A SOLID FOUNDATION FOR CAD/CAM SYSTEM BUILDERS

Lexidata’s 8100/GS is a unique hardware/software combination which offers the system builder the solid foundation necessary in the design of powerful, interactive CAD/CAM or other application-oriented systems. The 8100’s high-level graphics functionality simplifies application development, enabling the system builder to meet rigorous design deadlines. Its wide range of features provide flexibility for specific needs.

Street grid map (above) as displayed on the 8100/GS distributed graphics processing system.

FEATURES
• Dual Processor Architecture
• High-Level Functionality
• 2 Billion X 2 Billion Virtual Addressability
• Local Viewing Operations and Transformations
• Local Peripheral Support
• Power-Up Diagnostics

BENEFITS
• Speeds Throughput and Reduces Host Workload
• Simplifies Application Development
• Allows Local Storage of High-Precision Models
• Offers High Degree of User Interactivity
• Eases System Interface Bottlenecks
• Insures Trouble-Free Operation
SYSTEM ARCHITECTURE

The 8100/GS utilizes a dual processor architecture consisting of a geometry processor, and a display processor. The user's host computer, functioning as the master controller, completes the three-processor system hierarchy. Tasks are distributed to the appropriate processor, resulting in balanced system throughput. (See Figure 1.) allowing large design data bases to be stored locally.

Display Processor

The display processor is a high-speed, Lexidata-proprietary bipolar processor, designed to rapidly perform bit-map manipulations such as vector-to-raster conversion and area fill. In addition, the display processor controls other video display functions.

Figure 1. SYSTEM BLOCK DIAGRAM—The 8100's multiple processors perform operations in parallel, boosting overall system throughput. In conjunction with the user's host computer, the 8100/GS forms a highly-interactive distributed graphics system.

Geometry Processor

The geometry processor is a 16/32-bit Motorola MC68000 microprocessor, with 256KB Random Access Memory (RAM). High-speed, 32-bit arithmetic operations make the 68000 an ideal geometry processor; all viewing operations, including geometric transformations, are performed locally. Additionally, the geometry processor manages four serial ports used for interfacing graphic input devices such as a data tablet, keyboard, joystick or trackball. Data from the input devices is buffered by the geometry processor until requested from the user's host software, eliminating the host requirement to respond quickly to device interrupts. Up to two megabytes of error-correcting RAM is optionally available to the user, such as the user-programmable hardware cursor and performs lookup table manipulations used for color selection.

Interfaces

The geometry processor and display processor are linked by a high-speed, Direct Memory Access (DMA) interface, allowing the two processors to perform operations in parallel—increasing the overall system throughput.

High-speed, 16-bit parallel interfaces are available for most popular mini-computers. An RS-232 host serial interface at up to 19.2K baud is also available.

Power-Up Diagnostics

After power-up, a set of PROM-resident diagnostics exercise all major components.
Hardware failures are reported via a set of four LEDs which indicate the source of the problem. The sequence is accomplished in approximately five seconds. Successful power-up is indicated by a system message on the display. Power-up diagnostics insure trouble-free operation and rapid field repair.

**Variety of Configurations**

The 8100/GS is a self-contained, rack-mountable unit, available in a wide variety of configurations. Display resolutions are available in 640x512/1280x1024 color or 1280x1024 monochrome. The 640x512 color and 1280x1024 monochrome systems are available for 50/60 Hz non-interlaced refresh, providing flicker-free operation. Up to 4096 colors can be displayed simultaneously from a palette of 16.7 million. Options include LX/GP1 provides a high-level, functional interface for the user’s application software. LX/GP1 functionality was initially modeled after the proposed Core System graphics standard, a commonly-recognized set of high-level graphics functions. The implementation of LX/GP1, however, extended the Core methodologies in several key areas, making LX/GP1 conceptually similar to the more-recently proposed Graphics Kernel System (GKS). An important difference is that LX/GP1 performs all viewing operations locally in the 8100/GS. (See Figure 2.)

