McIDAS

Man computer Interactive Data Access System

SDA Hardware Manual

Issued June 1992

Revised 1992

Space Science and Engineering Center
University of Wisconsin - Madison
1225 West Dayton Street
Madison, WI 53706
Telephone 608-262-2455
TWX 608-263-6738

Preface

The SSEC Display Adapter is a plug-in printed circuit board developed by the University of Wisconsin's Space Science and Engineering Center (SSEC) that allows an IBM® PS/2™ * desktop computer to function as a McIDAS (Man computer Interactive Data Access System) workstation. The workstation is the component of the McIDAS that provides the animated display of satellite imagery and weather data.

The SSEC Display Adapter (hereafter called "SDA") Workstation consists of off-the-shelf units and the SDA. This manual was prepared by SSEC to aid technicians in diagnosing hardware failures on the SDA to the component level. Hardware documentation for the off-the-shelf units is provided by the respective manufacturers.

The theory of operation in this manual is organized in these three increasing levels of detail:

- System Overview
- Functional Description
- Detailed Circuit Description

This three-tier approach allows all users (technicians, managers, etc.) to progress through this manual to the technical level of detail desired. That is, systems managers may use only the first or second level of detail to assist in making workstation-related decisions, while technicians may use all levels of detail.

* IBM® is a registered trademark of International Business Machines Corporation.

PS/2™ is a trademark of International Business Machines Corporation.

McIDAS SSEC Display Adapter

SDA Workstation Overview . . . 1
McIDAS System Description . . . 1
Satellite Data Antennas . . . 1
Frame and Bit Synchronizers . . . 3
Archive/Playbacks . . . 3
Multisourcerer . . . 3
Ingestors . . . 3
IBM Mainframe Computer . . . 4
Disk Storage . . . 4
Local Area Networks . . . 4
Remote Communications Controller . . . 5
Protocol Converter . . . 5
Tape Drives . . . 5
Workstation . . . 5
McIDAS SDA Workstation Components . . . 7
IBM PS/2 Computer . . . 7
Keyboard . . . 8
Mouse . . . 8
RAM Expansion Card . . . 8
SDA . . . 9
LAN Adapter . . . 9
Color Monitors . . . 10

SDA Card Functional Description . . . 11
PS/2 Interface . . . 12
Micro Channel Interface . . . 12
Data Bus Transceiver . . . 12
Dynamic RAM . . . 15
Image Storage Requirements . . . 15
Graphics Storage Requirements . . . 15
Memory Partitioning . . . 15
RAM Control . . . 16
Video Output Section . . . 17
Digital to Analog Converters (DACs) . . . 18
Interlace FIFO . . . 19
Vertical Drive and Sync Drivers . . . 20
Interlaced Sync Driver . . . 20
SDA Outputs . . . 20
Master Controller . . . 21
U9 Functional Description . . . 23
U10 Functional Description . . . 27
Cursor Generator . . . 30
Horizontal Start Offset . . . 31
Horizontal Half-Size . . . 31
Vertical Start Offset . . . 32
Vertical Half-Size and Type . . . 32

Detailed Circuit Description 33
Schematic Conventions 33
Logic Conventions 33
PS/2 Interface 34
PALS U1 and U30 (22V10) 34
Data Bus Transceiver (U2) 35
Micro Channel Interface 35
Board Bus and Transceiver Control 35
XCVR 35
POS Registers 37
Programmable Decoder 37
Chip Select Logic 38
Chip Select Feedback "OR" 38
Unused Functions 38
Dynamic RAM (2M Byte) 39
Video Output Section 41
RAMDACs 41
Interlace FIFO 43
Output Drivers 44
Master Controller 45
U9 Detailed Circuit Description 45
Memory Write Latch Strobe Generator 45
Data Bus Interface 46
Graphics Mask and High/Low Word Select 49
Graphics Prioritizer and Processor 50
One-of-Eight Selector 51
Resynchronizing Latch 51
Graphics Demultiplexer 51
Command Latch 52
RAM Timing Generator 53
U10 Detailed Circuit Description 57
Timing Generator 57
Address Strobe Generator 57
RAM Read/Write Addressing Section 58
Address Input Latch 67
RAMDAC Byte Selector 67
Interlace Generator 67
Interlaced Graphics Bus Demultiplexer 67
Interrupt Logic Block 67
Cursor Generator 68

SDA Diagnostics 69
High Level Diagnostics 69
Installing the Diagnostics 70
Running the Diagnostics 71
Deleting the Diagnostics Menus 72
Low Level Diagnostics 74
Installing the Diagnostics 75
Running the Diagnostics 76
Deleting the Diagnostics 80

Supplemental Data 81
PAL Equation Listings 81
Symbols and Abbreviations 82
P22V10 Located at U1 83
P22V10 Located at U30 85
Schematic Drawings 1 - 4
Assembly Drawing 1

Appendix A
PS/2 Architecture A-1
Micro Channel Characteristics A-1

Appendix B
Video Scan Timing Characteristics B-1
Interlaced Timing Characteristics B-1
Progressive Scan Timing Characteristics B-2

Appendix C
Programmable Gate Array Background C-1
Input/Output Block (IOB) C-3
Configurable Logic Blocks (CLB) C-3
Programmable Interconnect Network C-3

Appendix D
PS/2 Micro Channel Control Signals D-1
ADL/ (Address Decode Latch) D-1
CDDS16/ (Card Data Size 16) D-1
SBHE/ (System Byte High Enable) D-1
MADE24 (Memory Address Enable 24) D-2
M/IO (Memory/ Input Output) D-2
S0/ and S1/ (Status bits 0 and 1) D-2
CMD/ (Command) D-2
CDSFDBK/ (Card Selected Feedback) D-2
IRQ15/ (Interrupt Request 15) D-3
CDSETUP/ (Card Setup) D-3
CHRESET (Channel Reset) D-3

Figures, Tables and Timing Diagrams

Figures

Figure 1. Simplified SSEC McIDAS 2
Figure 2. SDA Workstation 6
Figure 3. SDA Functional Block Diagram 13
Figure 4. U9 Functional Block Diagram 22
Figure 5. U10 Functional Block Diagram 26

Figure 6. EPB2001 Functional Block Diagram 36
Figure 7. U9 Detailed Functional Block Diagram 47
Figure 8. RAM Frame Organization Diagram 54
Figure 9. RAM Read/Write Addressing Section 63
Figure C-1. IOB Functional Block Diagram C-2
Figure C-2. CLB Functional Block Diagram C-4
Figure C-3. Interconnect Network C-5

Tables

Table 1. Typical Frame Mix RAM Address Space 16
Table 2. Progressive Scan and Interlaced (TV) Timing Comparisons 18
Table 3. Memory Mapped Address Strobe Generation 28
Table 4. Cursor Address Port Definitions 31
Table 5. Cursor Types 32
Table 6. Memory Data Bus Organization 39
Table 7. RAMDAC Memory Map 42
Table 8. RAMDAC Data Priority 43
Table 9. Memory Write Latch Strobe Generation 46
Table 10. Graphics Mask Truth Table 50
Table 11. One-of-Eight Selector Inputs and Outputs 51
Table 12. Command Latch Mapping 52
Table 13. Zoom Factor Versus Area Magnification 61
Table 14. High Level Diagnostics Files 69
Table 15. SDADIAG.ZIP Modules 70
Table 16. SDA High Level Diagnostics Tests 72
Table A-1. Micro Channel Data Transfers A-2
Table D-1. I/O and Memory Transfer Controls D-2

Timing Diagrams

Timing Diagram 1. RAM Timing (No Frame Load) 55
Timing Diagram 2. RAM TIMING (Frame Load in Progress) 55
Timing Diagram 3. Progressive Scan Timing 59
Timing Diagram 4. Interlaced Scan Timing 59

Installing the SDA Board

The SDA (SSEC Display Adapter) board installation requires:

- one SDA board
- one diskette labeled **SDA Diagnostics**, version 1.1

The SDA Diagnostics diskette contains the SDA High Level and Low Level diagnostics software and the SDA Adapter Description File (ADF).

Use the steps below to install the SDA software and board.

1. Turn off the computer and open the top cover. Insert the SDA board into any unused 32-bit Microchannel Adapter (MCA) slot in your IBM PS/2. Handle the card by its edges, trying not to touch parts or the edge connector.

2. Replace the top cover on the PS/2. Connect the video cable from the VGA monitor to the top plug on the SDA.

3. Insert your Backup Copy of the Reference diskette (not the Option diskette) and turn on the computer. Following the memory check, a 165 error is displayed on the screen, indicating an adapter was added or removed. The system will boot with the Backup Copy of the Reference diskette.

4. To continue,

   Press: **Enter**

   When the system asks if you want to automatically configure the system,

   Press: **N**

   You must choose this option, as the Adapter Description File (ADF) is not yet present on your Backup Copy of the Reference diskette.

5. When the Main Menu is displayed,

   Select: **4** (Copy an Option Diskette)

   Remove the Backup Copy of the Reference diskette and insert the SSEC supplied SDA Diagnostics diskette that contains the SDA ADF file. This diskette is located in the inside front cover pocket of this manual.

6. When prompted, remove the SDA Diagnostics diskette and insert the Backup Copy of the Reference diskette. Then,

   Press: **Enter**

   to copy the ADF file to the Backup Copy of the Reference diskette.

7. When the Main Menu is displayed,

   Select: **2** (Set Configuration)

8. When the Set Configuration Menu is displayed,

   Select: **2** (Run Automatic Configuration)

9. To exit,

   Press: **F3**

   Remove the Backup Copy of the Reference diskette. Then,

   Press: **Enter**

   The system boots with the new hardware configuration and completes the installation of the SDA board.

10. Change to the \MCIDAS\TOOLS directory and run the SETUP program to configure McIDAS. See page 1-11 of the *McIDAS-OS2 Users Guide* for more information on SETUP.

SDA Workstation Overview

The McIDAS SSEC Display Adapter Workstation is the link between the user and McIDAS. It provides the animated display of satellite imagery and weather data. The workstation allows the user to issue commands to the McIDAS mainframe and receive digital image and graphics information from it. The SDA Workstation stores and processes image and graphics data and displays this data on one or more color monitors. Users' systems range from a single McIDAS workstation linked remotely to SSEC's McIDAS, to complete systems like SSEC's McIDAS, to more elaborate systems similar to SSEC's McIDAS but with more or different components.

This overview provides:

- a brief description of SSEC's McIDAS to show how the SDA Workstation fits into the total system
- a description of the components that make up a complete SDA Workstation

McIDAS System Description

Figure 1 on the next page is a simplified block diagram of the SSEC McIDAS. It shows the major data paths into and out of McIDAS and the various methods of linking workstations to McIDAS. Each block or group of blocks is described below in terms of its function in the system.

Satellite Data Antennas

The Satellite Data Antennas provide several types of satellite digital data inputs to McIDAS. These inputs are:

- POES (Polar Orbiting Environmental Satellite)
- GOES (Geostationary Orbiting Environmental Satellite)
- METEOSAT (European Meteorological Satellite)
- GMS (Japan's Geosynchronous Meteorological Satellite)

Each satellite data source has its own antenna system. These antennas receive image and sounder data signals from their respective satellites. These antennas receive signals in the VHF and microwave frequency spectra. Low noise amplifiers, down converters and demodulators associated with each antenna recover a modulation signal consisting of a composite of data and clock.

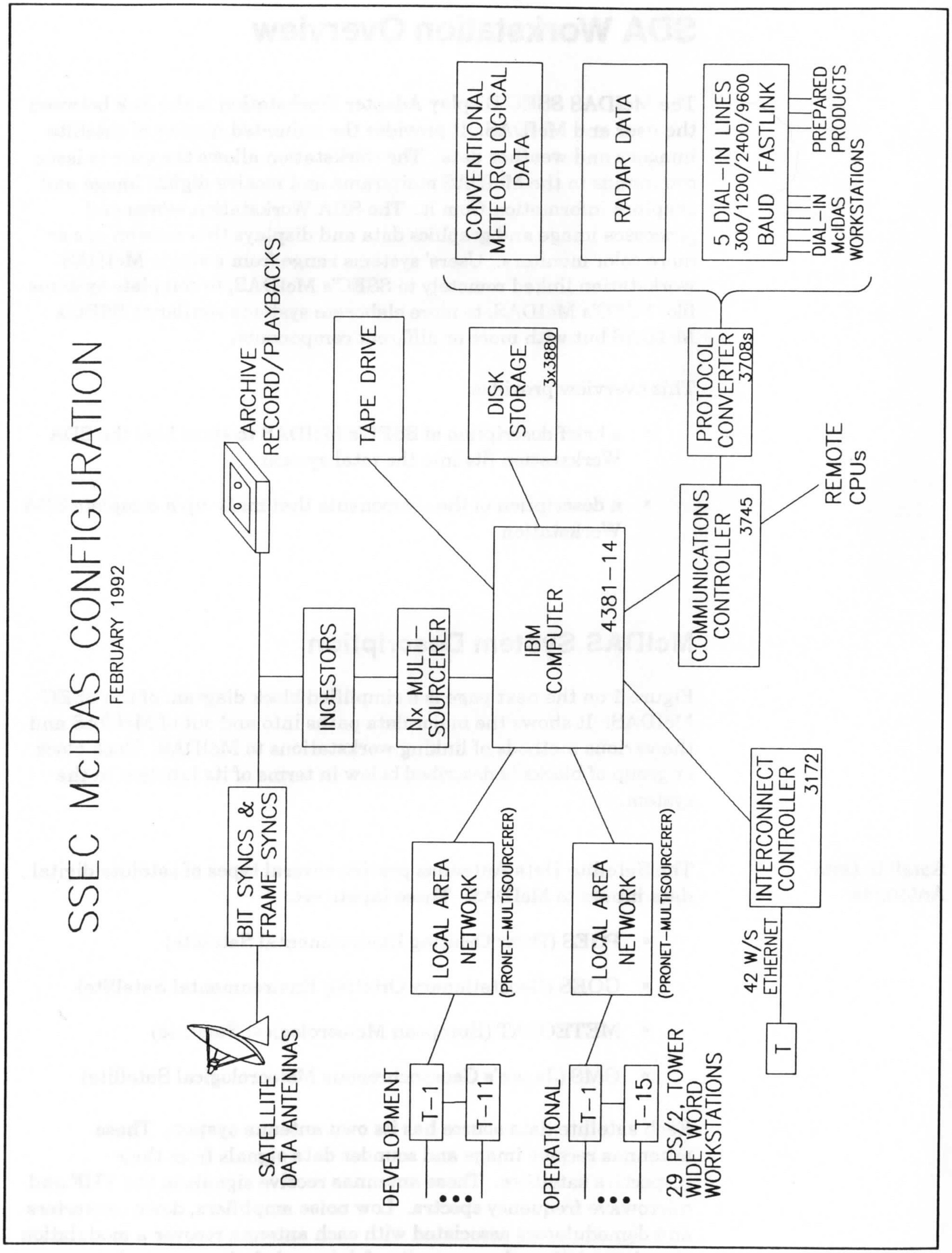


Figure 1. Simplified SSEC McIDAS

Frame and Bit Synchronizers

Each demodulated signal is processed by a Bit Synchronizer (Bit Sync) which separates the signal into a serial data stream and clock signal. The Bit Syncs send their outputs to a respective Frame Synchronizer (Frame Sync) which outputs a serial or parallel data format (depending on the signal type - POES, GOES, METEOSAT, etc.) to the Archive/Playbacks and Ingestors.

Archive/Playbacks

The Frame Syncs' outputs are stored digitally on video cassette cartridges. The playback unit allows previously recorded satellite data to be read into the system at any time for user analysis.

Multisourcerer

IBM mainframe inputs and outputs are via the mainframe's I/O channels. The Multisourcerer is an SSEC designed and built programmable interface between an IBM I/O Channel and up to six external devices (ingestors, etc.). It contains a card cage with an interface controller and six unused Multibus compatible board slots. The following Multisourcerer applications cards are available:

- ingestors
- Local Area Network (LAN) controllers
- synchronous communications controllers

SSEC's McIDAS uses several Multisourcerers because of its complexity. However, a single Multisourcerer may control all three types of applications cards simultaneously in a less complex McIDAS.

Ingestors

Each type of satellite has its own ingestor card. SSEC has developed ingestors for the following satellite systems:

- POES
- GOES Imaging
- GOES Sounding (2-card set)
- METEOSAT
- GMS (Japan's Geosynchronous Meteorological Satellite)
- GOES Mode AAA Auxiliary Block

Ingestors receive data, clock and control signals from their respective Frame Sync or Archive/Playback. Ingestors assemble incoming data into complete scans or blocks of data. Upon command from the mainframe, the ingestors send requested frames or blocks (or parts of frames or blocks) to the mainframe via the Multisourcerer's interface controller. The ingestors relieve the mainframe of much of the computing overhead associated with the ingest process.

IBM Mainframe Computer

The IBM mainframe computer (4381-14) receives imagery data from the ingestors via their Multisourcerer, and radar and conventional meteorological data via the Communications Controller. The mainframe computer contains the operating system, applications programs and subroutines. It functions as a data processor, database management system and data analyzer.

Disk Storage

Currently, SSEC's McIDAS has a storage capacity of approximately 33.8 gigabytes (33.8 billion bytes). The disk storage is divided into 6999 digital areas in which several databases reside. Three of the major databases are:

- image files
- Meteorological Data (MD) files
- grid files

An image (area) file contains digitized satellite visual or infrared sensor data. An MD file is a generic file for single location observations (nonimage). It accommodates many types of data under one general structure. A grid file contains fields analyzed at regularly-spaced latitude and longitude locations (grids) from observational data.

Local Area Networks

Local Area Networks (LANs) link groups of workstations to the mainframe. Figure 1 shows these three LANs:

- Operational
- Ethernet
- Development

These LANs provide high speed data transfers between the mainframe and workstations. No data is transferred between workstations though this is not a LAN limitation.

The Operational network is a ProNET™ * LAN manufactured by Proteon, Inc. The LAN is a two-card set that is plugged into a Multisourcerer. The Multisourcerer is an interface between the LAN cards, which control up to 15 McIDAS workstations, and the IBM computer channel.

* ProNET™ is a trademark of Proteon, Inc.

The Ethernet LAN is interfaced to the mainframe via the Interconnect Controller (IBM 3172). SSEC currently has about 42 workstations on the Ethernet LAN.

The Development LAN is similar to the Operational LAN (ProNET LAN). It is used to develop new hardware.

Remote Communications Controller

The Remote Communications Controller is an interface between the IBM computer and:

- other McIDAS installations
- the Protocol Converter

The Remote Communications Controller allows SSEC's McIDAS to exchange a variety of meteorological data. The Protocol Converter provides several additional types of data inputs and outputs.

