display.

The overlay planes on 24-bit Indy Graphics hold 8 bits per pixel.

The framebuffer is connected to the REX3 ASIC as described above for 8-bit Indy Graphics.

4.5 Color Lookup

The display of pixel data is controlled by the VC2 ASIC. It determines the timing of transfer cycles performed by the REX3. The serial VRAM data is buffered in the RO1 ASICs and then sent at half video rate to a pair of XMAP9 ASICs. The XMAP9 ASICs format the lookup index on a per-window basis as instructed by the VC2 for both color index and RGB modes, double-buffering, pixel depths, a pop-up cursor, and overlay planes.

Depending on the mode, each pixel to be displayed is formatted with an offset of n x 256 in order to index the desired section of the CMAP RAM as shown in Figure 12. This resulting lookup index is passed to the CMAP ASICs which generate 24-bit RGB values for display. These values finally enter the RAMDAC where they are gamma-corrected for the monitor by the use of a second level of lookup tables, and analog RGB output is produced.

FIGURE 12 Organization of the CMAP RAM (indexed by row).

4.6 GIO32-bis Expansion Subsystem

The two GIO32-bis expansion slots, connected directly to the GIO64 bus, provide direct access to the system for Silicon Graphics and third party plug-in boards for such applications as high-speed networking, image compression, video deck control, and additional I/O.

GIO32-bis is a cross between GIO32 and GIO64. It can be considered a 32-bit version of the non-pipelined GIO64 or a GIO32 with pipelined control signals.