**LX/GP1 Unique Features**

LX/GP1 employs a descriptive Object Data Structure (ODS), allowing entire objects to be stored locally in the 8100/GS. The stored hardware blink, hardware pan and zoom, and graphic input devices such as data tablets, keyboards, joysticks and trackballs. (See 8100/GS Configuration Summary.)

**LX/GP1 SYSTEM SOFTWARE**

LX/GP1 is a sophisticated software package which runs locally in the 8100/GS. In conjunction with a host FORTRAN library, object can be manipulated dynamically through a series of simple commands provided by the host application program. Since LX/GP1 performs all viewing operations locally, screen update occurs rapidly in response to user interaction. Additionally, LX/GP1 offers a uniform means of handling graphic input devices, relieving the application program of this burdensome task.
LOCAL DATA STORAGE

Coordinate Systems

The 8100/GS utilizes three distinct coordinate systems: World Coordinates (WC), Normalized Device Coordinates (NDC), and Device Coordinates (DC).

World Coordinates

A World Coordinate description is a geometrical representation of the object in two-dimensional space. All objects are stored locally in terms of their World Coordinate descriptions. The 8100/GS represents \( x, y \) coordinates with 31-bits of precision, yielding a virtual address space of two billion \( x \) two billion units—a significant advantage when designing complex or high-precision models.

Normalized Device Coordinates

Normalized Device Coordinates represent an abstract coordinate system which lies between World Coordinates and Device Coordinates. Since graphic input devices have various addressing ranges, and physical display resolutions may also vary, LX/GPI utilizes NDC to map all devices to a common, “idealized” set of coordinates. This simplifies application programming and provides software device independence with respect to screen/bit-map resolution.

Device Coordinates

Device Coordinates are directly related to the screen/bit-map resolution. LX/GPI automatically translates World Coordinate object descriptions into Device Coordinates, in order to display images on the screen.

Figure 3. 2 BILLION X 2 BILLION VIRTUAL ADDRESSABILITY—LX/GPI utilizes three coordinate systems. Since LX/GPI stores and manipulates objects in a high-precision World Coordinate Space, the 8100/GS is ideal for creating complex or large-scale models.
OBJECT DATA STRUCTURE

World Coordinate objects are stored locally as a collection of descriptive graphic entities and are referred to as the Object Data Structure (ODS). There are three basic types of graphic entities which comprise the ODS: primitives, cells and segments. (See Figure 4.)

Primitives
The 8100/GS supports output primitives such as lines, polylines (connected line sequences), polygons, circles, text and messages. Primitives are described in terms of their most elementary defining characteristics (i.e. the endpoints of a line, or the center and radius of a circle) but also have other characteristics such as color or line thickness known as attributes. In contrast to "Display List" systems which store output primitives as a series of commands, LX/GPI utilizes an approach (similar to GKS) where primitives are stored as descriptive blocks, thereby providing the means for simple editing or powerful manipulation of the ODS. (For a complete listing of output primitives supported by LX/GPI, see the Command Summary.)

Segments
A segment is a collection of primitives and/or cells which represent a single unit. Segments are the entities upon which the powerful LX/GPI image manipulation algorithms operate. LX/GPI supports two classes of segments: temporary and retained. Temporary segments are not permanently stored in the ODS and can be used to display items temporarily on the screen. The retained segment is stored in the ODS for subsequent manipulation or editing. An important feature of the 8100/GS is that segments are stored with enclosing rectangle data. This greatly simplifies determination of those segments which fall in a particular area of interest, allowing LX/GPI to process the ODS efficiently during viewing operations.
Cells

Cells allow the user to create collections of primitives for the purpose of referring to them as a single entity. Cells differ from segments in that segments become immediately visible once created, whereas cells remain invisible until "instantiated" within a segment. "Cell instantiation" is the process whereby a previously-defined cell is given a location and orientation in World Coordinate Space. Cells provide ODS storage efficiency. Although a single copy of the "cell definition" is stored, the cell may be instantiated many times within one or more segments. Another important cell characteristic is nesting. As can be seen in Figure 5, a cell definition can itself include a cell instance of a previously defined cell. This process can occur to a depth of 32 levels, allowing the user to formulate complex cell structures to meet particular application requirements.