Protocol Converter

The Protocol Converter links asynchronous dial-up workstations to the mainframe. It also provides radar and conventional meteorological data inputs to the mainframe. This conventional data, called point source data, is used to maintain the Meteorological (MD) database. Sources for conventional data include radiosondes, rocketsondes, ship reports, aircraft and radar.

Tape Drives

The McIDAS operating system is periodically downloaded onto magnetic tape for upgrading other McIDAS installations.

Workstation

The McIDAS workstation, which is designed for animated display of satellite imagery and weather data, has the following features:

- real-time access to image and conventional data
- graphics overlays of images without image destruction
- animated displays of image or graphics frames at user selectable looping rates up to 15 frames/second
- pseudocoloring of imagery
- user selected graphics colors
- user defined hardware cursor

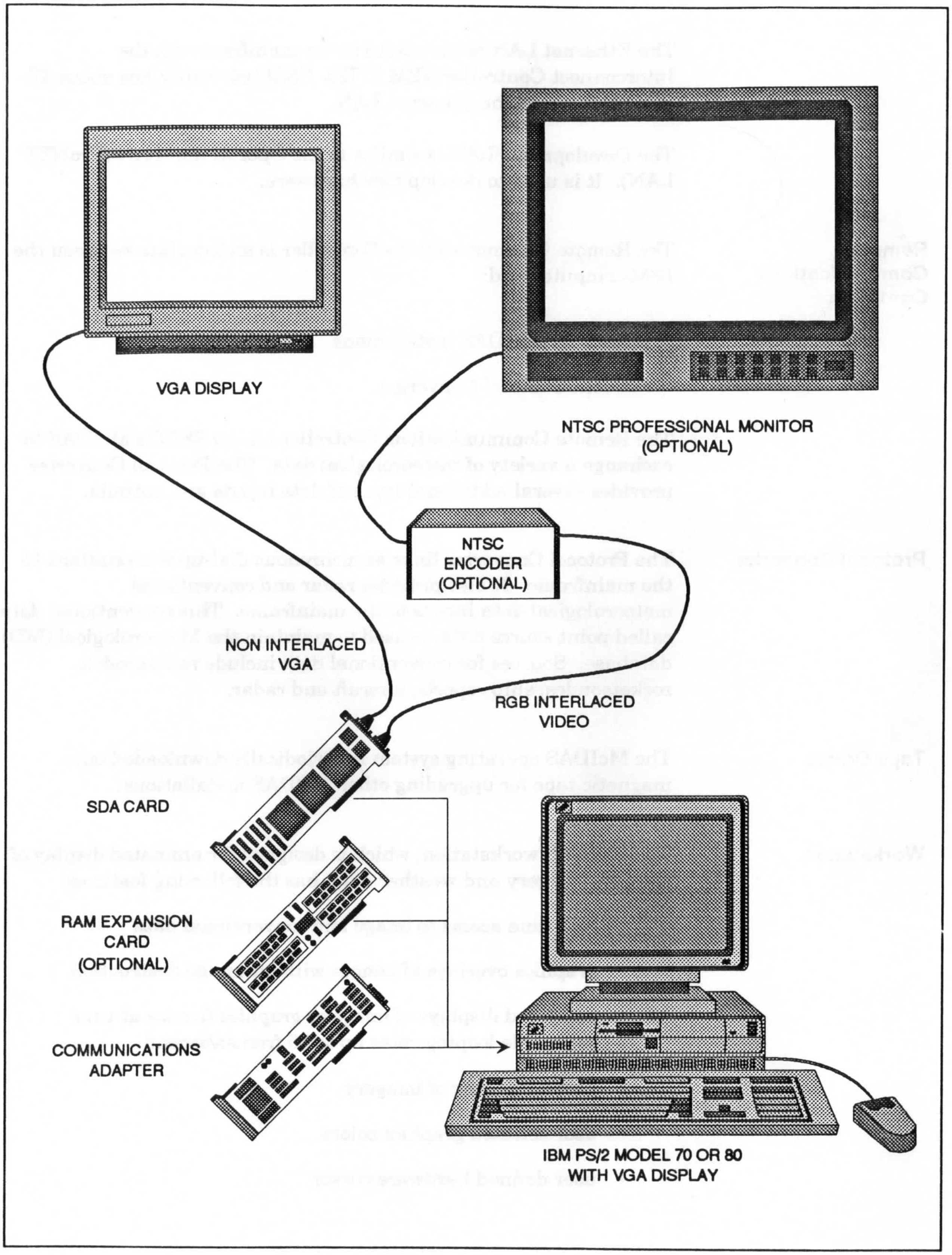

VGA DISPLAY
NTSC PROFESSIONAL MONITOR (OPTIONAL)
NTSC ENCODER (OPTIONAL)
NON INTERLACED VGA
RGB INTERLACED VIDEO
SDA CARD
RAM EXPANSION CARD (OPTIONAL)
COMMUNICATIONS ADAPTER
IBM PS/2 MODEL 70 OR 80 WITH VGA DISPLAY

Figure 2. SDA Workstation

McIDAS SDA Workstation Components

Figure 2 on the adjacent page shows the components that make up a complete SDA Workstation. Blocks containing "OPTIONAL" may be omitted in some workstations, with a corresponding reduction in workstation capability and flexibility.

The IBM PS/2 can be connected to the IBM host via a ProNET or Ethernet LAN link, or an asynchronous link (modem) to the host's Communications Controller. Regardless of the link type, there is a bidirectional data path between the IBM host and the PS/2.

The IBM PS/2 system provides the workstation intelligence and contains the following components:

- IBM PS/2 Model 70 or 80 computer
- keyboard
- mouse
- RAM Expansion Card
- SDA
- LAN Adapter
- color monitors (Zenith Model ZCM-1492 or equivalent)

Each of these components is described below.

IBM PS/2 Computer

Most SDA Workstation installations are based on the IBM PS/2 Model 70. The main difference between the Model 70 and 80 is the number of card adapter slots. The Model 70 has three adapter slots; the Model 80 has eight. Typically, the SDA, a LAN Adapter and, optionally, a Memory Expansion card are installed in the Model 70's three slots. The LAN Adapter provides the Model 70 with a LAN link to the host. The PS/2's built-in asynchronous serial port can be connected to a modem if an asynchronous link to the host is desired. A built-in parallel port can be used to drive an optional parallel line printer, and the RAM Expansion card provides up to eight megabytes of optional memory.

The PS/2 integrates the keyboard, mouse and color monitor into a user interactive workstation. The PS/2 relieves the host of some of its workload by performing many of the workstation related tasks at the workstation. This is in contrast to some of the older style non-PC based workstations.

Keyboard

The keyboard is a standard keyboard supplied with the IBM PS/2 computer. The user interacts with McIDAS via the PS/2's keyboard to perform the following:

- request image and/or graphics frames from the host
- establish display loops
- select zoom factors
- roam within an image
- position the cursor
- colorize images
- define graphics colors
- perform high and low level diagnostics

The keyboard is documented by its manufacturer.

Mouse

The mouse provides the PS/2 with cursor control. It is used to position the video cursor, roam within a frame and position the cursor (flashing underline symbol) during alphanumeric displays on the color monitor.

RAM Expansion Card

The SDA Card contains sufficient on-board RAM to store up to six image frames. These frames can be displayed at animation looping rates of up to 15 images per second. If loop lengths greater than six images are required, additional images can be stored in PS/2 RAM. Because these additional frames must be downloaded into the SDA card to be displayed, the maximum looping rate decreases to six images per second (worst case). The RAM Expansion Card provides eight megabytes of additional memory.

SDA

The SDA is an IBM PS/2-compatible Micro Channel™ * adapter printed circuit board designed and built by SSEC. It performs all TV display functions. Specifically, it:

- converts the requested frame stored in its on-board memory as 64-bit digital words, to horizontal scan lines of analog video
- produces all necessary clocking and TV timing signals (horizontal and vertical sync plus blanking) required for displaying TV images
- colorizes the images
- overlays up to four graphics planes on the image data
- outputs the analog data in progressive scan and interlaced RGB (Red, Green and Blue) formats
- allows the user to set the zoom factor, enable or disable graphics bit planes, load colorizer tables, roam within a selected image and select the image frame to be displayed
- generates a cursor in response to PS/2 commands

The progressive scan output is the standard output. However, the same data stream that drives the progressive scan output also drives an interlace converter that produces an interlaced RGB output for driving an optional NTSC Encoder. The NTSC Encoder produces a composite NTSC output for use in TV studios, etc., and driving the optional NTSC monitor. The interlace converter can be configured to generate the European TV format in lieu of the US TV format.

LAN Adapter

Workstations located within a few hundred feet of the host can be connected to the host via a Local Area Network (LAN). SSEC supports two LAN types, ProNET and Ethernet. Either LAN requires an access adapter for each network (workstation or host) member. The host's adapter resides in a Multisourcerer which links the LAN adapter to the IBM channel. The workstation's adapter resides in the PS/2. The LAN Adapter card shown in Figure 2 is specific to the LAN being used. If the LAN is ProNET, this card is a ProNET Adapter Model P1800. Since the OS/2 environment supports several sources of Ethernet adapter cards, the LAN Adapter can be one of several Ethernet Adapter Cards if the LAN is Ethernet.

* Micro Channel™ is a trademark of International Business Machines Corporation

Color Monitors

The SDA Workstation uses two 14" Zenith Model ZCM-1492 color monitors or equivalent. One is used for PS/2 command and control functions; the other displays the image and graphics outputs of the SDA board. These monitors require 0.7V RMS red, green and blue drive signals, and composite sync from their respective drive sources. These monitors are non-interlaced and have horizontal and vertical scan frequencies of 31.5 KHz and 60 Hz, respectively. Other monitors meeting these specifications may be substituted.

If an optional NTSC monitor is used, it requires an NTSC encoder. This encoder is driven by the interlaced RGB and interlaced composite sync outputs of the SDA card.

SDA Card Functional Description

The SDA Workstation is a collection of commercial devices and the SDA board. Documentation for the commercial devices is furnished by their respective manufacturers and is not repeated here. The SDA is documented in detail in this manual. The documentation consists of this Functional Description and a Detailed Circuit Description.

Much of the SDA control logic is contained in two Xilinx™ * Programmable Gate Array™ * chips. These chips contain the equivalent of 18,000 logic gates. Hundreds of pages of documentation could be dedicated to the functional description of these two chips without necessarily resulting in more efficient corrective maintenance. Since there are only four components in the control section, substitution of the suspected component with a known good component may be the most effective troubleshooting procedure. Therefore, documentation for these components provides sufficient functional information and timing diagrams to allow you to understand the interaction between the components in the control section and between the control section and the remaining SDA card sections. This should allow you to diagnose control section problems to one of the four chips.

It would be difficult, if not impossible, to understand the SDA board from a functional or circuit perspective without at least a cursory understanding of the PS/2 Micro Channel architecture. Appendix A provides basic information on the PS/2 architecture. For additional Micro Channel information, consult the manufacturer's documentation.

The SDA is built on an IBM PS/2 Micro Channel Adapter form factor board. Refer to Figure 3, the SDA Functional Block Diagram, on page 13. This board consists of a PS/2 Interface, two megabytes of dynamic RAM, a Video Output Section and a Master Controller. The extensive functionality designed into the SDA board is primarily the result of using two Programmable Gate Array chips for the Master Controller.

*Xilinx™ and Programmable Gate Array™ are trademarks of Xilinx Incorporated.

PS/2 Interface

The PS/2 Interface section makes the remainder of the SDA card appear as memory. This section consists of the Micro Channel Interface block and the Data Bus Transceiver.

Micro Channel Interface

The Micro Channel Interface consists of a programmable (EPROM) custom logic IC specifically designed and manufactured by the Altera Corporation to interface the IBM Micro Channel to a PS/2 add-on card (called *adapter* by IBM), i.e., the SDA.

Address lines A24 - A31 select the 16M byte memory block in which the SDA address space is located. Address lines A21 - A23 further specify a 4M byte address block within the 16M byte range. Thus, A21 - A31 specify the 4M byte address space within the 4G byte address capability of the PS/2. This allocation space information is stored in the Micro Channel Interface block. The PS/2 Interface section monitors the Micro Channel to determine if an address on the address bus is intended for the SDA card. When an SDA memory address is detected by the Micro Channel Interface block, it asserts one or more inputs to the Master Controller.

PS/2 addresses may be either memory or I/O type. The Micro Channel Interface uses the PS/2's cycle, and bus-cycle status input lines to determine whether the address is a memory read or write, or an I/O read or write. Only memory write operations are used by the SDA card. When a valid memory write address is detected, the PS/2 Interface asserts MEMW/. This signal functions as a write control signal for the Master Controller.

The SDA accepts display control data from the PS/2 and synchronizes it during the vertical blanking period. This prevents the PS/2 data from causing interference in the video display. The Master Controller asserts IRQ11/ or IRQ15/ (Interrupt Request) at the start of each progressive scan frame/field pair. The PS/2 uses the interrupt to drive the loop rate. Upon interrupt, the PS/2 sends video display control data to the SDA which it uses to display the next frame.

Data Bus Transceiver

The Data Bus Transceiver interfaces the Micro Channel's data bus to the Micro Channel Interface and Master Controller blocks. This transceiver reduces the Micro Channel loading caused by these two blocks to one unit load. The transceiver passes PS/2 data inputs to the Micro Channel Interface and Master Controller blocks and passes POS (Programmable Option Select) register status from the Micro Channel Interface to the PS/2 (DD0 - DD7). Via the Master Controller, the PS/2 writes image and graphics data to the RAM and color enhancement data to the Video Output Section (OD0 - OD15). Currently, no data passes from the Master Controller to the PS/2.

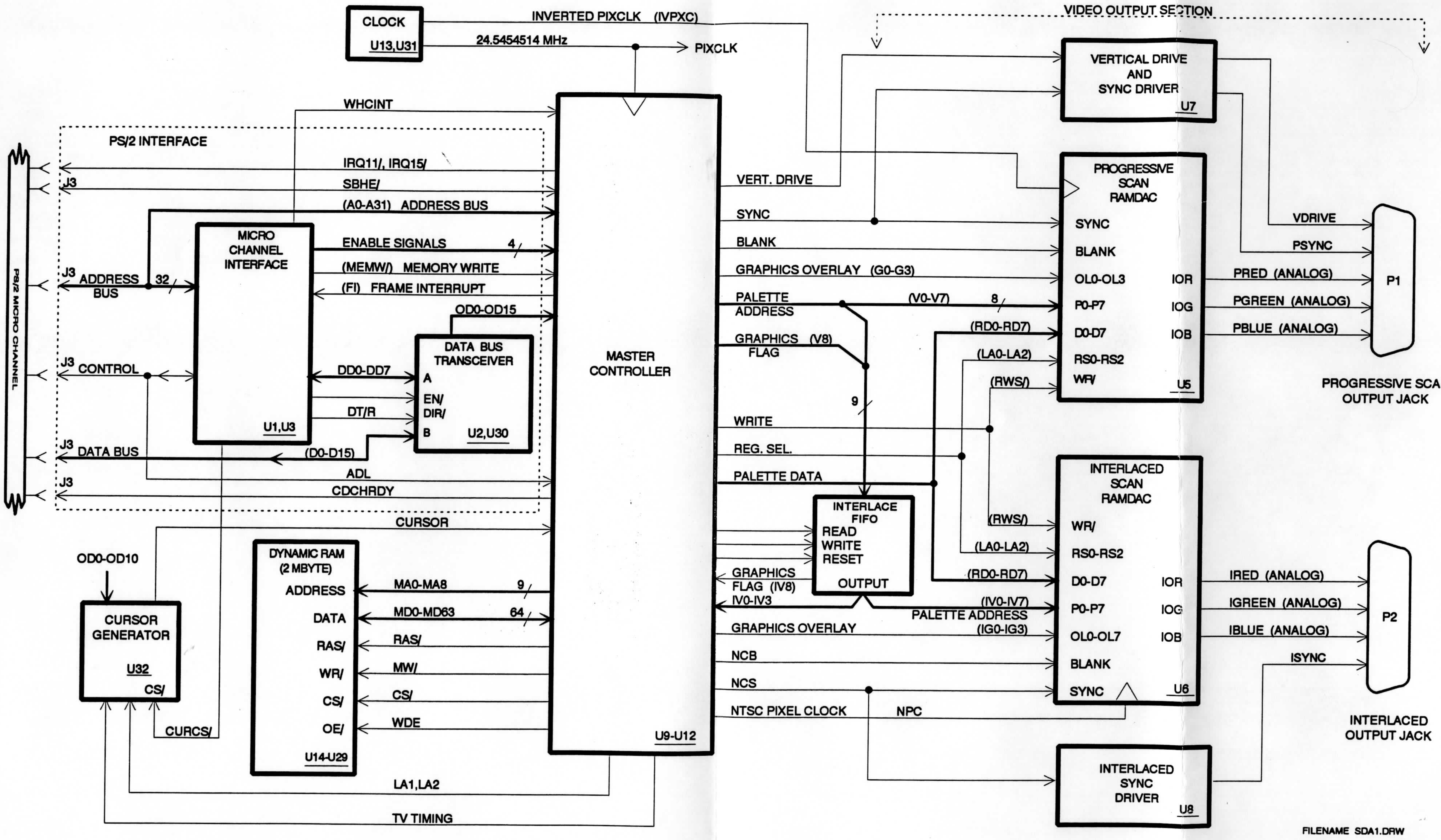


Figure 3. SDA Functional Block Diagram

Dynamic RAM (2M Bytes)

The SDA card contains a 2M byte dynamic RAM memory; actual storage capacity is 2,097,152 bytes. The memory is organized as 262,144 (40000_H) 64-bit words (2M bytes). It stores the image and graphics frames.

Image Storage Requirements

A McIDAS TV frame is defined as 480 horizontal lines of 640 pixels each. In the SDA Workstation, each image pixel is represented by 8 bits. Since each word contains 64 bits, they each represent 8 pixels. Therefore, a full resolution SDA Workstation horizontal line requires 80 words (80 words x 8 pixels = 640 pixels). A full resolution TV frame requires 38,400 words, (80 words/line x 480 lines = 38,400 words). This is the storage requirement for each stored image frame.

Graphics Storage Requirements

Graphics frames require one-half as much storage as image frames because only four bits are used for each pixel. Thus, four graphics planes are available (each bit represents a graphics plane). Each graphics byte represents two graphics pixels; each 64-bit RAM word represents 16 graphics pixels. Each graphics frame requires 19,200 64-bit RAM words.

