CHAPTER 1

OVERVIEW

The MGR graphics adapter provides a high performance 3D graphics capability for the host system in which it is installed. The host system sends commands and data to the MGR to specify various 3D drawing primitives and drawing attributes which describe an objects geometric position, color, surface normal vectors and other graphics information. The drawing primitives include points, lines, convex and concave polygons, mesh triangles, characters and various types of curves and b-splines. The geometric position data is in world coordinates and the MGR adapter does the projection, viewing and modeling transformations to convert the 3D data into 2D screen coordinate data. The adapter performs the geometric calculations in hardware using a microcode driven floating point geometry engine. It also does the frame buffer rendering and display functions. The adapter connects to the host system via the Micro Channel Architecture (MCA) Bus.

The MGR adapter consists of four subsystems, as shown in Figure 1.1, which include the Host Interface Subsystem, the Geometry Subsystem, the Raster Subsystem and the Display Subsystem.

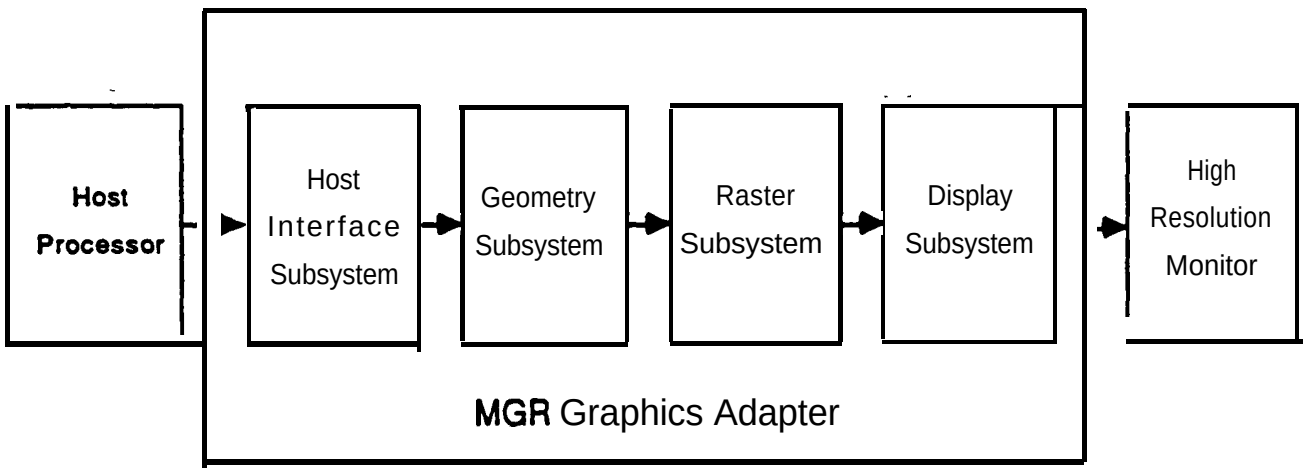

Figure 1.1 MGR Block Diagram

The Host Interface subsystem interfaces to the host processor via the Micro Channel Architecture (MCA) bus. It provides all of the necessary data and control signals required by the MCA bus. It also interfaces to the other three subsystems over a local bus which is compatible with the SGI private bus. The Host Interface Subsystem provides the programmable Option Select (POS) registers which are used to configure the MCA bus interface aspects of the adapter.

The Geometry subsystem contains the Geometry Engine which is used to perform the geometric transformations and lighting calculations on the graphics data. It also performs the clipping and high level rendering calculations before sending the data to the Raster subsystem. The Geometry Engine performs high speed floating point calculations under the control of the onboard microcode. The host system issues commands to the Geometry Engine by sending command tokens and data parameters down a FIFO. The Geometry Engine reads the FIFO and performs the desired functions. The Geometry Subsystem also contains hardware which controls the addressing of the hardware components in the Raster and Display Subsystems.

The Raster subsystem receives instructions and other data parameters from the Geometry Subsystem and does the low level rendering of the data into the image frame buffer bitplanes. It also handles the Z buffer depth comparisons and updates if the optional Z buffer card is installed. The pixel values for each of the 1.3 million pixels on the 1280 by 1024 high resolution monitor are stored in the video random access memory (VRAM) frame buffer for display on the monitor at a selected refresh rate.

The Raster Subsystem provides the necessary data and control signals to manage the frame buffer bitplanes as well as the Window ID bitplanes and the Auxiliary bitplanes. The Window ID bitplanes are used to control the display format for up to sixteen different on screen windows. The Auxiliary bitplanes are used for drawing pop-up menus, overlays and underlays. The Raster Subsystem also contains two hardware cursor chips.

The Display subsystem performs the necessary operations to convert the frame buffer image data into analog RGB signals which are sent to the high resolution RGB monitor. The Display Subsystem uses the Window ID bitplane data to determine the format of the pixel data stored in the frame buffer. The frame buffer pixel data can be either a color index value or an RGB value. The frame buffer bitplanes can also be subdivided Into two buffers for flicker free motion of on screen objects. The Display Subsystem performs the necessary operations to access the frame buffer as a single buffer or as a double buffer. The Digital-to-Analog Converters (DACs) are used to convert the digital RGB data into analog RGB data which output to the monitor. The Display Subsystem also provides the necessary timing signals to allow four different types of monitors to be connected to the adapter.

The following chapters describe the hardware configurations, the Host Interface Subsystem, the Geometry Subsystem, the Token Definitions, the Raster Subsystem and the Display Subsystem in detail. The chapters describe the external interfaces, the major components, the programmable registers, the interrupts, the basic operations and the programming considerations for each of the subsystems.

This document is intended for a very wide audience which accounts for the detailed nature of the document. It is meant as a reference for both hardware and software engineers with the primary emphasis being for software engineers. The document is intended for use by both internal Silicon Graphics engineers and outside Original Equipment Manufacturers (OEM). The document attempts to provide example code fragments which show how to program the adapter.For readers who have access to SGI source code these code fragments are not really necessary. However, for OEM customers who have purchased only the MGR adapter hardware and do not have access to the SGI source code, the code fragments are meant to give them at least a chance at doing something useful with the adapter.