Attributes

Primitives, segments and cells have additional descriptive characteristics known as attributes. LX/GPI supports two major classes of attributes: static, those which once assigned remain fixed; and dynamic, those which may be altered as required.

Static Attributes

Primitives can be created with a variety of static attributes. Lines, for instance, can have color, width, and pattern specified. Polygons can be filled or unfilled. For filled polygons, the user may specify a unique fill pattern. In the case of text, size may be specified. (For a complete listing of attributes supported by LX/GPI, see the Command Summary.)

Figure 5. DESCRIPTIVE DATA STRUCTURE—LX/GPI stores graphic entities as descriptive blocks. Segments can be dynamically manipulated using simple Host Commands. Cells can be nested to a depth of 32 levels. "Cell Instantiation" saves Object Data Structure storage space.
A cell reflects the attributes of its associated primitives. Additionally, each cell instance has its own static attribute which allows the user to specify an offset and orientation for the cell; a single cell can therefore be instantiated in several different orientations. This cell attribute feature eliminates the need to define multiple cells, reducing ODS storage overhead.

**LOCAL MANIPULATION OF THE ODS**

The ODS can be manipulated using a variety of powerful LX/GPI functions. Segments may be transformed dynamically and edited locally without having to be recreated. These local processing capabilities contribute to making the 8100/GS a highly-interactive graphics subsystem, while reducing the requirement for a high-speed host interface.

**Dynamic Attributes**

Dynamic attributes allow the user to modify segments independently. Simple FORTRAN calls are used to dynamically alter position, orientation and visibility. For example, in a mechanical application, a gear may be represented using a segment. The operator would then be able to rotate, scale and position the gear to suit the design.

**Transformations**

One class of dynamic attributes — transformations — include translation, rotation and scaling. (See Figure 7.) These attributes may be applied separately or in combination in order to achieve the desired result. Both scaling and rotation are possible around an arbitrary point, enhancing the flexibility with which the user can manipulate the Object Data Structure.

Dynamic attributes affecting visibility include the capability to define a segment as visible or invisible within a particular view. This feature can be used to remove segments selectively from the display — decluttering. Additionally, segments can be highlighted (color/intensity change), providing useful operator feedback.

**Entity Pick**

Entity pick refers to the ability of users to single out a particular entity (segment, cell, or primitive) for editing, inquiry or manipulation. The ability to pick an entity provides the user with a convenient mechanism for manipulating or updating the 8100/GS Object Data Structure (i.e. “picking” a single Integrated Circuit (IC) on a Printed Circuit (PC) layout and moving it to a new location). LX/GPI performs the “pick” operation locally in the 8100/GS. An area of interest, a pick aperture, can be set under user control. At the time of creation, the user may specify whether entities may be picked. All pickable

![Figure 7. DYNAMIC MANIPULATION OF SEGMENTS - LX/GPI performs all transformations locally in the 8100/GS. Simple host commands mean low interface traffic. The result – a powerful and highly-interactive design system.](image-url)
entities which fall within the specified pick aperture will be returned sequentially to the host application program. Because segments are stored in the ODS with enclosing rectangle information, the pick operation occurs rapidly with only those segments falling within the pick aperture being selected.

**Segment/Cell Editing**
The user may edit segments and cells freely. LX/GPI allows primitives to be deleted or appended to segments and cells without requiring them to be recreated. As items are deleted, LX/GPI reclaims freed space automatically, utilizing available ODS storage space efficiently. Local editing results in significant savings in both host processing and interface traffic, providing rapid response to operator inputs.

**CONTROLLING THE VIEW**
The 8100/G5 provides the user with a good deal of control over viewing the object described by the ODS. The user may specify one or more rectangular areas in World Coordinate Space. LX/GPI will automatically project these areas onto the physical display. The effect is to permit any rectangular portion of the object to be displayed at any magnification. (See Figure 8.)