Memory Partitioning

The RAM is arbitrarily partitioned into a mix of image and graphics frames. The image and graphics frame size is fixed at 38,400 and 19,200 words, respectively. Typically, there are six image frames and one graphics frame. Thus, 249,600 RAM words are used (6 x 38,400 plus 1 x 19,200 = 249,600). If additional graphics frames are desired, two additional graphics frames are available for each image frame sacrificed. Thus, there can be as many as 13 graphics frames if no image frames are needed. Regardless of the mix, the memory always contains 249,600 active words and 12,544 unused words.

The graphics frame(s) can be physically located before, between or after the image frames. Thus, for the typical mix (six image frames and one graphics frame), there are seven possible start addresses for the graphics frame and two possible start addresses for each image frame. Table 1 on the next page shows the possible start addresses.

Image Frame	Image Space if Graphics Start Address is Above Image Start Address	Image Space if Graphics Start Address is Below Image Start Address
1	0 - 38,399	19,200 - 57,599 *
2	38,400 - 76,799	57,600 - 95,999
3	76,800 - 115,199	96,000 - 134,399
4	115,200 - 153,599	134,400 - 172,799
5	153,600 - 191,999	172,800 - 211,199
6	192,000 - 230,399 **	211,200 - 249,599

* Graphics are located at 0 - 19,199

** Graphics are located at 230,400 - 249,599

Table 1. Typical Frame Mix RAM Address Space

RAM Control

The Dynamic RAM consists of sixteen 262,144-word by 4-bit dynamic RAM chips connected to form a 262,144-word by 64-bit memory. Each RAM chip contains the following inputs/outputs:

- RAS/ (Row Address Strobe)
- CS/ (Chip Select)
- OE/ (Output Enable)
- WR/ (Write)
- A0-A8 (Address bits 0-8)
- I/O1-I/O3 (Input/Output data bits 0-3)

Except for the data pins, the corresponding pins of all chips are connected in parallel. The data pins are separate. Thus, there is one RAS/ input, one CS/ input, one OE/ input, one WR/ input, a 9-bit address input and a 64-bit data input/output port.

Internally, each chip is organized as a 512 by 512 by 4-bit array. These chips require a 9-bit row address and a 9-bit column address to address the array. The Master Controller places the row address (least significant 9 bits of the 18-bit address) on the address bus and strobes RAS/ to latch these bits into the RAM chips. Then, it places the column address on the address bus and strobes CS/ and OE/ (read) or WR/ (write).

The Dynamic RAM requires periodic refresh cycles to maintain data integrity. The memory's chips must receive refresh cycles at a minimum rate of 512 refreshes per 8 milliseconds. The refresh cycles are part of the display data retrieval cycle. This cycle is 16 pixel periods in length (approximately 640 nanoseconds). It consists of a *graphics word* fetch followed by an *image word* fetch followed by a *write opportunity* and concluded with another *video word* fetch. These four cycle phases are equal in duration (approximately 160 nanoseconds). A refresh cycle is executed during the *write opportunity* phase if there is no data to be written in the Dynamic RAM.

A horizontal line consists of 640 visible pixels and 140 blanked pixels. There are 40 potential refresh cycles during the visible portion. During horizontal blanking, only *write opportunity* cycles are executed, resulting in an additional 35 *write opportunities*. This results in about 75 potential refresh cycles per horizontal line. Since there are 255 horizontal lines per 8 milliseconds, there are about 19,125 refresh opportunities per 8 milliseconds or approximately 40 times the minimum refresh requirements (512 per 8 milliseconds). At maximum data throughput from the PS/2 to the SDA card, it's unlikely that half of the *write opportunity* cycles would be used for writing data into the Dynamic RAM. Under this condition, there would still be 20 times more refresh cycles than necessary.

Video Output Section

The Video Output Section generates two RGB analog outputs from the 8-bit binary inputs. One output is a progressive scan format; the other is an interlaced format. Appendix B at the end of this manual provides a description of the differences and similarities between these two formats. The following description assumes you are familiar with the formats.

The SDA card uses the same video and graphics data to produce the interlaced and progressive scan drives. Comparing the two sets of timing characteristics in Table 2 below suggests a simple approach for converting the progressive scan drive to the interlaced format.

Fields/ Frame	H.Lines Field	H.Scan Time	V.Scan Time	H.Blank	V.Sweep
TV 2	262.5/ Field	63.55µs	59.94Hz	17% H.Scan	21/Field 42/Frame
PS* 1	525/ Frame	31.78µs	59.94Hz	17% H.Scan	42/Frame

* PS = Progressive (noninterlaced) scan. Note, PAL timing is different but the frame versus field relationships remain the same.

Table 2. Progressive Scan and Interlaced (TV) Timing Comparisons

Table 2 suggests that buffering every other horizontal line of the progressive scan data and writing that data at one-half the progressive scan horizontal rate, will generate one TV field (262.5 even or odd lines) during a progressive scan frame. If this process is repeated during the next progressive scan frame, the other TV field is generated, completing the interlaced TV frame.

The SDA card currently outputs interlaced data in the RGB format. Since the RGB interlaced output meets the NTSC timing standards, it can be converted to the composite NTSC format using a standard external NTSC encoder. The SDA card can also be reconfigured to output RGB interlaced data in the PAL timing standards which can be converted to the composite PAL format using a standard external PAL encoder. PAL is the European interlaced TV standard.

Digital to Analog Converters (DACs)

Each of the two DACs contains an integral video color palette RAM and a graphics overlay color palette RAM. This type of DAC is called RAMDAC™ * by its manufacturer, the Brooktree Corporation.

After the palettes are programmed via the PS/2 Interface and the Master Controller sections (RD bus in Figure 3), these RAMDACs use the 8-bit video input (P0 - P7 port in Figure 3) to internally address one of 256 24-bit video color palette RAM words. The 24-bit word consists of three bytes of magnitude data, one byte each for RED, GREEN and BLUE. The DAC simultaneously converts its three bytes of magnitude data to analog. Thus, the RAMDACs convert the 8-bit inputs to preassigned values of red, green and blue analog outputs.

* RAMDAC™ is a trademark of Brooktree Corporation.

The RAMDACs have a 4-bit graphics overlay color palette RAM addressing input (OL port - see Figure 3). If any bit in this input is high (on), the 256-word by 24-bit video color palette RAM is switched off and a 15-word by 24-bit graphics overlay color palette RAM is switched on.

The Cursor Generator (described later) asserts CURSOR whenever a the current pixel being painted is a cursor pixel. CURSOR is an input to the Master Controller which sets all four graphics bits high if CURSOR is asserted. Thus, if CURSOR is asserted, the Master Controller writes F_H to the OL port. The 24-bit data stored in this overlay address determines the cursor's color. The Master Controller allows only these five graphics overlay addresses:

- **1_H (graphics bit plane one - lowest priority)**
- **2_H (graphics bit plane two)**
- **4_H (graphics bit plane three)**
- **8_H (graphics bit plane four)**
- **F_H (cursor bit - highest priority)**

Thus, on a pixel-by-pixel basis, the cursor has priority over all other data, and graphics data has priority over image data. However, from the RAMDAC's perspective, the cursor is treated as graphics overlay data. Thus, any references to graphics in the Video Output Section descriptions also pertain to the cursor.

Interlace FIFO

The Interlace FIFO in Figure 3 is a 9-bit by 512-word First-In First-Out buffer. It buffers every other progressive scan horizontal line. Eight-bit Input data is written at the pixel clock rate, but the data is read at one-half the pixel clock rate. Immediately after loading the 640th pixel of a progressive scan line, the FIFO contains 320 stored pixels. The other 320 pixels were read out during the unload process. Since only every other progressive scan is stored, the FIFO is emptied just prior to the start of the next used progressive scan. Data reading and writing are controlled by the Read and Write control signals generated by the Master Controller.

V0 - V7, the data inputs to the FIFO, are multiplexed video and graphics. The Graphics Flag, a Master Controller output, indicates whether the current data on the V0 - V7 bus is video or graphics data. If the current pixel data is graphics, only the lower four input bits carry valid data and the Graphics Flag bit is a one; if the current pixel data is video, all eight input bits are valid and the Graphics Flag bit is zero. The Graphics Flag is stored as the ninth bit in the FIFO.

The FIFO's eight output bits (IV0 - IV7) drive the P0 - P7 port on the RAMDAC. In addition, IV0 - IV3 are also sent to the Master Controller along with the output Graphics Flag (FIFO's ninth bit). If the output Graphics Flag is high, the Master Controller routes the IV0 - IV3 inputs to the IG0 - IG3 outputs, respectively. Thus, the Master Controller demultiplexes the graphics from the multiplexed V0 - V7 bus. When the output Graphics Flag is high, the V4 - V7 inputs and the IV4 - IV7 outputs are invalid because the lower four video bits were replaced by the graphics bits in the multiplexing process. This causes no problem, however, because the RAMDACs' entire video color palette RAM is switched off whenever any overlay bit input is high.

Vertical Drive and Sync Drivers

The Sync Driver is driven by SYNC, the progressive scan composite sync output of the Master Controller. The Sync Driver provides the PSYNC (Progressive composite SYNC) output to P1, the Progressive Scan Jack. The Vertical Drive Driver provides the VDRIVE (Vertical DRIVE) output to P1. VDRIVE is required by some progressive scan monitors.

Interlaced Sync Driver

Separate composite sync and blanking signals are generated for the Interlaced Scan RAMDAC. In addition, the Interlaced Sync Driver, driven by NCS (NTSC Composite Sync), provides composite interlaced sync (ISYNC) to the Interlaced Output Jack.

SDA Outputs

The interlaced RGB and sync outputs can be converted to a broadcast compatible composite signal by a suitable NTSC or PAL converter. The converter's composite output signal can drive an NTSC or PAL monitor, VCRs, TV broadcast transmitter equipment, etc. The progressive scan output drives a progressive scan monitor such as the Zenith Model ZCM-1492 color monitor.

Master Controller

The Master Control block in Figure 3 consists of four ICs that contain the equivalent of several thousand logic gates. This block:

- generates all display timing (sync, blanking, etc.)
- assembles 16-bit PS/2 words into 64-bit RAM words
- disassembles 64-bit RAM words into image bytes or graphics nibbles
- modifies image and graphics data read rates as a function of the zoom factor
- modifies the image and graphics start location as a function of roaming
- generates the Interlace FIFO read, write and reset strobes
- generates all RAMDAC control signals
- interfaces the OD data bus to the RAMDAC RD bus during RAMDAC programming and palette loading
- generates all RAM control signals and refresh cycles
- provides wait states, if necessary, during PS/2 to SDA RAM transfers

The Master Controller consists of two Programmable Gate Arrays. Appendix B in this manual provides a brief background on the Xilinx Programmable Gate Array technology to show its functional capabilities. For additional information on these devices, refer to the *Programmable Gate Array Data Book* by Xilinx Incorporated.

The Master Controller consists of two Programmable Gate Array Logic devices (U9 and U10) and their associated programming PROMs (U11 and U12, respectively). Though the two gate arrays are interconnected to function as a single master controller, each gate array performs specific functions. In the following description, the two gate arrays are referred to by their printed circuit board references, U9 and U10. Figures 4 (page 22) and 5 (page 26) show the functional equivalent of U9 and U10, respectively.

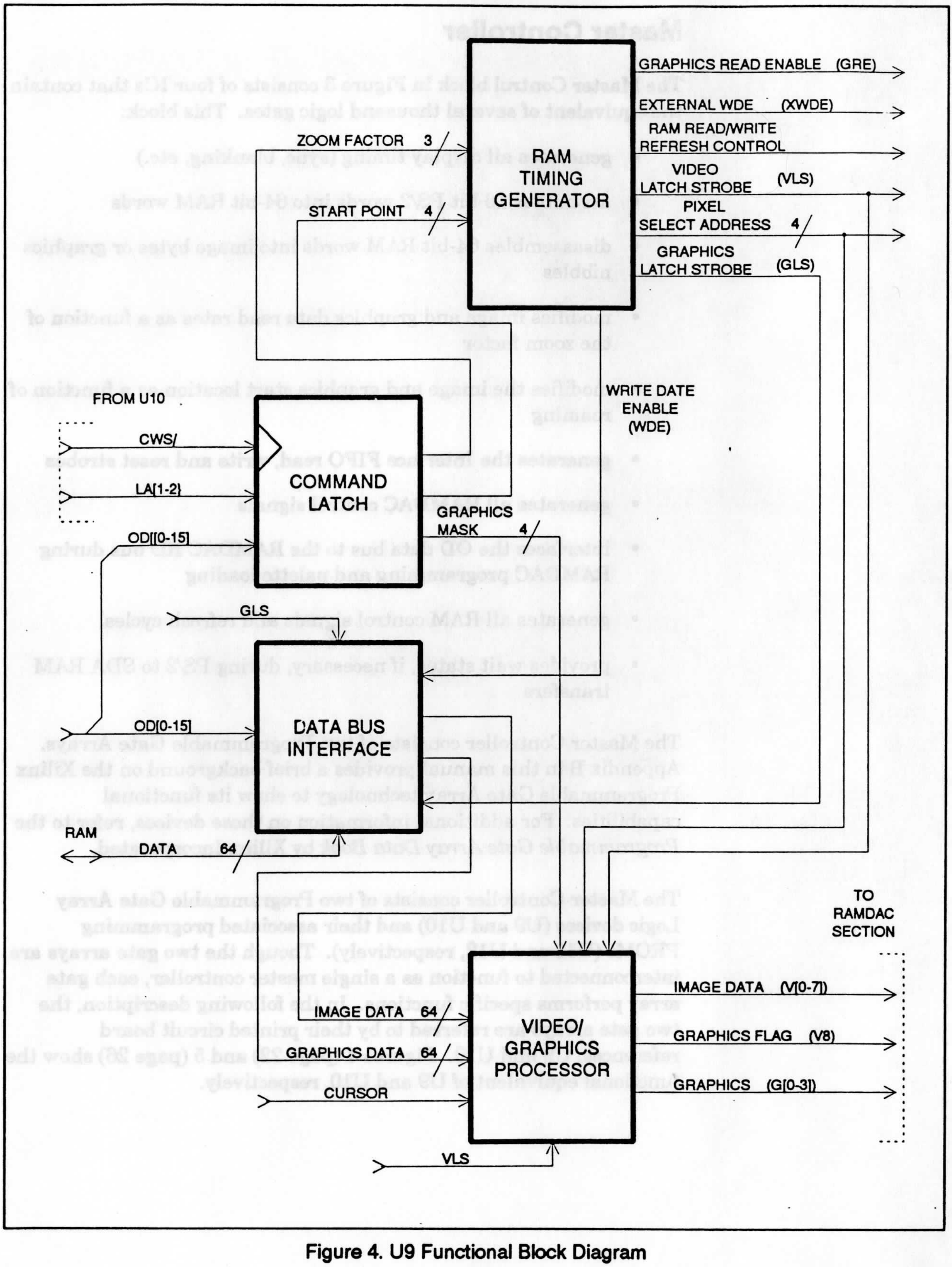


Figure 4. U9 Functional Block Diagram

U9 Functional Description

Refer to Figure 4 on the adjacent page. U9 consists of these functional sections:

- Data Bus Interface
- RAM Timing Generator
- Command Latch
- Video/Graphics Processor

Data Bus Interface

The Data Bus Interface provides a data write path between the PS/2 and the Dynamic RAM, and a data read path between the Dynamic RAM and the Video/Graphics Processor.

The PS/2 writes 16-bit data to the Data Bus Interface which latches four consecutive PS/2 data write transfers to form one 64-bit data word. Upon receipt of the fourth transfer, it writes the 64-bit word to the Dynamic RAM.

The 64-bit data read from the Dynamic RAM may be either graphics or image data. Graphics data is latched and output to the Video/Graphics Processor via the 64-bit Graphics Data bus. Image data is output directly (unlatched) to the Video/Graphics Processor via the 64-bit Image Data bus.

RAM Timing Generator

The RAM Timing Generator:

- generates all Dynamic RAM read, write, address multiplexer controls and chip selects
- performs the Dynamic RAM refresh cycles
- generates the video and graphics latch strobes
- decodes and latches the pixel select addresses
- generates RAM access cycle control signals such as WDE (Write Data Enable), GRE (Graphics Read Enable - used by U10), and XWDE (External Write Data Enable - used by U10)

Dynamic RAM reading, writing and refresh occurs within a fixed RAM access timing cycle. The diagram below shows this cycle. Dynamic RAM refreshing is accomplished during the Write Opportunity phase if there is no data to be written into the Dynamic RAM. Each phase in the diagram below is four pixel clock periods in duration (about 160 nanoseconds).

<------------------------------ 16 Pixel Periods ------------------------------>

Graphics Read	Image Read	Write Opportunity *	Image Read

* RAM refresh is performed during this time if there is no Write data ready for transfer to the RAM.

The Video and Graphics Latch Strobes (VLS and GLS, respectively latch the 64-bit image and graphics data words, respectively. The graphics latches are located in the Data Bus Interface; the image latches are part of the Video/Graphics Processor.

The Pixel Select Address is a 4-bit address that points to the pixel within a 64-bit word address. Only the three LSBs are required to address the image byte (0-7 range). However, all four bits are required to select the current graphics nibble (0-15_D range).

The first image or graphics pixel in a horizontal line does not always start with the first byte or nibble in a 64-bit word. As an example, suppose the SDA is operating in a zoom factor of 4. Remember that a zoom factor of 1 requires eighty 8-byte (640 bytes total) image words or forty 16-nibble graphics words to display a horizontal line of video or graphics, respectively. A zoom factor of 4 requires only 160 image bytes or graphics nibbles because each pixel (byte or nibble) is repeated four times. Thus, twenty 8-byte image or ten 16-nibble graphics words are required if each byte or nibble is used. Suppose the user roams one pixel to the right. To accomplish this, the memory read logic will retrieve the first word, but the first pixel (byte or nibble) must be discarded. This is done by using the Start Point input to the RAM Timing Generator to preset the Pixel Select Address to a count of one. This causes it to point at the second image byte and graphics nibble. The SDA will display that pixel and each of the remaining pixels in the first word four times each. Each pixel of the next 19 image and/or 9 graphics words is displayed four times each. At this point, 636 pixels have been painted. To display the last four pixels, the next word must be read (twenty-first image word and/or eleventh graphics word); only the first pixel is used and replicated four times.

To summarize, the Start Point address is used only to preset the Pixel Select Address counter to the pixel address of the first pixel in the first word of each horizontal line. For the remainder of each line, the Start Point has no effect. The total number of bytes or nibbles used per horizontal line is always 640/(Zoom Factor). Finally, some pixels in the first and last words fetched for a horizontal line may not be used when roaming.