**Windows**
As depicted in Figure 8, windows are rectangular areas specified in World Coordinates. LX/GPI automatically selects segments which fall within these areas and performs clipping operations as necessary. The user may interactively change window size or location, immediately updating the display with a different view of the object. Changing window size provides a “soft zoom” capability; changing window location effectively allows the user to pan across objects stored in the ODS. Local World Coordinate windowing results in fast screen update with minimal host intervention.

---

**Figure 8.** MULTIPLE ACTIVE VIEWS - With LX/GPI, the user is free to partition the physical display in up to 32 work areas simultaneously! He can then pan across the locally stored, high-precision ODS and display separate portions of the object at any magnification.
Viewports

In order to display a window, the user defines a viewport— an area of the screen onto which the window is to be projected. So as to provide a degree of device independence, the user specifies the viewport in terms of NDC space. Areas within this 32Kx32K space correspond directly to areas on the physical screen, regardless of its resolution. The user’s application need not concern itself with physical display resolution; LX/GPI performs the projection directly from the WC space using the NDC viewport as a map.

A unique feature of the 8100/GS is that up to 32 multiple viewports may be active simultaneously. In addition, viewports may overlap, further increasing the user’s flexibility with respect to the display.

View Surfaces

The user may also control the range of colors available within a view. This is accomplished by specifying a view surface which relates a set of display memory planes to a specific window and viewport definition. The range of colors available within a view is determined by the number of memory planes selected. This feature also allows the user to allocate the display memory efficiently.

View Association

Once the view is created, the user has control over which segments are displayable within the view. Using the default case, all segments within the rectangular window area will be displayed. However, the user may select one or a group of segments, and associate them with a view. Only those segments which have been associated will be displayed within the viewport. This enables the user to display convenient groupings of segments. In the case of a PC layout application, segments describing the board layers could be associated with specific views. IC segments could be associated with yet another view. The user would be able to display the entire board by making all views active simultaneously, while layers could easily be stripped away as required.

View Pipeline

The process of displaying the object on the physical display once a window and viewport have been established is called the “view pipeline.” As can be seen in Figure 9, this process begins by applying transformations as required. Objects are then clipped against the WC window. In a standard Core System implementation, this is the point at which retained segments would be stored in NDC space. Since the 8100/GS allows segments to be stored in WC space, LX/GPI simply uses the viewport definition (NDC space) as a map and issues commands to the display processor in actual Device Coordinates.

![Diagram](image)

Figure 9. ALL VIEWING OPERATIONS PERFORMED LOCALLY—LX/GPI handles the entire view pipeline. Note that the 8100/GS has no need to store NDC segments as in a Core System implementation.
Maintaining the ODS in World Coordinates

The advantage of maintaining an Object Data Structure in World Coordinates as opposed to a clipped normalized device coordinate image is apparent in the following example. Clipping occurs when an object such as an arrow crosses the defined window. The image may then be transformed to bring it entirely into view. (See Figure 10.) A system which implements the Core proposal would retain only clipped images. Therefore, after the transformation, only the portion of the arrow that was originally within the viewing window would be displayed. Because the 8100/GS retains a description of the entire arrow, the complete image is displayed on the screen.

CONTROL FUNCTIONS

The 8100/GS can support up to three graphic input devices: a data tablet, a joystick/trackball, and a keyboard. This eliminates the need for the host to respond quickly to device interrupts which may represent a severe burden, especially if the host is not dedicated to the graphics application.

Locator Device Input

Data tablets, joysticks and trackballs are classified as locator-type devices. LX/GPI can use locator input to perform local cursor tracking. Additionally, LX/GPI can return locator input to the user’s application program in either World Coordinates, Normalized Device Coordinates, or actual Device Coordinates. This feature makes cursor tracking within specific viewports a straight-forward operation, regardless of the graphic input device.

Keyboard Input

Input from the Lexidata keyboard can be returned directly to the host on a character-by-character basis, or it may be buffered and returned a line at a time. Additionally, text can be echoed locally on the screen without host intervention. These features can be especially helpful to an overburdened host.

Error-Logging

LX/GPI features an error-logging facility for debugging new user applications. Errors can be reported to the display or on a separate console device, and can be returned to the host application program upon request.

SUMMARY

Lexidata’s 8100/GS is a powerful hardware/software combination, designed to meet the flexible needs of the system builder.

A sophisticated software package, LX/GPI runs locally in the 8100/GS graphics subsystem. LX/GPI’s high-level functionality streamlines application development and minimizes system implementation time.

The 8100’s local intelligence frees the host of many computation-intensive tasks and provides rapid response to operator requests regardless of host workload.
**LX/GP1 COMMAND SUMMARY**

The following is a list of LX/GP1 functions currently available on the 8100/GS. They are organized into logical groups for convenient reference. All commands, with the exception of the IDOS (Image Display Operating System) Emulation Commands, operate on the 8100/GS Object Data Structure. IDOS is a set of functions previously offered on Lexidata 3400 systems. IDOS emulation provides software compatibility for 3400 IDOS users. In addition, IDOS emulation enables the user to send commands directly to the display processor Device Coordinates as desired.

**OUTPUT PRIMITIVE FUNCTIONS**

- GCOLI: Add a Cell Instance
- GCIR: Create Circle Defined by Radius
- GCR: Create Circle Defined by Specified Point
- GINQCP: Inquire Current Position (CP)
- GLINA: Generate Line from CP to Specified Point
- GLINR: Create Line from CP to Relative Offset
- GMV: Move CP to Absolute Position
- GMVR: Move CP Relative to Current Position
- GPLNA: Create Multivertex Line Defined by Specified Points
- GPLN: Create Multivertex Line Defined by Relative Offset
- GPOL: Create Polygon Defined by Specified Vertices (WC)
- GPOLR: Create Polygon Defined by Relative Offsets (WC)
- GTXNDC: Create Text String (NDC)
- GTXTC: Create Text String (WC)

**OBJECT SEGMENTATION**

- GLSEG: Close Opened Segment
- GCRC: Create Cell Definition
- GCRSEG: Create Retained Segment
- GCTSEG: Create Temporary Segment
- GDASEG: Delete All Retained Segments
- GDLCDF: Delete Cell Definition
- GDLINS: Delete Cell Instances
- GDLRSG: Delete a Retained Segment
- GRNSEG: Rename Retained Segments or Cells
- GRPCFD: Replace Cell Definition
- GRPSEG: Replace Retained Segment or Cell Definition

**ATTRIBUTE SETTING FUNCTIONS**

- GDCOLI: Define Color Index Table
- GDINTI: Define Lookup Table Intensities
- GRSXFM: Reset Cell Transformation
- GSCHSZ: Set Text Character Size
- GSCNDX: Set Intensity Index for Output Functions
- GSETDM: Set Display Mode
- GSFTY: Set Character Font
- GSPAT: Set the Line Pattern Parameters
- GSWID: Set the Line Width or Weight
- GSMSIZ: Set Message Text Characters Size (NDC)
- GSSEG: Set Highlighting Attribute (Dynamic)
- GSVIS: Set Segment Visibility (Dynamic)
- GTRSLC: Translate a Cell Instance (Static)
- GTRSR: Translate Retained Segment (Dynamic)
- GXFMC: Transform a Cell Instance (Static)
- GXFM: Transform Retained Segment (Dynamic)
- GDP: Send Fill Pattern Definition

**VIEWING FUNCTIONS**

- GACVS: Activate Viewing Surface
- GACVU: Activate Specified View
- GCF: Control New Frame Actions
- GDAV: Remove a Viewing Surface
- GDAVI: Deactivate Specified View
- GDEFVS: Define View Surface
- GDF: Define View
- GDFWIN: Define Window Position
- GDFPV: Define Viewpoint Position
- GDYNAM: Select Dynamic Mode
- GFRAM: Provide New Frame
- GINTVS: Initialize Selected View Surface
- GMPCUR: Force All Delayed Picture Changes
- GSVIS: Set Immediate Visibility
- GSUAS: Set View Association
- GTRMV: Terminate Access to Specified Surface
- GRUPD: Update Graphics