Command Latch

The SDA requires the following control information before it can display an image:

- Relative Start Address (19 bits)
- Zoom Factor (3 bits)
- Video Frame Start Address (9 MSB bits)
- Graphics Frame Start Address (9 MSB bits)
- Graphics Mask (4 bits)

The above information is called *command data* and is packed into four concatenated 16-bit words.

The division of functions between U9 and U10 causes both chips to require some or all of the command data. Currently, the PS/2 writes this data to addresses XXE00008_H - XXE0000F_H (subject to change, "XX" is defined by A24 - A31), causing the Micro Channel Interface chip to assert CWD/ (Command Word). If CWD/ is a result of a 16-bit memory write, U10 asserts CWS/ (Command Word Strobe). CWS/ is used internally by U10 and is also exported to U9.

CWS/ strobes the Command Latch in U9. LA1 - LA2 (second and third LSB latched Address bits) identify the data on the OD bus. Internal latches, qualified by LA1 and LA2, latch the appropriate OD bus bits on the trailing rising edge of CWS/.

The RAM Timing Generator uses the Zoom Factor to determine the pixel replication number, and thus, the Dynamic RAM read rate. It uses Start Point to determine the start pixel in the first word of each horizontal line.

The Graphics Mask is required by the Video/Graphics Processor to determine which graphics bit planes are displayed.

Video/Graphics Processor

The Video/Graphics Processor disassembles 64-bit image and graphics words into bytes and nibbles, respectively, and transfers them to the RAMDACs at the pixel clock rate.

The Pixel Select Address input points at the current image and graphics pixel data within the respective 64-bit word; the Graphics Mask input determines which graphics bit planes are active. The Cursor input causes the Video/Graphics Processor to set Graphics bits G0 - G3, the Graphics Flag, and Image Data bits V0 - V3 high.

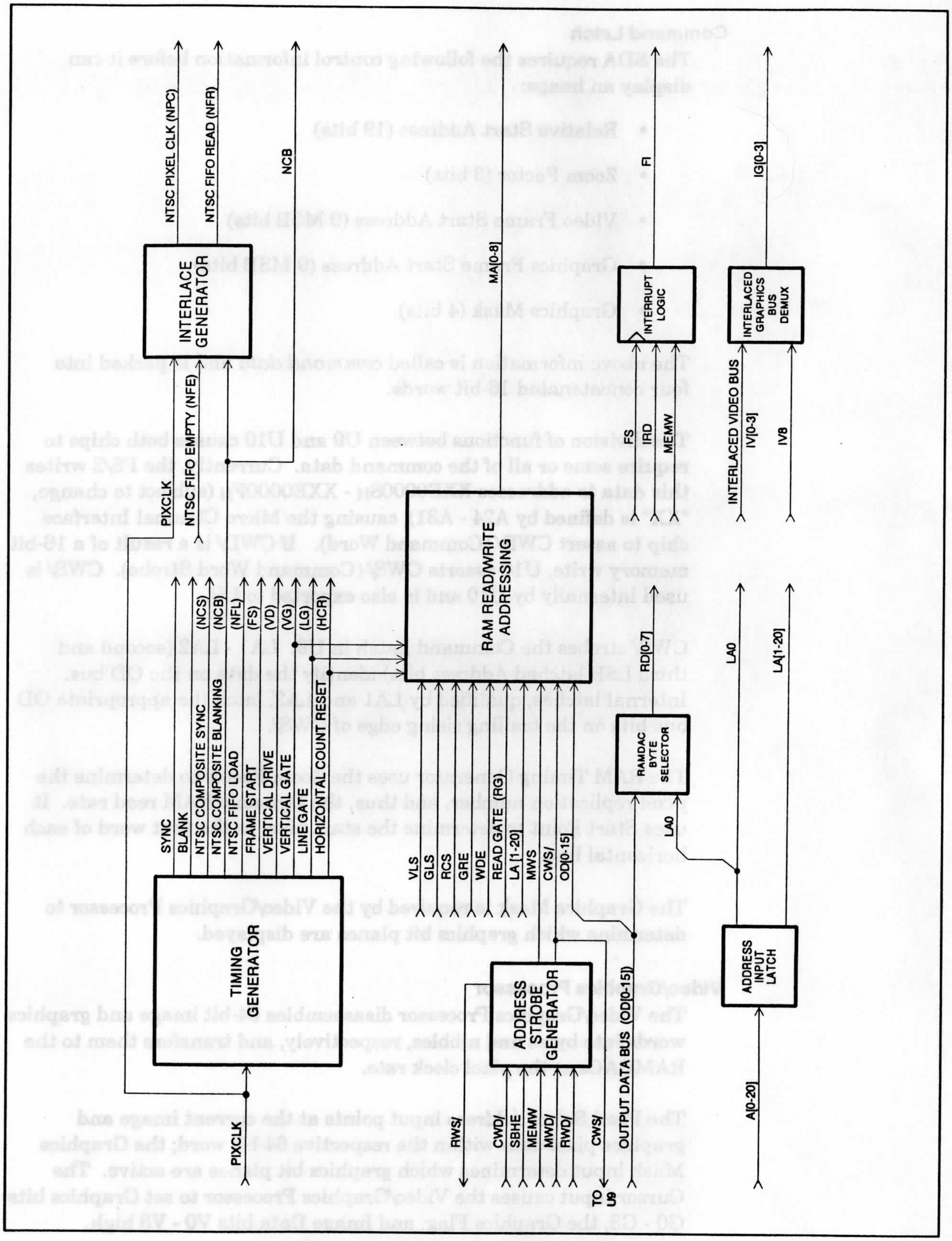


Figure 5. U10 Functional Block Diagram

U10 Functional Description

The U10 Functional Description is based on Figure 5 shown on the adjacent page. U10 contains the following functional blocks:

- Timing Generator
- Address Strobe Generator
- RAM Read/Write Addressing
- Address Input Latch
- RAMDAC Byte Selector
- Interlace Generator
- Interlaced Graphics Bus Demultiplexer
- Interrupt Logic

Timing Generator

The Timing Generator produces all display related timing signals. The Composite Sync (SYNC) and Composite Blanking (BLANK) are used by the Progressive Scan RAMDAC. The NTSC Composite Sync (NCS) and NTSC Composite Blanking (NCB) are used by the Interlaced Scan RAMDAC. Other timing signals are:

- Horizontal Count Reset (HCR/)
- Line Gate (LG)
- NTSC FIFO Load (NFL - partially externally generated)
- Vertical Gate (VG)
- Vertical Drive (VD)
- Frame Start (FS)

Horizontal Count Reset (HCR/) is active low during the 779th pixel clock period in each horizontal line. It resets the horizontal counter that determines the length of a horizontal line. This counter, clocked by PIXCLK (Pixel Clock), counts from 0 to 779 before being reset (780 pixel clock periods). HCR/ is also used by the RAM Read/Write Addressing section of U10 and the RAM Timing Generator in U9.

Line Gate (LG) is active during the 640 visible pixels within the 780-pixel horizontal line.

NTSC FIFO Load (NFL) clocks image data into the NTSC FIFO which is loaded with alternate lines of progressive scan image data at the pixel clock rate. NFL is inactive during every other horizontal line; it consists of 640 PIXCLK pulses during active horizontal lines. The 640 NFL clock pulses coincide with the active portion of LG.

Vertical Gate (VG) is used only by the RAM Read/Write Addressing section. It is high during the visible portion of the vertical field.

Vertical Drive (VD) is low during lines 7 - 12 of each progressive scan. Because VD is required by some monitors, it is output to the Video Output section which outputs it to P1 (see Figure 3) via a line driver.

Progressive frames are paired for interlacing. Frame Start (FS) identifies the first line in a 1050-line group. For the progressive scan, these lines make up two consecutive fields or frames. Frame 0 consists of lines 0 - 524; frame 1 consists of lines 525 - 1049. The even lines from frame 1 (lines 0, 2, 4, ... 522 and 524) make up the even interlaced field. The even lines from frame 1 (lines 526, 528, 530, ... 1046 and 1048) make up the odd interlaced field. Note that by using 1050 lines as the frame base, it is not possible to pair up two adjacent 525-line frames incorrectly because FS resets the interlace logic at the beginning of each frame pair.

Address Strobe Generator

The Address Strobe Generator generates these three strobes:

- Memory Write Strobe (MWS/)
- Command Write Strobe (CWS/)
- RAMDAC Write Strobe (RWS/)

Each is assigned a unique address space. Access of a strobe's address space results in a respective memory mapped chip select output from the Micro Channel Interface chip. The respective chip select can be valid for 8- or 16-bit data words. The Address Strobe Generator restricts its output strobes to memory write operations only. In addition, it allows the respective strobe to be generated only if the data bus word size is valid for the logic being strobed. Table 3 below summarizes the strobe generation.

Strobe	Input Chip Select	Address Space *	Data Bus Word Size
RWS/	RWD/	XXBFFFF0$_H$ - XXBFFFF7$_H$	08-bit
MWS/	MWD/	XXC00000$_H$ - XXDFFFFF$_H$	16-bit
CWS/	CWD/	XXE00008$_H$ - XXE0000F$_H$	16-bit

* XX is defined by address bits A24 - A31 when running the configuration diskette.

Table 3. Memory Mapped Address Strobe Generation

MWS/ writes data into the Dynamic RAM; RWS/ writes data into the RAMDACs; CWS/ writes display parameters to the RAM Read/Write Addressing section.

RAM Read/Write Addressing

The RAM Read/Write Addressing section is an automatic RAM read address generator and a RAM write address latch. To generate RAM read addresses, this section requires the following five parameters:

- Image Frame Start Address
- Graphics Frame Start Address
- Relative Start Address within the frame
- Graphics Mask
- Zoom Factor

Once the above parameters are passed by the PS/2, the RAM Read/Write Addressing section automatically computes the image and graphics start address for each scan line. At the start of a frame and the end of each scan line thereafter, these line start addresses are downloaded into their internal image and graphics address counters. Once the counters are preloaded, they are incremented by VLS (Video Latch Strobe - clocks the video address counter) or GLS (Graphics Latch Strobe - clocks the graphics address counter).

Four PS/2 write transfers are required to transfer 64 data bits to the SDA's RAM (eight bytes per RAM word). The RAM Read/Write Addressing section latches the output of the Address Input Latch during the fourth data transfer and shifts this address three places to the right (divides by eight) to produce the RAM's word address.

Address Input Latch

The Address Input Latch latches A0 - A20 from the PS/2 on the rising trailing edge of ADL/ (Address Latch) forming LA0 - LA20. LA0 is used by the RAMDAC Byte Selector only. LA1 - LA2 are used in the RAM Read/Write Addressing section to decode the input parameters, by U32 (Cursor Generator) and by the Command Latch in U9 to decode the Start Point, Zoom Factor and graphics Mask parameters. LA3 - LA20 address the Dynamic RAM during data write operations.

RAMDAC Byte Selector

The RAMDAC Byte Selector is a 16-line to 8-line selector. When the PS/2 transfers single byte data, as required by the RAMDACs, the even addressed bytes are transferred on the lower half of the OD bus and the odd addressed bytes are transferred on the upper half. Thus, the RAMDAC Byte Selector selects the lower byte of the OD bus (OD0 - OD7) to drive the RD bus when A0 is low; it selects the upper byte of the OD bus (OD8 - OD15) to drive the RD bus when A0 is high.

Interlace Generator

The Interlace Generator generates the Interlaced Scan RAMDAC's input clock (NPC - NTSC Pixel Clock) and the Interlace FIFO read strobes (NFR - NTSC FIFO Read). Both signals are one-half the pixel clock rate. NPC is required continuously by the Interlaced Scan RAMDAC; NFR is allowed only during the visible pixel portion of each visible line. NPC is generated by applying PIXCLK to a toggle-type flip-flop. NFR is generated by gating NPC with interlaced composite blanking (NCB).

Interlaced Graphics Bus Demultiplexer

The Interlace FIFO output drives the Interlaced Graphics Bus Demultiplexer. The 9-bit output data from this FIFO consists of an 8-bit multiplexed video/graphics bus and the Graphics Flag. The multiplexed output of the FIFO, along with the FIFOed Graphics Flag, drives the Interlaced Graphics Bus Demultiplexer. This demultiplexer uses the Graphics Flag to gate the lower four bits of the FIFOed image data onto the Interlaced Graphics bus (IG0 - IG3).

Interrupt Logic

The SDA interrupts the PS/2 at the beginning of each interlaced frame (progressive frame pair) to inform it that it can now accept new display parameters. The Interrupt Logic uses FS (Frame Start) to set a D-type latch. FI (Frame Interrupt), the output of this latch, is inverted and applied to two tristate buffers whose outputs are IRQ11/ and IRQ15/. IRQ11/ is selected as the PS/2 interrupt input if WHCINT (Which Interrupt) is low; IRQ15/ is selected if WHCINT is high. The PS/2 sets the level of WHCINT when running the configuration diskette. The PS/2 clears the interrupt logic by writing to address XXBFFFF8$_H$. The SDA uses this address as an interrupt acknowledge.

Cursor Generator

The Cursor Generator outputs CURSOR which is a flag that goes high if the current pixel being scanned is part of the cursor. The generator creates one of five user selectable cursor shapes (type). The user can set the horizontal and vertical size parameters of the selected cursor type from a minimum of 5 pixels by 3 pixels to a maximum of 640 pixels by 480 pixels (horizontal and vertical, respectively).

The generator consists of an addressable Programmable Logic Device that contains counters, latches, comparators and control logic. It contains four 16-bit addressable memory ports that store cursor size, shape and type parameters. The ports are addressed by LA1 and LA2 (Latched Address bits 1 and 2, respectively) when enabled by CURCS/ (active low). CURCS/ is asserted by the Master Controller for addresses XXEFF000$_H$ - XXEFF007$_H$. Table 4 below defines the input ports and their addresses.

LA2	LA1	CURCS/	Address *	Function
0	0	0	XXEFF000$_H$	Horizontal Start Offset
0	1	0	XXEFF002$_H$	Horizontal Half-Size
1	0	0	XXEFF004$_H$	Vertical Start Offset
1	1	0	XXEFF006$_H$	Vertical Half-Size, Mode
Y *	Y *	1	non-cursor	Don't Care

Y = Don't Care

* The upper 8 bits of the address (XX) are defined during system configuration.

Table 4. Cursor Address Port Definitions

OD0 - OD10 provide the parameter data to the Cursor Generator. Thus, the maximum number of bits defined for each 16-bit port is 11.

Horizontal Start Offset

The Horizontal Start Offset parameter is defined as:

Bit Position 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Undefined	OS	Horizontal Start Offset

Where OS (bit 10) is the Off-Screen Flag. It is set if any part of the cursor is off the left edge of the screen. The 10-bit start offset provides a horizontal offset range of 0 to 1024 pixels (referenced to the start of the horizontal line) though there are only 640 visible pixels.

Horizontal Half-Size

The Horizontal Half-Size parameter is 9 bits and is defined as:

Bit Position 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Undefined	Horizontal Half-Size

The cursor is always twice as wide as the value of the Horizontal Half-Size parameter.

Vertical Start Offset

The Vertical Start Offset parameter is 11 bits and is defined as:

Bit Position	15 14 13 12 11	10	9 8 7 6 5 4 3 2 1 0
	Undefined	OS	Vertical Start Offset

Where OS (bit 10) is the Off-Screen Flag. It is set if any part of the cursor is off the top edge of the screen. Bits 0 - 9 define an offset of 0 - 1024 lines. This allows the cursor to start in either field of a frame.

Vertical Half-Size and Type

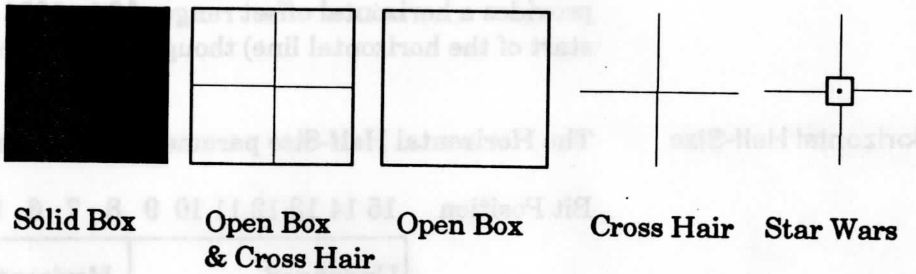
The Vertical Half-Size and Type parameters are defined as:

Bit Position	15 14 13 12 11	10 9 8	7 6 5 4 3 2 1 0
	Undefined	Type	Vertical Half-Size

Where Type (bits 8 - 10) defines the five cursor types shown below.

OD10	OD9	OD8	Type Code	Cursor Type
0	0	0	0	Open Box
0	0	1	1	Solid Box
0	1	0	2	Star Wars
0	1	1	3	Off
1	0	0	4	Open Box
1	0	1	5	Off
1	1	0	6	Cross Hair in Box
1	1	1	7	Cross Hair

Table 5. Cursor Types



Detailed Circuit Description

The schematic diagrams of the SDA card are shown on SSEC drawing 6450-0639 (Revision F, dated 06/02/92). This Detailed Circuit Description discusses the logic of each block shown in the functional block diagram in Figure 3 on page 13. These include the:

- PS/2 Interface
- Dynamic RAM (2M Bytes)
- Video Output Section
- Master Controller
- Cursor Generator

Refer to Figure 3 and the schematics as necessary.

Schematic Conventions

The SDA is built on an IBM Micro Channel Adapter form factor printed circuit board. Locations on the board are described by the silk screened component location symbol (U1 through U30). The Detailed Circuit Description that follows refers to ICs by these location symbols.

The Micro Channel Interface portion of the PS/2 Interface section and the Master Controller section consists of five chips. Therefore, the circuit description for these sections consists of additional functional levels of detail coupled with timing diagrams where possible.

Logic Conventions

Logic signals are indicated by all uppercase letters and numbers, e.g., DEN. A logic signal name ending with a trailing slash represents an active low signal, e.g., CWD/. Several conventions that can describe the state of a logic signal are true or false, high or low, one or zero, and active or inactive. In the following description, all logic states are described as high and low. This convention best describes the physical condition of a logic signal and is better suited for troubleshooting.

Frequently, a logic signal is "asserted." If a signal with a trailing slash is asserted, it is low. If a signal without a trailing slash is asserted, it is high. Thus, asserted means that a signal is in its active state.

PS/2 Interface

An understanding of the PS/2 Interface requires at least a cursory understanding of the IBM Micro Channel control signals. Refer to Appendix C for a brief description of these signals. For more complete information on the IBM Micro Channel, refer to the *IBM Personal System/2 Model 80 Technical Reference.*

Refer to sheet 1 of the schematics. The PS/2 Interface consists of the following four components:

- U1 and U30 (22V10 PAL® *) - Data Bus Interface
- U2 (74F245) - Data Bus Transceiver
- U3 (EPB2001J) - Micro Channel Interface

PALs U1 and U30 (22V10)

U30 functions as a data bus interface buffer from the PS/2 to U9 and U10 for the lower eight bits of the PS/2's data bus (D0 - D7). U1 performs the same function for the upper eight bits of the data bus (D8 - D15). U1 also functions as a self latch for the CDSETUP/ Micro Channel control signal.

The PS/2 asserts CDSETUP/ during adapter board programming which is part of the power-up sequence. It asserts CMD/ while Micro Channel data is valid. When both signals are asserted simultaneously, U1 asserts GSETUP/ (Setup cycle and data is valid). Once asserted, GSETUP/ remains asserted while CMD/ is invalid (hold term). The need for latching this signal was discovered after the Micro Channel Interface chip (U3) was designed and manufactured and is described in an engineering application note.

In addition to functioning as a data bus interface for D0 -D7, U30 also generates FLF (FIFO Load Flag). FLF is valid during the unblanked portion of the even numbered visible horizontal lines. U10 ANDs FLF with pixel clock to produce NFL (NTSC FIFO Load).

* PAL® is a registered trademark of Monolithic Memories Inc.

Data Bus Transceiver (U2)

The Data Bus Transceiver (U2) interfaces the lower half of the Micro Channel data bus (D0 - D7) to U3, the Micro Channel Interface chip. U2 must be bidirectional because the PS/2 writes setup data to U3 and reads status and adapter ID data from U3.

Micro Channel Interface

The Micro Channel Interface consists of Altera Corporation's EPB2001 IC. It provides all the essential functions to interface a PS/2 add-on card (i.e., SDA card) to the Micro Channel Bus. Some of the EPB2001's functions are not required by the SDA card. However, the SDA card's PS/2 interface is still in full compliance with IBM's Micro Channel interface specifications because the functions that are not used by the SDA card are optional. Figure 6 on the next page is a functional block diagram of the EPB2001. The shaded blocks in Figure 6 have programmable elements that allow them to be customized for a wide variety of applications.

Board Bus and Transceiver Control

The Board Bus portion of the Board Bus and Transceiver Control block is an implementation of Table D-1, located in Appendix D. This section either generates the outputs shown in Table D-1 or generates POS (Programmable Option Select) register read/write controls. Table D-1 is implemented if CDSETUP/ is high (inactive); POS read/write strobes are generated if CDSETUP/ is low. Thus, the POS registers are accessed only during the adapter card setup procedure which is part of the Power-On System Test (POST).

The EPB2001 is designed to be connected to the Micro Channel's data bus via a data bus transceiver. The Transceiver Control portion of the Board Bus and Transceiver Control block provides the data bus transceiver output enable (DEN/) and direction controls (DT/R). DT/R is high during Micro Channel read cycles and low during Micro Channel write cycles. DEN/ is active for a valid read or write cycle for about the same period of time as CMD/.

The Transceiver Control portion of the Board Bus and Transceiver Control block also controls the Transceiver block (XCVR in Figure 6) by providing it with a tri-state enable and a direction control.

XCVR

The XCVR (Transceiver) block interfaces the lower eight bits of the Micro Channel data bus to the POS Registers.

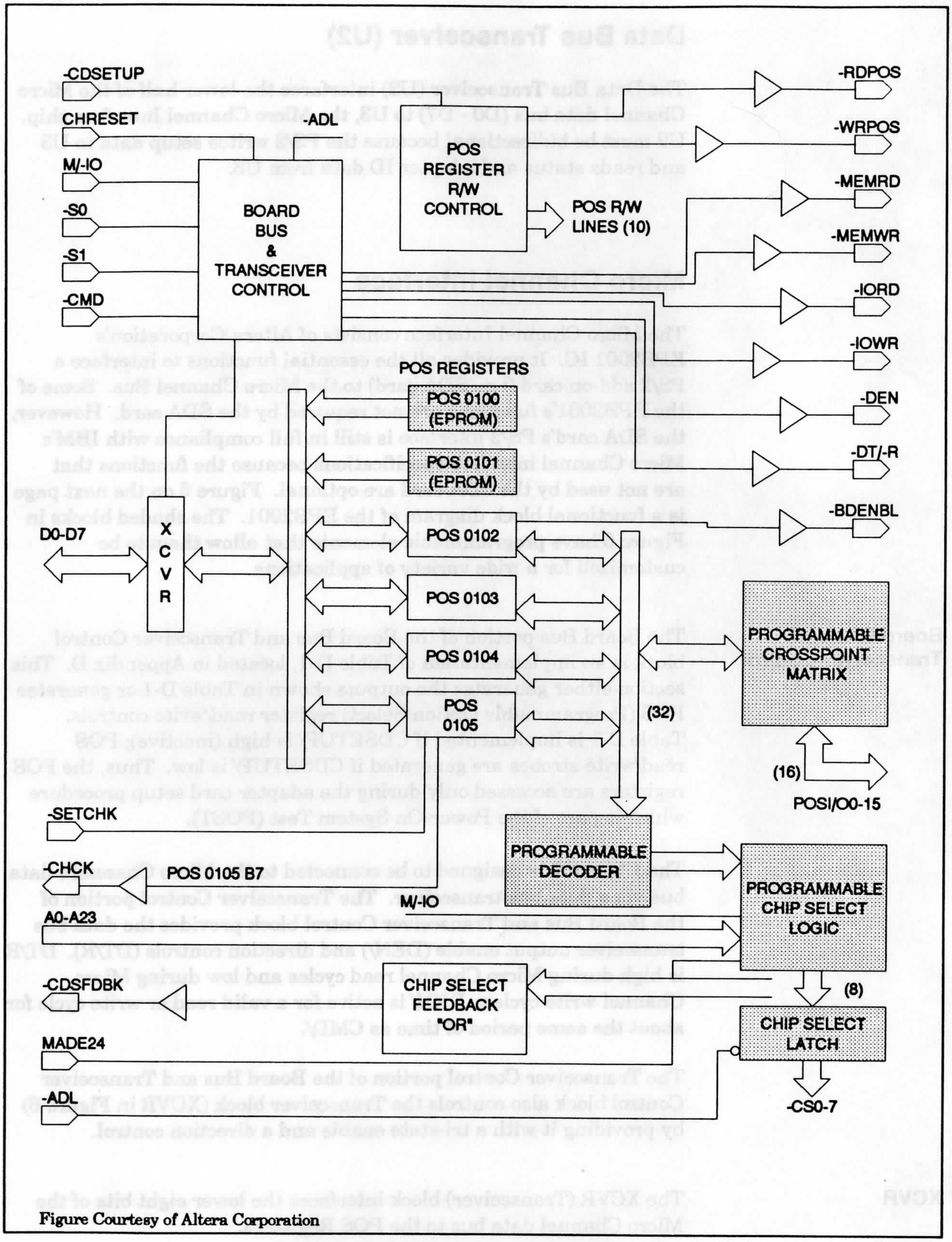


Figure Courtesy of Altera Corporation

Figure 6. EPB2001 Functional Block Diagram

POS Registers

The six 8-bit POS (Programmable Option Select) Registers, labeled POS 0100 - POS 0105, are accessed by reads and/or writes to I/O ports 0100_H - 0105_H. Nonvolatile POS Registers 0100 and 0101 (EPROM) contain the card ID. The PS/2 reads this ID during the adapter card configuration sequence that occurs during the power-on sequence.

Three bits of the remaining four registers are special function bits. Bit 0 of POS Register 0102 functions as an adapter card enable bit. It is reset by CHRESET or by the PS/2's microprocessor writing a zero to this bit during a CDSETUP/ cycle. When this bit is zero, the SDA card responds only to setup reads and writes. When this bit is set to a 1 by a setup bus write cycle, the card is enabled and bit 0 of register 0102 cannot be accessed by normal I/O bus cycles. The BDENBL/ pin reflects the state of this bit.

Bits 6 and 7 of register 0105 are the remaining two special purpose bits. They are not used by the SDA card and are not described here.

The remaining 29 bits of registers 0102 - 0105 function as control inputs to the Programmable Decoder. The upper four bits of POS 0103 define outputs to U10. The chart below describes these bits.

Bit	Name	Function
7	D/P/1	Reconfigures U9 - Used during card checkout only
6	D/P/2	Reconfigures U10 - Used during card checkout only
5	WCHINT	Selects IRQ11/ if low, selects IRQ15/ if high
4	INTEBL/	Enables the selected interrupt line to the PS/2

Programmable Decoder

The EPB2001 provides eight programmable chip select outputs. The Programmable Decoder is part of this chip select logic. It consists of sixty-four 32-input AND gates, eight AND gates for each chip select. Any POS Register 0102 - 0105 bits can be connected to any AND gate. Thus, any POS Register bit or combination of POS Register bits can be selected to qualify any AND gate. The 64 outputs are inputs to the Chip Select Logic block.

Chip Select Logic

The Chip Select Logic block consists of eight identical logic blocks, one for each chip select. M/IO, A0-A23, MADE24, ADL/ and eight Programmable Decoder outputs are inputs to each logic block. Each logic block contains eight programmable address range blocks. The outputs of each range block are enabled by one of the Programmable Decoder outputs. A High Speed Comparator compares the selected range block output with the address currently on the A0-A23 lines. If the address falls within the selected range (including M/IO and MADE24), this logic block's chip select is asserted. Thus, specific POS Register combinations can enable up to eight address ranges for each chip select.

Chip Select Feedback "OR"

Any or all active low chip select outputs of the Programmable Chip Select Logic can be ORed together by the Chip Select Feedback "OR" block to form an active low CDSFDBK/ signal. This output serves as a "cycle acknowledge" signal to the PS/2.

Unused Functions

The remaining functions in U3 are optional Micro Channel functions that are not used in the SDA application. For additional information on the EPB2001 IC, refer to pages 118 - 135 of the *Altera Data Book*, published September, 1988.

Dynamic RAM (2M Byte)

The Dynamic RAM consists of the components shown on sheets 4 and 5 of the schematics (U14 through U29).

The 64-bit data bus consists of these four 16-bit banks labeled A - D:

- MDA (Memory Data A)
- MDB (Memory Data B)
- MDC (Memory Data C)
- MDD (Memory Data D)

Table 6 describes the organization of the data bus.

Chip	Bus Bits	Respective Pins
U14	MDA0 - MDA3	6, 7, 3, 4
U15	MDA4 - MDA7	6, 7, 3, 4
U16	MDA8 - MDA11	6, 7, 3, 4
U17	MDA12 - MDA15	6, 7, 3, 4
U18	MDB0 - MDB3	6, 7, 3, 4
U19	MDB4 - MDB7	6, 7, 3, 4
U20	MDB8 - MDB11	6, 7, 3, 4
U21	MDB12 - MDB15	6, 7, 3, 4
U22	MDC0 - MDC3	6, 7, 3, 4
U23	MDC4 - MDC7	6, 7, 3, 4
U24	MDC8 - MDC11	6, 7, 3, 4
U25	MDC12 - MDC15	6, 7, 3, 4
U26	MDD0 - MDD3	6, 7, 3, 4
U27	MDD4 - MDD7	6, 7, 3, 4
U28	MDD8 - MDD11	6, 7, 3, 4
U29	MDD12 - MDD15	6, 7, 3, 4

Table 6. Memory Data Bus Organization

The four signals that control Dynamic RAM reading and writing are:

- WDE (Write Data Enable)
- CS/ (Chip Select)
- RAS/ (Row Address Strobe)
- MW/ (Memory Write)

WDE drives the OE/ input of each chip (pin 1) and controls the direction of the chip's bidirectional data port. When WDE is low, the data port is in the output mode (reading); when WDE is high, the port is in the input mode (writing).

CS/ drives the CS/ pin on each chip (pin 2). CS/ must be low during reading or writing operations. That is, the combination of CS/ and WDE determine whether the chip is in a read, write or tri-state mode.

RAS/ strobes the 9-bit row address (nine LSBs of the 18-bit RAM address) into the RAMs.

MW/ writes the I/O bus data into the RAM. WDE must be high during the storage process.

Video Output Section

The Video Output Section is shown on sheet 2 of the schematics. Refer to Figure 3 and the schematics as necessary.

RAMDACs

The RAMDACs, located at U5 and U6, produce the progressive and interlaced RGB analog outputs, respectively. In the SDA application, each RAMDAC functionally consists of:

- a 256-word by 24-bit video lookup table
- a 15-word by 24-bit graphics lookup table
- three 8-bit D/A converters
- two address registers
- bus control logic

The PS/2 writes 8-bit words to the RAMDACs via U10 (part of the Master Controller). Logic in U10 interfaces the PS/2's data bus to the RD bus (RD0 - RD7) which drives the D0 - D7 inputs (pins 8 - 15, respectively) of both RAMDACs.

The RAMDACs' D0 - D7 ports are bidirectional. However, in the SDA application, only data writing is performed. The D0 - D7 port programs the 256-word video lookup table, the 15-word graphics lookup table and two address registers (color palette address and overlay palette address). Internal control of the D0 - D7 port is accomplished via the RS0 - RS2 inputs as described below.

As shown in Table 3 in the U10 Functional Description section (page 28), RAMDAC writing is memory mapped to addresses XXBFFFF0$_H$ - XXBFFFF7$_H$. U3 asserts RWD/ (RAMDAC Word Decode) when the PS/2 accesses this address space. U10 asserts RWS/ (RAMDAC Write Strobe) when RWD/ is the result of an 8-bit Memory Write cycle. RWS/ drives the WR/ (Write) pin on the RAMDACs (pin 16).

RS0 - RS2 (Register Select inputs), pins 17 - 19, respectively, address several internal RAMDAC control registers which control the D0 - D7 port. RS0 - RS2 are driven by the three LSBs of the PS/2's address bus (LA0 - LA2, respectively). Thus, in conjunction with RWS/, each internal register is memory mapped to a specific address. Table 7 below describes the memory mapping.

Address *	Function
XXBFFFF0$_H$	Video Palette Address Register (Write)
XXBFFFF1$_H$	Video Palette RAM Programming Data
XXBFFFF2$_H$	not used in SDA
XXBFFFF3$_H$	not used in SDA
XXBFFFF4$_H$	Overlay Address Register (write)
XXBFFFF5$_H$	Overlay Palette Programming Data
XXBFFFF6$_H$	reserved
XXBFFFF7$_H$	not used in SDA

* XX is defined when running the system configuration diskette.

Table 7. RAMDAC Memory Map

The Video Palette RAM (video lookup table) address range is 00 - FF$_H$ (0 - 256 decimal). To program this 256-word by 24-bit palette, begin by writing the desired starting address to BFFFF0$_H$. If the entire palette is to be programmed, as done during startup, this address will be 00$_H$. Once the starting address is loaded, write three 8-bit values to address XXBFFFF1$_H$. The RAMDAC treats these consecutive inputs as red, green and blue values. Upon receiving the blue data, the RAMDAC concatenates the three data bytes to form one 24-bit word and writes it to the address specified in the address register. Then it increments the address in the address register. Thus, to programmed consecutive addresses, only the start address is required.

The Overlay Palette (graphics lookup table) has a range of 1 - F$_H$. Programming the Overlay Palette is similar to programming the Video Palette RAM. First, write the palette address to BFFFF4$_H$. Then write three 8-bit values to BFFFF5$_H$ to form a 24-bit palette word. The main difference between programming this palette and the Video Palette RAM is that only five of the 15 possible cells are loaded. These cell locations are 1, 2, 4, 8 and F (hex). Locations 1, 2, 4 and 8 represent the colors assigned to graphics bit planes 0, 1, 2 and 3, respectively; location F represents the color assigned to the cursor.

P0 - P7 address the Video Palette RAM. For the Progressive Scan RAMDAC, this input port is driven by V0 - V7. U9 (part of the Master Controller) drives this bus with the current pixel's brightness magnitude. Thus, data addresses one of the cells in the Video Palette RAM. The RAMDAC converts the three bytes in the addressed cell to analog red, green and blue outputs. The P0 - P7 inputs of the Interlaced Scan RAMDAC are driven by V0 - V7 via the Interlace FIFO (described below).

OL0 - OL3 address the Overlay Palette. The Master Controller drives this bus with 4-bit graphics/cursor data. The OL bus is prioritized. If the current pixel being painted is part of two or more graphics planes but is not part of the cursor, the Master Controller allows only the highest priority graphics bit to drive the OL bus. If the current pixel being painted is part of the cursor, all OL bits are set high, regardless of whether the pixel is also a graphics pixel. If one or more OL bus lines are high, the outputs of the Video Palette RAM are disabled and the Overlay Palette cell addressed by OL0 - OL3 drives the D/A converters. If all OL bits are low, the Overlay Palette output is disabled and the Video Palette RAM output is enabled. Thus, graphics data has priority over video data. Table 8 below shows the RAMDAC data priority.

OL Bus Value (hex) *	Priority	RAMDAC Output
0	0 (lowest)	Image Data
1	1	Graphics Bit Plane 1
2	2	Graphics Bit Plane 2
4	3	Graphics Bit Plane 3
8	4	Graphics Bit Plane 4
F	5 (highest)	Cursor

* OL bus values other than those listed are not allowed.

Table 8. RAMDAC Data Priority

Interlace FIFO

The Interlace FIFO is located at U4 and shown on sheet 2 of the schematics. This 512-word by 8-bit FIFO converts the progressive scan video to interlaced video.

Data reading and writing are asynchronous. During the visible pixel portion of every other visible progressive scan line, data is written to the FIFO at the pixel clock rate (24.5454514 MHz). NFL (NTSC FIFO Load) strobes the V0 - V8 data into the FIFO. Data is read out of the FIFO at one-half the pixel clock rate. Thus, upon completion of the storage of one progressive scan of video (640 pixels), 320 of these pixels will have been read out to form the first half of an interlaced scan line. During the next progressive scan, no pixels are stored, but the remaining 320 pixels from the previous progressive scan are read out. Thus, at the completion of an interlaced scan line, the FIFO is empty. This process repeats 480 times during the writing of two progressive scan frames, resulting in one interlaced frame.

The V0 - V8 input data is multiplexed data. The lower four bits are either the four LSBs of the 8-bit video byte or the four graphics/cursor bits. V8 is called the Graphics Flag and is used by U10 to demultiplex the graphics/cursor bits. IV0 - IV8 return to U10 in addition to driving the P0 - P7 port of U6. U10 ANDs the Graphics Flag bit (V8) with each of the four LSB outputs of the FIFO. If the Graphics Flag is high, V0 - V3 drive IG0 - IG3 (OL inputs to U6).