**INPUT PRIMITIVE FUNCTIONS**

- GCBK: Clear Keyboard Echo Area
- GDFB: Define a Single Button
- GDFBG: Define Group of Buttons
- GDFBB: Define Keyboard Echo Type
- GDFLM: Define Locator Transformation Matrix
- GDFLPT: Define Locator Port Extents
- GDVS: Enable Device
- GDVEN: Initialize Device
- GDVTR: Terminate I/O Operation
- GRRB: Read Single Button Status
- GRRST: Read Status of All Buttons
- GRRDK: Read Keyboard
- GRRDLC: Read Locator Coordinates (WC and NDC)
- GRRDNC: Read Locator Coordinates (NDC)
- GRRDVAL: Read Locator Coordinates (DC)
- GRRDWC: Read Locator Coordinates (WC)
- GRSLG: Turn Off a Group of Lights
- GRSL: Turn Off a Single Light
<table>
<thead>
<tr>
<th>ENTITY SELECT (PICK)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GAPND</td>
<td>Pick Aperture Dimensions (NDC)</td>
</tr>
<tr>
<td>GAPWC</td>
<td>Set Pick Aperture Dimensions (WC)</td>
</tr>
<tr>
<td>GPCLS</td>
<td>Close Pick Function</td>
</tr>
<tr>
<td>GPDEV</td>
<td>Open Pick</td>
</tr>
<tr>
<td>GPICK</td>
<td>Pick Segment or Primitive</td>
</tr>
<tr>
<td>GPLAB</td>
<td>Set Primitive Base Label</td>
</tr>
<tr>
<td>GPNDC</td>
<td>Open Identify Function (NDC)</td>
</tr>
<tr>
<td>GPWC</td>
<td>Open Identify Function (WC)</td>
</tr>
<tr>
<td>GRDAP</td>
<td>Read Pick Aperture Center</td>
</tr>
<tr>
<td>GRLAB</td>
<td>Read Most Recent Primitive Label</td>
</tr>
<tr>
<td>GRPRI</td>
<td>Get Labelled Primitive</td>
</tr>
<tr>
<td>GSDET</td>
<td>Set Detectability</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SEGMENT EDITING AND READBACK</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GDCPR</td>
<td>Delete Current Primitive Record</td>
</tr>
<tr>
<td>GDPRI</td>
<td>Delete Labelled Primitive Record</td>
</tr>
<tr>
<td>GNPRI</td>
<td>Get Next Primitive</td>
</tr>
<tr>
<td>GOCEL</td>
<td>Open Cell</td>
</tr>
<tr>
<td>GOSEREG</td>
<td>Open Retained Segment</td>
</tr>
<tr>
<td>GTPRI</td>
<td>Return Primitive Record</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDOS EMULATION FUNCTIONS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DSBCTL</td>
<td>Blink Control</td>
</tr>
<tr>
<td>DSBLIN</td>
<td>Set Blink Rate</td>
</tr>
<tr>
<td>DSBLOC</td>
<td>Send Data Block</td>
</tr>
<tr>
<td>DSBBLR</td>
<td>Read Data Block</td>
</tr>
<tr>
<td>DSCER</td>
<td>Erase Cursor Matrix</td>
</tr>
<tr>
<td>DSCFG</td>
<td>Set Hardware Configuration for the Display</td>
</tr>
<tr>
<td>DSDCHAN</td>
<td>Define Channels for Write Operations</td>
</tr>
<tr>
<td>DSCIR</td>
<td>Display Circle</td>
</tr>
<tr>
<td>DSCLD</td>
<td>Load Cursor Matrix</td>
</tr>
<tr>
<td>DSCRRL</td>
<td>Clear Display</td>
</tr>
<tr>
<td>DSCLS</td>
<td>Terminate I/O in Progress and Detach Driver</td>
</tr>
<tr>
<td>DSCSL</td>
<td>Select Hardware Cursor</td>
</tr>
<tr>
<td>DSCXY</td>
<td>Set Cursor Position</td>
</tr>
<tr>
<td>DSECHO</td>
<td>Set Echo Mode</td>
</tr>
<tr>
<td>DSGET</td>
<td>Sequential Pixel Read</td>
</tr>
<tr>
<td>DSGKB</td>
<td>Get Keyboard Data and Set Lamps</td>
</tr>
<tr>
<td>DSGXY</td>
<td>Read Cursor Position</td>
</tr>
<tr>
<td>DSFWT</td>
<td>Wait for Read Operation to Complete</td>
</tr>
<tr>
<td>DSLIM</td>
<td>Set Rectangular Limits</td>
</tr>
<tr>
<td>DSLLU</td>
<td>Generate Lookup Table Ramp</td>
</tr>
<tr>
<td>DSLRD</td>
<td>Read from Lookup Table</td>
</tr>
<tr>
<td>DLSWIT</td>
<td>Write to Table</td>
</tr>
<tr>
<td>DSMOV</td>
<td>Execute Movie</td>
</tr>
<tr>
<td>DSMRG</td>
<td>Set Display Margins</td>
</tr>
<tr>
<td>DSOEN</td>
<td>Open Specified Device</td>
</tr>
<tr>
<td>DSOWT</td>
<td>Wait for Write Operation to Complete</td>
</tr>
<tr>
<td>DSPLD</td>
<td>Load and Start Software</td>
</tr>
<tr>
<td>DSPNT</td>
<td>Display Points</td>
</tr>
<tr>
<td>DSPUT</td>
<td>Sequential Pixel Write</td>
</tr>
<tr>
<td>DSRNR</td>
<td>Random Pixel Read</td>
</tr>
<tr>
<td>DSRNW</td>
<td>Random Pixel Write</td>
</tr>
<tr>
<td>DSSAO</td>
<td>Set Text Parameters</td>
</tr>
<tr>
<td>DSSL</td>
<td>Set Trackball/Joystick Lamps</td>
</tr>
<tr>
<td>DSTXT</td>
<td>Display Text</td>
</tr>
<tr>
<td>DSVDR</td>
<td>Display Vector</td>
</tr>
<tr>
<td>DSZOM</td>
<td>Execute Zoom and Pan</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CONTROL FUNCTIONS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GBSWAP</td>
<td>Swap All Bytes</td>
</tr>
<tr>
<td>GDLED</td>
<td>Define Error Logging Device</td>
</tr>
<tr>
<td>GDLP</td>
<td>Define Error Message Area</td>
</tr>
<tr>
<td>GINIT</td>
<td>Initialize 8100/GS</td>
</tr>
<tr>
<td>GINQDR</td>
<td>Inquire Maximum View Surface Display Coords</td>
</tr>
<tr>
<td>GRMRER</td>
<td>Report Most Recent Error</td>
</tr>
<tr>
<td>GTERM</td>
<td>Terminate 8100/GS Operations</td>
</tr>
<tr>
<td>LDATA</td>
<td>Host Parameter Table</td>
</tr>
</tbody>
</table>
## 8100/GS CONFIGURATION SUMMARY

<table>
<thead>
<tr>
<th>Model</th>
<th>8110</th>
<th>8120</th>
<th>8130</th>
<th>8140</th>
<th>8150</th>
<th>8160</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display resolution</td>
<td>640x512</td>
<td>640x512</td>
<td>1280x1024</td>
<td>1280x1024</td>
<td>1280x1024</td>
<td>640x512</td>
</tr>
<tr>
<td>Display refresh rate</td>
<td>50/60 Hz</td>
<td>50/60 Hz</td>
<td>50/60 Hz</td>
<td>50/60 Hz</td>
<td>50/60 Hz</td>
<td>50/60 Hz</td>
</tr>
<tr>
<td>non-interlaced</td>
<td>non-interlaced</td>
<td>non-interlaced</td>
<td>interlaced</td>
<td>interlaced</td>
<td>non-interlaced</td>
<td></td>
</tr>
<tr>
<td>Standard display memory</td>
<td>4 planes Color</td>
<td>8 planes Color</td>
<td>1 plane Monochrome Color</td>
<td>4 planes Color</td>
<td>3 planes Color</td>
<td>12 planes Color</td>
</tr>
<tr>
<td>Video output</td>
<td>16</td>
<td>256</td>
<td>Not Available</td>
<td>16</td>
<td>8</td>
<td>4096</td>
</tr>
<tr>
<td>Maximum number of simultaneously displayable colors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Color lookup table</td>
<td>10x8</td>
<td>10x8</td>
<td>Not Available</td>
<td>4x4</td>
<td>4x4</td>
<td>12x8</td>
</tr>
<tr>
<td>Overlay option</td>
<td>3 planes</td>
<td>3 planes</td>
<td>Not Available</td>
<td>1 plane</td>
<td>1 plane</td>
<td>4 planes</td>
</tr>
<tr>
<td>Hardware blink</td>
<td>Option</td>
<td>Option</td>
<td>Option</td>
<td>Option</td>
<td>Option</td>
<td>Option</td>
</tr>
<tr>
<td>Hardware zoom</td>
<td>Option</td>
<td>Option</td>
<td>Option</td>
<td>Option</td>
<td>Option</td>
<td>Option</td>
</tr>
<tr>
<td>Standard ODS memory</td>
<td>256KB</td>
<td>256KB</td>
<td>up to 2MB</td>
<td>up to 2MB</td>
<td>up to 2MB</td>
<td>up to 2MB</td>
</tr>
<tr>
<td>Optional 512KB ECC ODS memory</td>
<td>up to 2MB</td>
<td>up to 2MB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Host parallel interface</td>
<td>Available</td>
<td>Available</td>
<td>Available</td>
<td>Available</td>
<td>Available</td>
<td>Available</td>
</tr>
</tbody>
</table>