Output Drivers

Output drivers located at U7A, U7B and U8A drive the 75 ohm progressive scan composite sync (PSYNC), vertical drive (VDRIVE) and interlaced scan composite sync (ISYNC) outputs, respectively.

Master Controller

This description consists of additional functional detail for U9 and U10. It also includes timing diagrams of externally available signals.

U9 Detailed Circuit Description

Many of the internal 64-bit buses in U9 are labeled as four 16-bit buses. This is a labeling choice and does not imply that different segments of a particular 64-bit bus are processed differently or that they have different functions than other segments of the same bus.

The U9 Detailed Circuit Description is based on Figure 7 on page 47. This figure consists of the following functional blocks:

- Memory Write Latch Strobe Generator
- Data Bus Interface
- Graphics Mask and High/Low Word Select
- Graphics Prioritizer and Processor
- One-of-Eight Selector
- Resynchronizing Latch
- Graphics Demultiplexer
- Command Latch
- RAM Timing Generator

Memory Write Latch Strobe Generator

The RAM Data Bus Interface block assembles 64-bit words from four 16-bit PS/2 data bus words by latching four consecutive words into four separate 16-bit input latches. The Memory Write Latch Strobe Generator produces the latch strobes for these latches.

The PS/2 transfers data to the RAM Data Bus Interface in the 16-bit mode. The PS/2's LSB address bit (A0) functions as the LSB of a byte address and is undefined when the PS/2 is transferring 16-bit words. Address bits A1 and A2, the PS/2's second and third address bus LSBs, respectively, function as the LSB and second LSB, respectively, of the word address during 16-bit transfers.

The Memory Write Latch Strobe Generator uses MWS/ (Memory Write Strobe - memory mapped to addresses XXC00000H - XXDFFFFFH), and LA1 and LA2 (Latched A1 and Latched A2, respectively) to generate the latch strobes. MWS/ serves as an enable for the Memory Write Latch Strobe Generator. Table 9 below shows the strobe generation.

MWS/	LA2	LA1	LA0	Strobe
0	0	0	0	MWLA
0	0	1	0	MWLB
0	1	0	0	MWLC
0	1	1	0	MWLD
1	X	X	X	none

X = Don't Care

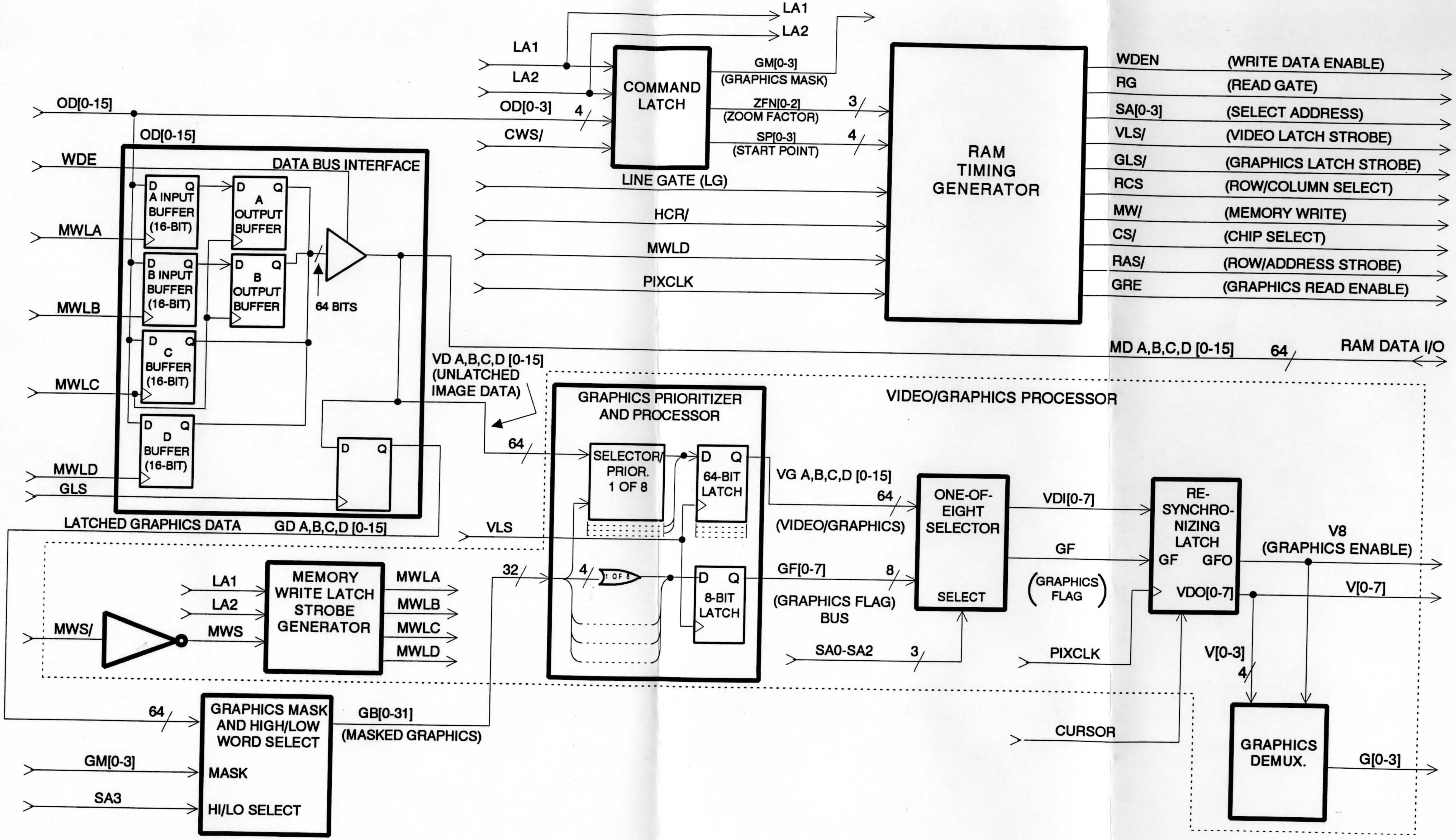
Table 9. Memory Write Latch Strobe Generation

Data Bus Interface

The Data Bus Interface functions as an interface between the PS/2's data bus and the Dynamic RAM, and between the Dynamic RAM and the video and graphics processing logic.

The PS/2's data bus (OD0 - OD15) is applied to the inputs of four 16-bit latches. Each latch is strobed as a function of LA1 and LA2 (see Table 9 above). The latches strobed by MWLA and MWLB are double buffered. MWDC not only latches the third least significant word (LSW), it also transfers the first and second LSWs from their respective input latch to their respective output latch. Once all four latches are loaded, the 64 output bits are transferred to the Dynamic RAM when WDEN (Write Data Enable) enables the 64 tri-state output buffers (one clock pulse after MWLD is asserted). WDEN is an output of the RAM Timing Generator.

Double buffering of the two LSW inputs reduces the possibility of the PS/2 overwriting the LSW and second LSW latches before U9 has had an opportunity to transfer its 64-bit word to the RAM. Sometimes, this is still not enough spare time to prevent an overwrite. Therefore, to guarantee that no data is overwritten before it is transferred, U9 asserts CHRDY (Channel Ready) when the PS/2 attempts to overwrite the second LSW. This forces the PS/2 into wait states until the SDA can accept more data.



FILENAME SDA5.DRW

Figure 7. U9 Detailed Functional Block Diagram

Dynamic RAM output data can be either video data or graphics data. If the data is graphics, the RAM Timing Generator asserts GLS (Graphics Latch Strobe), thereby latching the data into the 64 D-type latches shown in the lower-right corner of the Data Bus Interface block in Figure 7.

If the Dynamic RAM output data is video data, it is passed to the Graphics Prioritizer and Processor block via a 64-bit buffer in the Data Bus Interface. That is, video data is not latched by the Data Bus Interface block.

Graphics Mask and High/Low Word Select

The Graphics Mask and High/Low Word Select has two functions. First, it selects the upper or lower half of the 64-bit graphics word as the currently active half-word. Second, it provides a masking capability which allows all graphics bits within one or more bit planes to be masked off.

Since each graphics pixel requires only four bits, there are eight pixels in each half-word. The Graphics Mask and High/Low Word Select block uses SA3 to select the graphics half-word that corresponds to the current 8-byte video word. SA3 is low while processing video pixels from even RAM addresses; SA3 is high while processing video pixels from odd RAM addresses. SA0 - SA2 function as video and graphics pixel selectors.

SA3, the high/low graphics word selector, is the MSB of a 4-bit binary counter in the RAM Timing Generator that generates SA0 - SA3. This counter is preset to the Start Point at the beginning of each horizontal line by SP0 - SP3. It increments at the pixel clock rate divided by the Zoom Factor. Its SA2 output increments the Video Address Generator (see the U10 description) each time it goes from high to low (count 7 to 8 or count F_H to 0). SA3 is low during counts 0 - 7 and high during counts 8 - F_H. When SA3 is low, GB0 - GB31 are driven by GDA and GDB 0 - 15 (lower half-word); when SA3 is high, GB0 - GB31 are driven by GDC and GDD 0 - 15 (upper half-word). The Graphics Address Generator increments each time SA3 goes low.

The four graphics mask bits are labeled GM0 - GM3. Each mask bit is compared to its corresponding bit within each nibble (graphics pixel) of the current half-word. If the mask bit is on (high), the corresponding bit in all nibbles is blocked, turning off the entire bit plane.

The Graphics Mask and High/Low Word Select block consists of 32 three-to-one multiplexers. Each multiplexer has a corresponding bit from each half-word, a mask bit and a half-word selector input applied to it. Table 10 on the next page describes the multiplexer's output.

Graphics Prioritizer and Processor

The Graphics Prioritizer processes eight image and eight graphics pixels at a time. This block first prioritizes the graphics bits, limiting the number of graphics bits within each graphics nibble to a maximum of one. When more than one graphics bit in a graphics nibble input is high, all but the highest order bit is turned off.

Next, the graphics nibble conditionally replaces the lower four bits of its corresponding video byte. If any graphics bit in a graphics nibble is high at the output of the prioritizing logic, the replacement is made. If the graphics nibbles contain no high bits, their corresponding video bytes are unaffected. The bits in the Graphics Flag Bus (described below) identify those video bytes that contain graphics in their lower nibbles.

GM(a)	SA3	G(Y)	G(Y+32)	Output	
0	0	0	0	0	
0	0	0	1	0	
0	0	1	0	1	= G(Y)
0	0	1	1	1	
0	1	0	0	0	
0	1	0	1	1	= G(Y+32)
0	1	1	0	0	
0	1	1	1	1	
1	X	X	X	0	= 0 (Off)

X = Don't Care.
GM(a) is the graphics mask bit where (a) is 0 - 3.
G(Y) is a lower half-word graphics bit (Y = 0 - 31).
G(Y+32) is the corresponding upper half-word graphics bit and equals the lower half-word bit plus 32.

Table 10. Graphics Mask Truth Table

Each of the eight Mux/Prioritizer blocks in the Graphics Prioritizer and Processor section shown in Figure 7 simultaneously prioritizes the graphics and overwrites the prioritized graphics onto the video bus. Each Selector/Prioritizer block processes one pixel. The output of each block drives a byte-wide D-Type latch that is strobed by VLS/ (Video Latch Strobe). VLS/ is an output of the RAM Timing Generator.

Finally, the Graphics Prioritizer and Multiplexer forms an 8-bit Graphics Flag Bus by ORing the four bits in each of the eight graphics nibbles. That is, the ORed result of the first nibble forms GF0, the ORed result of the second nibble forms GF1 ... and the ORed result of the eighth nibble forms GF7. GF0 - GF7 are the outputs of a byte-wide D-Type latch that is strobed by VLS/. GF0 - GF7 are used as demultiplexing controls.

One-of-Eight Selector

The One-of-Eight Selector sequentially addresses video bytes and their associated graphics flag at the pixel clock rate. The 64-bit input consists of four 16-bit buses labeled VGA (bit 0 - 15), VGB (bits 16 - 31), VGC (bits 32 - 47) and VGD (bits 48 - 63). SA0 - SA2 (Select Address bits 0 - 2) address the bytes on these buses. Table 11 below shows which bytes and word bits are addressed by SA0 - SA2.

SA2	SA1	SA0	Video Output	Graphics Flag Output
0	0	0	VGA[0 - 7]	GF0
0	0	1	VGA[8 - 15]	GF1
0	1	0	VGB[0 - 7]	GF2
0	1	1	VGB[8 - 15]	GF3
1	0	0	VGC[0 - 7]	GF4
1	0	1	VGC[8 - 15]	GF5
1	1	0	VGD[0 - 7]	GF6
1	1	1	VGD[8 - 15]	GF7

Table 11. One-of-Eight Selector Inputs and Outputs

Resynchronizing Latch

The Resynchronizing Latch block consists of nine D-Type latches and five 2-input OR gates. The latches are clocked by pixel clock (PIXCLK) and driven by the eight video bits (VDI0 - 7) and associated graphics flag (GF). The latches for VDI0 - VDI3 and GF are driven by OR gates that OR the respective input with CURSOR, the output of the Cursor Generator. Thus, if CURSOR is high, V0 - V3 and V8 are high. This block merges the output of the Cursor generator into the multiplexed video and graphics bus and ensures that these bits and graphics flag are in synchronism with each other and the pixel clock when they exit the chip.

Graphics Demultiplexer

The Graphics Demultiplexer recovers the four graphics bits from the lower four bits of the output multiplexed video bus (V0 - V3). These graphics bits drive the Progressive Scan RAMDAC's Overlay inputs (OL0 - OL3).

If V8, the Graphics Flag, is high, V0 - V3 are carrying graphics or cursor data. To recover this data, each of the lower four bits is applied to a two-input AND gate along with V8. When V8 is high, the data on V0 - V3 passes to G0 - G3 (Graphics bits 0 - 3), respectively.

Command Latch

The Command Latch generates these three memory mapped outputs used internally by U9:

- SP0 - SP3 (Start Point)
- ZFN0 - ZFN2 (Zoom Factor Inverted)
- GM0 - GM3 (Graphics Mask)

The Command Latch block consists of three sets of D-Type latches. These latches are the SP, ZFN and GM latches.

As stated in the Functional Description, the SDA is controlled by four 16-bit memory mapped control words. These control words are memory mapped to addresses E00008$_H$ - E0000F$_H$. The Start Point, Zoom Factor and Graphics Mask are segments of three of these Command Words. Bits 0-3 of E00008$_H$ are SP0 - SP3, respectively; bits 8-A of E0000A$_H$ are ZF0 - ZF2, respectively; and bits C-F of E0000E$_H$ are GM0 - GM3, respectively. Within the Command Latch block, the appropriate OD bus lines drive the D-inputs of the latches.

CWS/, LA1 and LA2 enable one of three Command Latch strobes. CWS/ is asserted during memory write cycles to addresses E00008$_H$ - E0000F$_H$. LA1 and LA2 identify the 16-bit word. Table 12 below describes the Command Latch memory mapping.

Address (hex)*	CWS/	LA2	LA1	Latch Strobe
XXE00008	0	0	0	Start Point
XXE0000A	0	0	1	Zoom Factor
XXE0000C	0	1	0	**
XXE0000E	0	1	1	Graphics Mask
0 - XXE00007	1	X	X	None

* XX is determined when running the system configuration diskette.

** Video Frame Start Address. This word is not used by the Command Latch. Refer to the RAM Read/Write Addressing section for information on the use of this word.

Table 12. Command Latch Mapping

The Start Point is part of the Roam feature. Its function is described below.

Figure 8 on the next page shows how image and graphics frames are stored in RAM. It shows that an image frame consists of 480 640-pixels (80 words) segments. When zoomed in, the number of bytes required is 640 divided by the Zoom Factor, i.e., a Zoom Factor of 5 requires 128 bytes. At the maximum Zoom Factor of 8, only 80 bytes are used in each frame line. Thus, for a particular Zoom Factor, the number of bytes required for displaying a horizontal line is fixed. There are 8 bytes per video word. Thus, in this example, 16 words are required for the 128 bytes if all bytes are used.

Horizontal roaming can be thought of as a window that can be moved left or right within the assigned video frame. For the Zoom Factor of 5 in the example above, this window is 128 bytes long. Note that if the window is moved slightly to the right (less than eight bytes), part of the first word must be discarded. When this happens, the right end of the window extends into the next word to retrieve the bytes required to complete the line. This window is not allowed to extend past the frame's line end. To roam more than eight bytes, the start word address is incremented or decremented as required. Thus, roaming consists of adjusting the start point (start word address and the byte within that word). Once the start point is selected, the required number of bytes is read to paint the line.

The Start Point parameter points to the start byte within the first word used in the painting of the horizontal line. SP0 - SP3 are used by the RAM Timing Generator to preset the SA0 - SA3 bus to point at the desired byte (and graphics nibble) of the first word in each horizontal line.

The Zoom Factor (ZFN0 - ZFN2) is required by the RAM Timing Generator. Pixel replication is required for zoom factors greater than one. This affects the Select Address (SA0 - SA3) that addresses the bytes within the current video and graphics words. As the Zoom Factor goes up, the Select Address increments at a lower rate (equals pixel clock divided by the Zoom Factor).

The Graphics Mask bus (GM0 - GM3) turns off graphics bit planes. It is used by the Graphics Mask and High/Low Select block (described above).

RAM Timing Generator

The RAM Timing Generator is described by Timing Diagrams 1 and 2 located on pages 55 and 56, respectively. Timing Diagram 1 shows RAM image and graphics display timing while no new frame is being loaded into RAM. Timing Diagram 2 shows RAM image and graphics display timing while a new frame is being loaded into RAM.