## 8100/GS SYSTEM SPECIFICATIONS

### Alphanumeric
A table-driven stroke font supporting upper and lower case, alphanumerics, and punctuation for graphic text. A pixel-oriented font for message text.

### Cursor
Size and shape of non-destructive cursor is user-loadable within 64 pixel x 64 pixel matrix. Full screen cross-hair is also selectable.

### Data Transfer Rate
Up to 800K words per second (16 bits/word) from host computer in burst mode.

### Power Requirements
- 115 VAC ± 10% 47-63Hz (3 wire)
- 230 VAC ± 10% 47-63Hz (3 wire)
- 600 watts average.
Requirements vary depending on configuration size.

### Operating Temperature
10 to 40 degrees C

### Storage Temperature
- 35 to 70 degrees C

### Operating Relative Humidity
10% to 90% (non-condensing)

### Storage Relative Humidity
10% to 90% (non-condensing)

### Altitude
8,000 ft.

### Acoustic Noise Level
The acoustic noise level shall not exceed the NC-60 noise criteria curve.

### Regulatory Compliance
Complies with the conducted and radiated EMI emission levels defined in FCC docket #20780 for class A equipment.

### Chassis Dimensions
8.75" high x 19" wide x 27" deep (not including monitor).

### Weight
60-100 lbs. including power supply (not including monitor).
CORPORATE WORLD HEADQUARTERS
Lexidata Corporation
755 Middlesex Turnpike
Billerica, MA (USA) 01865
(617) 663-8550
TWX: 710-347-1574
U.S. offices:
Boston
Chicago
Houston
Ithaca
Los Angeles
New York
Raleigh
San Francisco
Washington, D.C.

EUROPEAN HEADQUARTERS
Lexidata, LTD
Weaver House
Station Road
Hook, Basingstoke
Hampshire
RG27 9JY
UNITED KINGDOM
(44) (025672) 4334
TWX: 851/858364 LEXUK G
International offices:
Australia
France
Israel
Japan
Netherlands
Switzerland
United Kingdom
West Germany