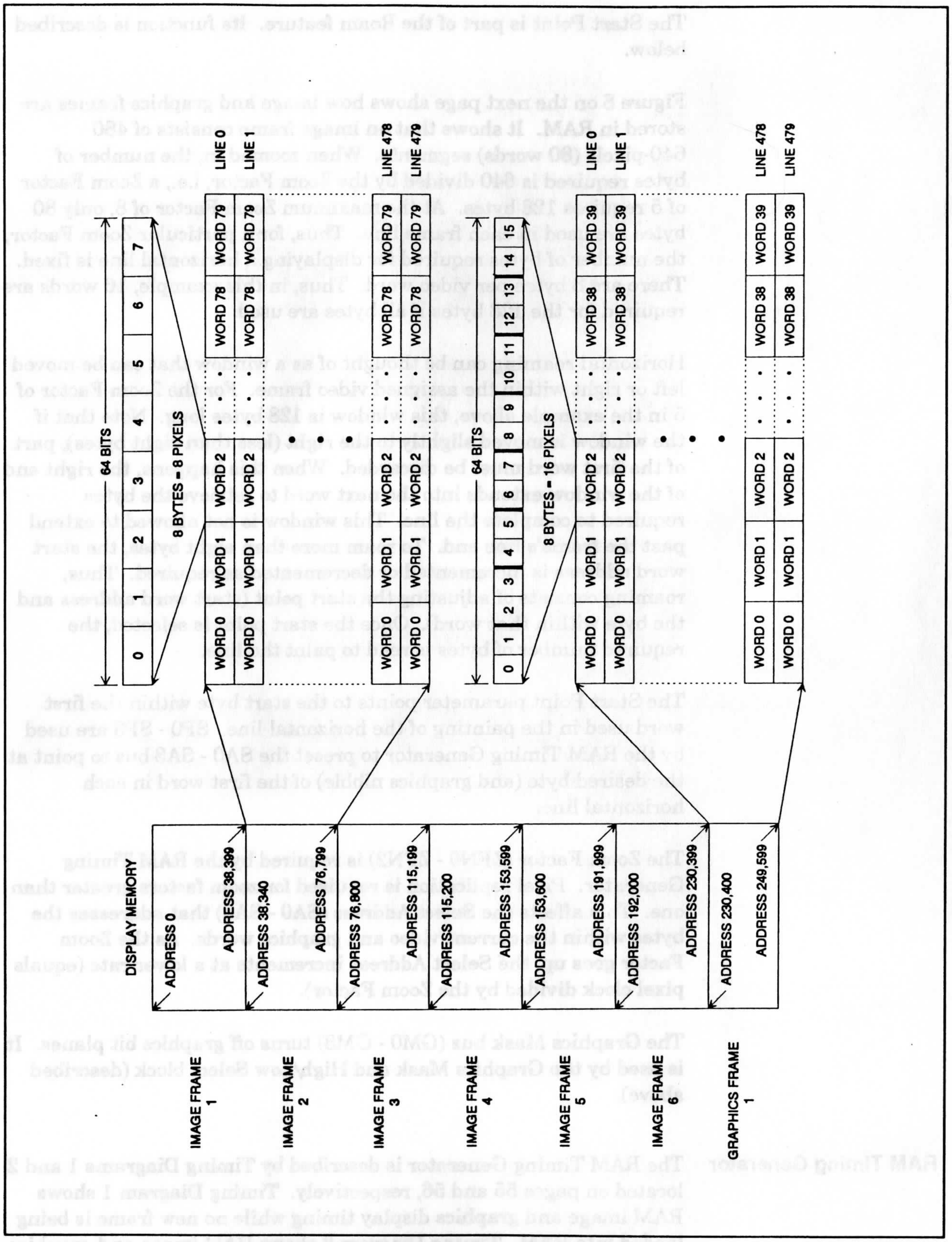
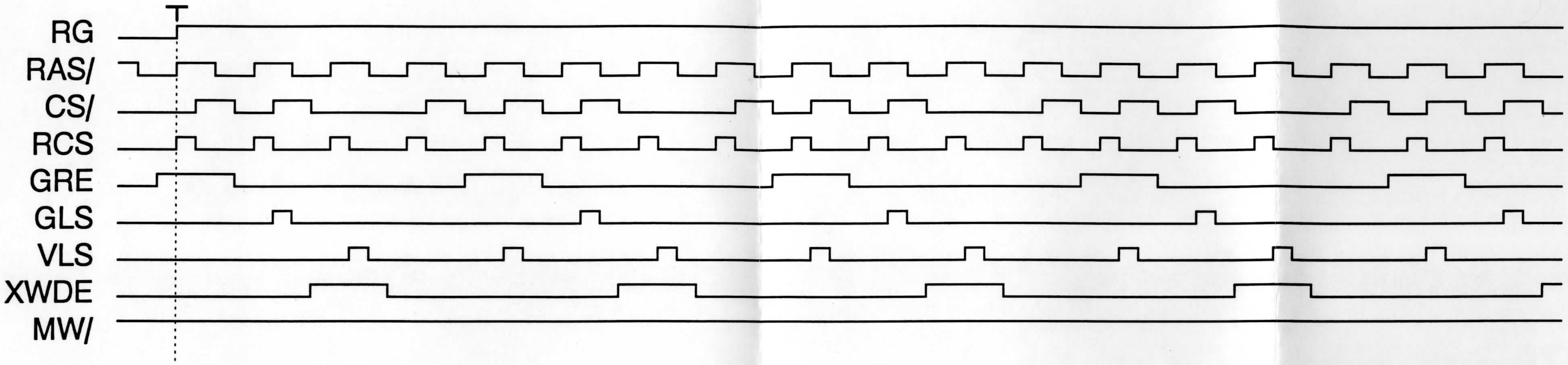
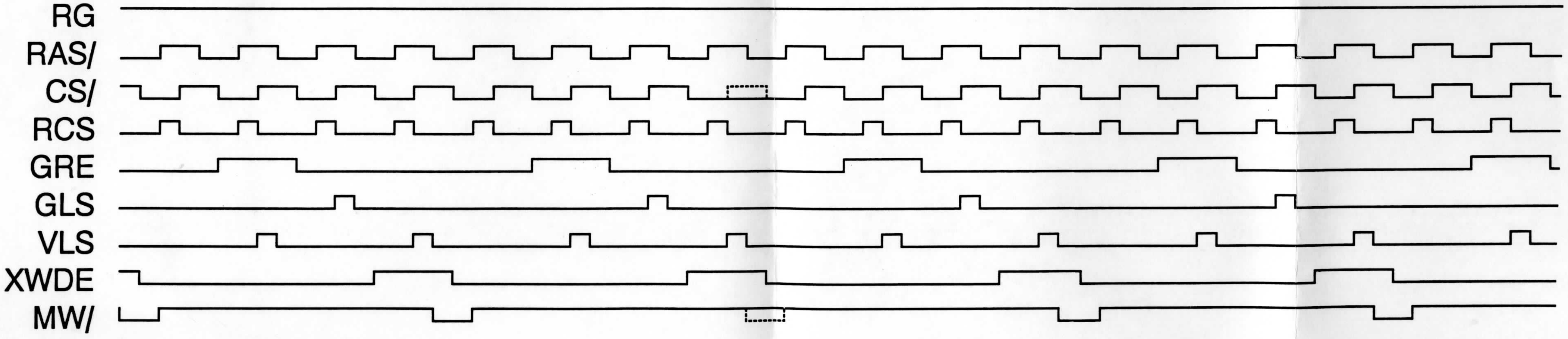


Figure 8. RAM Frame Organization Diagram



Timing Diagram 1. RAM Timing (No Frame Load in Progress)



Timing Diagram 2. RAM Timing (Frame Load in Progress)

U10 Detailed Circuit Description

Refer to Figure 5 on page 26. U10 consists of the following blocks:

- Timing Generator
- Address Strobe Generator
- RAM Read/Write Addressing Section
- Address Input Latch
- RAMDAC Byte Selector
- Interlace Generator
- Interlaced Graphics Bus Demultiplexer
- Interrupt Logic Block

Timing Generator

The RAMDACs require several TV timing signal inputs to produce the progressive and interlaced scans. These signals are:

- BLANK (progressive composite blanking)
- NCS (NTSC interlaced Composite Sync)
- NCB (NTSC interlaced Composite Blanking)
- PIXCLK (Pixel Clock - progressive scan)
- NPC (NTSC Pixel Clock - interlaced scan)
- VD (Vertical Drive - progressive scan)

The Timing Generator generates these signals from the 24.5454514 MHz Pixel Clock (PIXCLK). It consists of two intermediate generators, the Horizontal Timing Generator and the Vertical Timing Generator, which drive the Progressive and Interlace Synthesizers. The Horizontal Generator is driven by PIXCLK; the Vertical Timing Generator is driven by the LG (Line Gate) output of the Horizontal Timing Generator. Intermediate generator signals are combined to produce the complex output signals.

All Timing Generator outputs are used externally. Thus, they can be used for diagnostic troubleshooting. The documentation for this section consists of Timing Diagrams 3 and 4 shown on page 57.

Address Strobe Generator

Refer to the Address Strobe Generator description in the U10 Functional Description. See page 28.

RAM Read/Write Addressing Section

The Dynamic RAM requires an 18-bit address for reading and writing. Because this RAM has a multiplexed address input, these addresses are sent to the RAM as a 9-bit row address (nine LSBs) and a 9-bit column address (nine MSBs). The RAM Read/Write Addressing Section generates 18-bit read and write addresses and transmits them to the RAM as two 9-bit words.

The process of displaying an image consists of retrieving (reading) data from an image frame and a graphics frame simultaneously. The image data read, graphics data read, and write cycles are interleaved. The RAM Read/Write Addressing Section consists of an Image Address Generator, Graphics Address Generator and Write Address Latch. These address sources drive separate 18-bit ports of an output selector/multiplexer. The output selector/multiplexer selects the address source and then outputs the address as two 9-bit components.

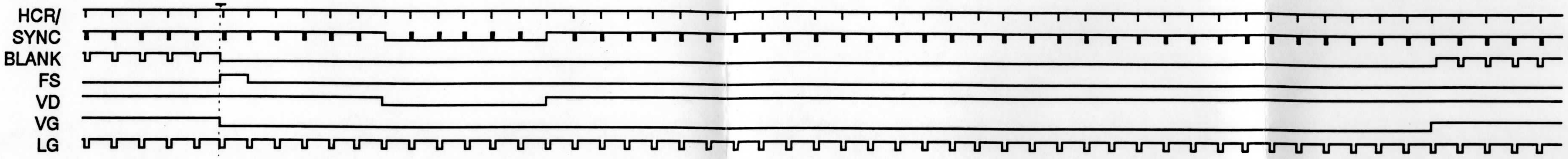
To minimize PS/2 overhead, SDA image and graphics read addressing is automated so that only a frame start address, a zoom factor and a roam offset are required from the PS/2 to display an image and/or graphics frame. Each of these inputs is described before discussing the RAM Read/Write Addressing Section.

Frame Start Address

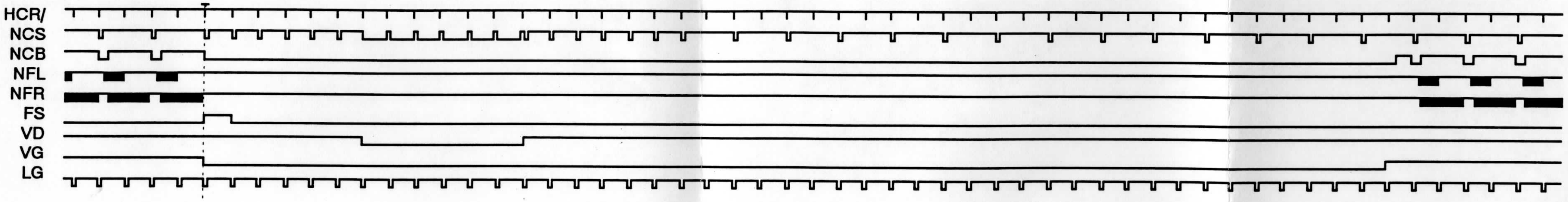
The Frame Start Address is the address of the 64-bit word containing the first pixel in a frame. In the process of displaying an image, the RAM Read/Write Addressing Section must provide two addresses at all times, an image read address and a graphics read address. Read addresses are computed by adding an offset to the Frame Start Address. Thus, two Frame Start Addresses are required, the Image Frame Start Address and the Graphics Frame Start Address.

Each pixel of each frame is part of a 64-bit word having an absolute word address. These addresses are computed by adding the pixel's relative word address within the frame (000 - 38,399 for image frames, or 000 - 19,199 for graphics frames) to the Frame Start Address. For example, assume image frame two begins at word address 38,400 as shown in Table 1 on page 16. The relative word address containing the first pixel (upper-left pixel) in the frame is 0000. Adding these two numbers together yields a word address of 38,400. Similarly, the last pixel in a frame is the eighth byte of the word located at relative address 38,399. Adding the Frame Start Address to this relative word address yields an address of 76,799.

By generating read addresses as a sum of the Frame Start Address and the relative address, the Image and Graphics Address Generators can share the Relative Address Generator logic, thereby reducing the circuitry needed to implement the RAM Read/Write Addressing Section. It also simplifies roam and zoom implementation.



Timing Diagram 3. Progressive Scan Timing



Timing Diagram 4. Interlaced Scan Timing

Zoom Factor

The SDA has eight user selectable zoom factors ranging from 1 to 8. The zoom feature allows the user to magnify a selected area of a frame by the square of the zoom factor. Table 13 below shows the zoom factors and resulting area magnifications.

Zoom Factor	Area Magnification
1	1 (no magnification)
2	4
3	9
4	16
5	25
6	36
7	49
8	64

Table 13. Zoom Factor Versus Area Magnification

Zooming is accomplished by replicating (repeating) pixels and horizontal lines by a factor equal to the zoom factor integer. For example, if the zoom factor is 5, each byte in the area to be zoomed is output to the RAMDACs five times. In addition, the data for each horizontal line is repeated five times. Thus, each pixel in the zoom area is expanded to a 5-pixel (horizontal magnification) by 5-line (vertical magnification) area, a magnification of 25.

Roam Offset

The zoom factor decreases the displayed portion of a frame by the square of the zoom factor. In the example above, only 128 bytes (16 words) are required to paint a 640-pixel horizontal line; only 96 lines are required to paint 480 lines on the display. Via the roam feature, this 128-byte by 96-line "zoom window" can be moved anywhere in the frame via the PS/2's mouse. No part of the zoom window can move outside the current frame.

Zoom Factors greater than 1 result in a zoom window that is smaller than the frame. Moving this window to a point of interest requires a Roam Offset. The zoom window's reference point is its upper-left corner pixel. The zoom window is positioned by assigning this pixel to a relative address within the frame called the Roam Offset address. In the example described in the Zoom Factor description, the zoom window is 128 bytes by 96 lines. If this window's upper-left pixel is to be located at pixel 320 on the eleventh horizontal line in a 640-pixel by 480-line video frame, a Roam Offset of 6720 is required (640 pixels/horizontal line times 10 lines plus the 320 pixels on the eleventh line = 6720).

RAM Read/Write Address Generators

Figure 8 (page 54) shows a typical RAM frame partitioning. It also shows that each image frame is treated as 480 groups of 80 words each, and each graphics frame is treated as 480 groups of 40 words each. These groups correlate directly to horizontal lines (640 pixels).

Figure 9 on the adjacent page is a functional block diagram of the RAM Read/Write Addressing section. The RAM Read Addressing portion of Figure 9 contains these three generators:

- Relative Address Generator
- Image Address Generator
- Graphics Address Generator

The Relative Address Generator requires the Roam Offset and Zoom Factor inputs; the Image and Graphics Address Generators require the Image and Graphics Frame Start input, respectively. The OD bus drives four sets of input latches that allow the PS/2 to load a Roam Offset, an Image Frame Start Address, a Graphics Frame Start Address and a Zoom Factor into the RAM Read Addressing section.

Relative Address Generator

Refer to Figures 8 and 9. Each image frame consists of 38,400 64-bit words; each graphics frame consists of 19,200 64-bit words. The Relative Address Generator treats all image frames as a 38,400-word block of memory starting at address 000 and organized as 480 concatenated 80-word blocks. The 19,200-word graphics frames are organized as 480 concatenated 40-word blocks.

The inputs to the Relative Address Generator are the Roam Offset and Zoom Factor. If the Zoom Factor is 1, the Roam Offset must be zero, as any attempt to roam under this condition would move part of the zoom window outside the frame area, which is not allowed.

When the Zoom Factor is greater than 1, roaming is allowed, and pixel and line replication occur. Line replication is performed by the Relative Address Generator; pixel replication occurs in the One-of-Eight Selector block in U9 which is controlled by the RAM Timing Generator section in that chip. When the Zoom Factor is greater than 1 (zooming), the PS/2, in response to user inputs, repositions the zoom window by sending the relative pixel address (Roam Offset) that corresponds to the desired location of the upper-left corner of the zoom window. This address is latched into the Roam Offset Latch.

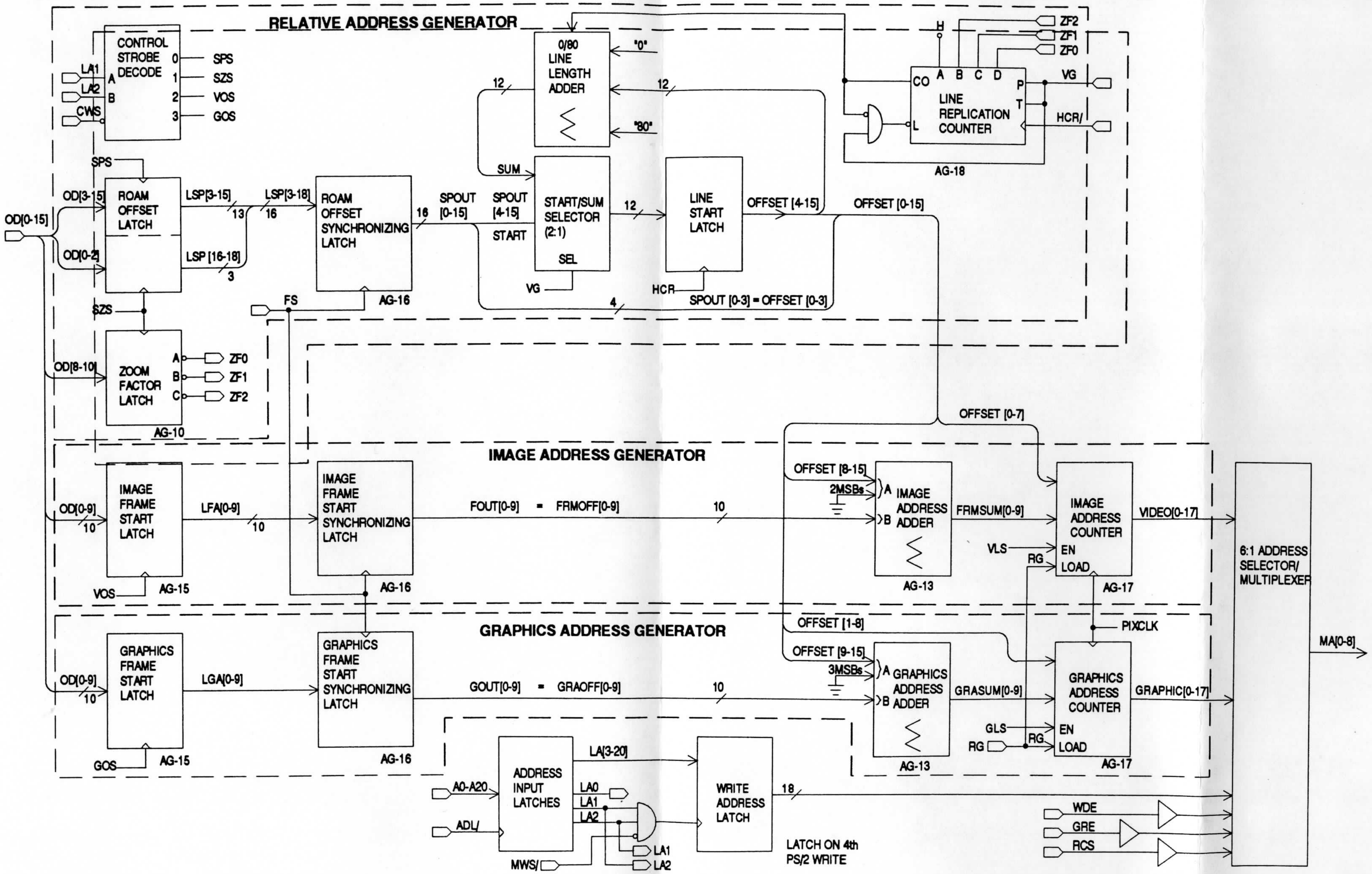


Figure 9. RAM Read/Write Addressing Section

The Relative Address Generator provides the relative line start address to the Image and Graphics Address Generators at the beginning of every scan line. After painting the last pixel in the current scan line, a new line start address is computed and loaded into the Image and Graphics Address Counters. This process repeats until all 480 lines are painted.

Note that only a segment of the frame line is used when zooming. Regardless of the Zoom Factor, the start address of the next line is always 640 pixels (80 words or 40 words for image or graphics frames, respectively) greater than the previous line start address. Thus, all that's required to generate the next line's relative start address is to add 80 (or 40 for graphics) to the previous line start address. This number is then saved in a latch for use in computing the next line start address. If the current line is to be repeated, the adder adds zero instead of adding 80 to the previous line start address. Thus, the Zoom Factor controls the adder.

Refer to Figure 9. The first horizontal line begins with the Roam Offset address. Nineteen binary bits are required to represent the full Roam Offset range in pixels (0 - 307,199$_D$ = 0 - 4AFFF$_H$). Since the 3 LSBs represent the pixel within the addressed word, they are not needed by the RAM Read/Write Addressing Section. However, these 3 LSBs are required by the RAM Timing Generator in U9 to determine the start pixel in the first word of each line (see Figure 7). The remaining 16 bits represent the Roam Offset word address. The PS/2 writes the 19-bit Roam Offset to the SDA card as two words. The first word is a 16-bit word written to address E00008$_H$. This address causes SPS in U10 to be asserted, latching OD3 - OD15 in the Roam Offset Latch (13 LSBs of the word address). At the same time, OD0 - OD3 are latched (pixel address) into U9's Command Latch (see Figure 7). Next, the PS/2 writes the upper three bits of the Roam Offset word address (OD0 - OD2) and the zoom factor (OD8 - OD10) by writing to address E0000A$_H$. To summarize, the output of the Roam Offset Latch is a 16-bit word offset address and also the 16 MSBs of a 19-bit pixel offset address.

The output of the Roam Offset Latch is latched into the Roam Offset Synchronizing Latch by FS (Frame Start). This latch and the Image Frame Start and Graphics Frame Start Synchronizing Latches simultaneously present the Roam Offset and Image and Graphics Frame Start addresses to the RAM Read Addressing Section logic. The output of the Roam Offset Synchronizing Latch drives the Start/Sum Selector. When VG is low (during vertical blanking), the latched Roam Offset passes through the selector to the Line Start Latch which is strobed by HCR (Horizontal Count Reset). The output of this latch drives the Image and Graphics Address Adders as well as the Line Length Adder logic.

The Line Length Adder logic adds either 0 or 80_D (0 or 1010000_B, respectively) to the output of the Line Start Latch. Since the adder never modifies the 4 LSBs of the Roam Offset Address, these bits (SPOUT0 - 3) bypass the adder. Thus, the adder either adds 000 or 101_B to OFFSET4 - 15. This saves four binary adder stages in U10. The selection between 0 and 80 is controlled by the Line Replication Counter. This counter is preset to the ones complement of the Zoom Factor and is incremented by HCR/ (Horizontal Count Reset). Upon reaching 1111_B, CO (Carry Out) goes high causing the Line Adder to add 101_B to the output of the Line Start Latch. The Line Length Adder's output drives an input port of the Start/Sum Selector. During the visible portion of the scan (after vertical blanking), VG selects the Line Length Adder's sum output as the input to the Line Start Latch. Thus, the address output of this latch increases by 101_B at the end of each scan line if the CO output of the Line Replication Counter is high.

Image and Graphics Address Generators

The Relative Address Generator is common to the Image and Graphics Address Generators. Each of these generators contains a 10-bit adder that adds the output of the Relative Address Generator to its respective frame start address. The Image and Graphics Address Adders compute the absolute line start addresses by adding their respective frame start address to the relative start address. The adder outputs are used to preset address counters that drive the 6:1 Address Selector/Multiplexer.

The Roam Offset required by the Graphics Address Generator is exactly half that of the Image Address Generator. This offset is divided by two by shifting it down one bit position in its connection to the Graphics Address Adder and grounding the MSB of the Graphics Address Adder's A-input.

RAM Write Address Generation

Refer to Figure 9. The PS/2 provides a 24-bit address input to the SDA card. The 21 LSBs of the address bus (A0 - A20) are latched into the Address Input Latches on the rising trailing edge of ADL/. LA3 - LA20, the latched 18 MSB outputs of the Address Input Latches block, drive the inputs of the Write Address Latch. This latch is strobed by an AND gate driven by LA1, LA2 and MWS/.

When the PS/2 writes to the SDA's Dynamic RAM, it writes 16-bit data words. The SDA assembles these data words into 64-bit words and stores them in the Dynamic RAM.

While the PS/2 is transferring 16-bit words, the LSB address bit (LA0) is undefined. During this time, LA1 functions as the LSB of a 16-bit word address and increments after each transfer. Therefore, LA1 and LA2 are both high after the fourth 16-bit transfer (8 bytes or 64 bits). If MWS/ is low, LA1 and LA2 cause the Write Address Latch to be strobed after each 8-byte transfer. Since the Write Address Latch only latches the portion of the address that represents the word address (LA3 - LA20), its 18-bit output increments after each 64-bit data transfer. During the next *write opportunity* cycle, the 64 data bits are stored at the RAM location specified by the output of the Write Address Latch.

MWS/ (Memory Write Strobe) is the output of combinatorial logic that goes low if:

- MEMW/ is low (PS/2 Memory Write Cycle), and
- SBHE/ is low (PS/2 is transferring 16-bit data), and
- MWD/ is low (PS/2 is addressing RAM address space)

Address Input Latch

Refer to the Address Input Latch description in the U10 Functional Description section (page 29).

RAMDAC Byte Selector

Refer to the RAMDAC Byte Selector description in the U10 Functional Description section (page 29).

Interlace Generator

Refer to the Interlace Generator description in the U10 Functional Description section (page 30).

Interlaced Graphics Bus Demultiplexer

Refer to the Interlaced Graphics Bus Demultiplexer description in the U10 Functional Description section (page 30).

Interrupt Logic Block

Either the IRQ15/ or the IRQ11/ interrupt input to the PS/2 is asserted if FI, INTEBL/ and BDENBL_ are asserted. FI is the latched Frame Start (FS) signal (latched in U10). INTEBL/ and BDENBL_ are asserted by U3 upon completion of the setup sequence performed by the PS/2 following power-up.

Cursor Generator

The Cursor Generator consists of U32. For information on this section, refer to the Cursor Generator functional description on page 30.

SDA Diagnostics

The SDA Diagnostics diskette consists of High Level and Low Level Diagnostics. The High Level Diagnostics run inside the McIDAS-OS2 environment and allow you to manipulate the SDA with a few simple keystrokes. The Low Level Diagnostics run outside the McIDAS-OS2 environment and allow you to manipulate the SDA at the register, word and bit level of its domain.

High Level Diagnostics

The High Level Diagnostics consist of the files listed in Table 14.

Files	Contents
INSTALL.CMD	installation script file
SDADIAG.ZIP	compressed diagnostic menu and supporting files
PKUNZIPF.EXE	decompression routine
README.TXT	text file containing a description of the diagnostics

Table 14. High Level Diagnostics Files

The SDADIAG.ZIP file contains 17 modules that make up the SDA High Level Diagnostics. The INSTALL.CMD places these files into two directories. All files with a ".EXE" extension are placed in the \MCIDAS\CODE directory; all other files are placed in the \MCIDAS\DATA directory. Table 15 on page 70 is a list of the modules that make up the SDADIAG.ZIP file.

Module	Contents
AREA9997	McIDAS Image file (Test pattern 1)
AREA9998	McIDAS Image file (Test pattern 2)
AREA9999	McIDAS Image file (Test pattern 3)
BEEP.EXE	McIDAS application - System Beep
IDLE.EXE	McIDAS application - System Timeout
PATTERN.EXE	McIDAS application - Draw graphics test pattern
SDADIAG.MNU	McIDAS F Key Menu used for the diagnostic interface
SDADIAG1.ET	Image enhancement file 1
SDADIAG2.ET	Image enhancement file 2
SDAGRAPH	McIDAS BATCH file for graphic test initialization
SDAINIT	McIDAS BATCH file for menu initialization
SDALOOP	McIDAS BATCH file for loop test initialization
VIRT9991	Virtual graphics file 9991 (Graphics Frame 1 text)
VIRT9992	Virtual graphics file 9992 (Graphics Frame 2 text)
VIRT9993	Virtual graphics file 9993 (Graphics Frame 3 text)
VIRT9999	Virtual graphics file 9999 (graphics color bar)
WIPUT.EXE	McIDAS application - Write message to text screen

Table 15. SDADIAG.ZIP Modules

Installing the Diagnostics

Use the steps below to install the High Level Diagnostics package.

1. Insert the diskette containing the Diagnostics into the floppy disk drive of the McIDAS SDA workstation.

2. From an OS/2 window session,

   Type: **A:\INSTALL**
   Press: **Enter**

   This copies the files from the diskette to the hard drive. In some cases, the files may already exist and you will be asked if you want to write over the existing files.

3. Switch to the McIDAS session. Then,

   Type: **MAKMNU SDADIAG**
   Press: **Enter**

   This is a McIDAS application which constructs the SDA diagnostic menu on your workstation. When the "done" message appears, continue with step 4.

4. To view the newly created SDA diagnostic menu,

   Press: **Esc**

Running the Diagnostics

The SDA diagnostic menu is based on the McIDAS F Key Menu interface. This interface is designed to work with the F keys located at the top of the workstation's keyboard (usually F1 through F12). The text on the workstation's text display indicates the function of selected F keys for each screen. Simply read the text and press the appropriate F key to initiate the associated function.

Please make sure you read the text before you press an F key!

The diagnostic package makes the following assumptions about the SETUP of your SDA workstation.

- The workstation is configured as an SDA.
- There are at least 3 image and 3 graphics frames allocated.

If the workstation does not meet these requirements, or you are unsure of the configuration, consult the SETUP program information in your *McIDAS-OS2 Users Guide* for more details on workstation configuration.

From the main menu, you can select from 7 different diagnostic tests. These tests are designed to test specific attributes of the SDA hardware/firmware and software. Table 16 below describes these tests.

F Key	Description
F1	Image Enhancements; creates and restores image enhancements to image plane 1
F2	Graphic Enhancements; tests the 4 graphic planes color and on/off toggles
F3	Cursor Functions; tests capabilities of the hardware cursor
F4	Zoom & Roam; exercises the full range of zoom factors and roaming modes
F5	Interlaced Display Test; stresses I/O buffering and checks for timing problems
F6	Loop Test; stresses the image and graphics looping functions at fastest loop speed
F7	Image/Graphic interaction; stresses looping, enhancement restore and graphic buffering

Table 16. SDA High Level Diagnostic Tests

When running the diagnostic tests, a BEEP indicates the END of the test. Once you hear the beep you can make another selection. Pressing an F Key prior to the beep will not invoke the function assigned to the selected F key.

Deleting the Diagnostic Menus

When you're satisfied the SDA is working properly, delete the diagnostic menu from the workstation by entering the following McIDAS command.

Type: OS "DEL MENU
Press: Enter

This command deletes the menu from the McIDAS F Key interface. Pressing the Esc key displays an error message indicating that the menu file is not installed.

An alternative to this procedure is available on the diagnostic diskette. Insert the diagnostic disk into the floppy disk drive. Then,

Type: **A:DINSTALL**
Press: **Enter**

This batch file deletes the entire diagnostic set (including the data and executable files) from the \MCIDAS\DATA and \MCIDAS\CODE directories.

At any time, you can reinstall the SDA diagnostic menu.

Type: **MAKMNU SDADIAG**
Press: **Enter**

This command recreates the SDA diagnostic menu for the F Key interface.

Low Level Diagnostics

Release 1.1 of the SDA diagnostics is the second release of the Low Level SDA diagnostics set. It is to be used with McIDAS-OS2 releases 5.6 and higher. Release 5.6 of McIDAS-OS2 is the first release to include code for the SSEC Display Adapter (SDA).

The Low Level Diagnostics utility is a menu driven set of programs invoked by executing SDALLD.

The following programs are presented in the main diagnostic menu:

- MCSDXS Diagnostics
- Micro Channel Related Diagnostics
- SDA Frame-level Diagnostics

MCSDXS Diagnostics

The MCSDXS diagnostic tool allows writing to any SDA memory location, allowing complete control of the SDA adapter. Since all SDA control registers and RAMDAC registers are memory-mapped, MCSDXS allows you to write to any SDA control register, DAC register, or video memory location (word). This can be done once, repeatedly at TV rates, or as "fast as possible".

Micro Channel Related Diagnostics

Micro Channel Related diagnostic utilities perform the following functions.

- POKEPOS reads from and writes to the SDA's POS registers.
- ENBLSDA3 enables the SDA adapter which must be enabled before it can be used (typically done once after the machine is booted up).

Frame-level Diagnostics

SDA Frame-level diagnostic utilities perform the following functions.

- GRESTF2 prompts for frame and file information. Then it loads the specified file to the specified SDA frame (0-based frame number). Available image files are $LAX.VIS, $LAX.IR, $NTEST.PAT and $OTEST.PAT.
- GRESTG8 prompts for frame and file information. Then it loads the specified .PIC file to the specified SDA graphics frame (0-based frame number). Available graphics files are $USAMAP.PIC, $WIMAP.PIC and $LAXMAP.PIC.

- DOMOUSE engages the cursor-to-mouse driver, allowing the mouse to be moved and used to determine the position in the frame. Mouse button 2 reports TV line and element. Mouse button 1 (and movement) changes cursor size. Mouse buttons 1 and 2 together disengage the cursor from the mouse.
- DUMPCMEM invokes a utility to examine the contents of the SDA Shadow Segment. It can be used to examine individual values or dump the entire segment (not implemented at this time).
- LDOITALL invokes a utility to load image and graphics frames, enhancement tables, and cursor (completely local).
- LOADETF loads an image enhancement from an ASCII text file (LLDIAGS.IET).
- LOADGET loads a graphics enhancement from an ASCII text file (SDAG.ET).

Installing the Diagnostics

1. Switch to the drive or partition on which the \McIDAS\TOOLS (and possibly \McIDAS\CODE) directories reside (usually the C: drive).

2. Make sure the SDA Diagnostics diskette is in the A: drive. Then,

   Type: **A:INSTSDA**
   Press: **Return**

   Use an optional drive argument if the McIDAS code and data reside on different drives or partitions on your PS/2 hard disk. For example, type A:INSTSDA D if you put the \McIDAS\DATA directory on the D: drive when you installed McIDAS-OS2.

   This installation procedure unpacks the files needed to run Low Level Diagnostics. It puts the executable programs and .PIC files into the \MCVGA directory; it puts appropriate menus and help files into the data directory (\MCIDAS\DATA).

Running the Diagnostics

To run the Low Level Diagnostics program, you must deactivate the SDA handler by doing one of the following.

1. Exit McIDAS by entering the McIDAS-OS2 EXIT command. Or,

2. Allow McIDAS-OS2 to stay active but deactivate the SDA handler by executing the McIDAS-OS2 command, UCU POKE 302 1, to temporarily freeze the SDA handler. When you're finished using the diagnostics, poke back to 0 to thaw.

Once the SDA handler is frozen, do the following.

1. Use the Alt key to switch to another OS/2 session.

2. Make sure you are in the \MCVGA directory. Use CD \MCVGA to accomplish this.

3. Type: **SDALLD**
   Press: **Enter**

You're now able to exercise the Low Level Diagnostics. Menus and a HELP will aid you in performing the tests. Use function key F10 to EXIT a menu level (including the main menu level which gets you out of SDALLD diagnostics entirely). Use function key F12 to display the help for a given menu level.

The Low Level Diagnostics allows you to write to any addressable SDA memory location and read status from the POS (Programmable Option Select) registers in the Micro Channel Interface section of the SDA. The POS registers are the only readable addresses in the SDA. There are no port addresses in the SDA. That is, all addresses are memory mapped. These three categories of SDA memory mapped addresses make up the SDA Memory Map:

- SDA Display Memory
- SDA Control Registers
- RAMDAC (Digital to Analog Converter plus RAM) Registers

SDA Memory Map

The SDA's display memory, control words and RAMDAC registers are memory mapped under OS/2. The SDA's address mapping, control words and DAC registers are defined below.

SDA Display Memory

The SDA's display memory is memory mapped to PS/2 addresses XXC00000$_H$ through XXDFFFFF$_H$. The two MSD hexidecimal digits (XX) are determined during system configuration. From the PS/2's perspective, this memory is a write-only, 16-bit, word-write type. The physical address is computed by adding XXC00000$_H$ to the relative address. The following frame mix is assumed for diagnostic purposes.

Frame Number	Frame Type	Rel. Addr. Decimal	Rel. Addr. Hex	Rel. Mod 2048 Hex Addr.
0	Image	0	00000000	0000
1	Image	307200	0004B000	0096
2	Image	614400	00096000	012C
3	Image	921600	000E1000	01C2
4	Image	1228800	0012C000	0258
5	Image	1536000	00177000	02EE
6	Graphics	1843200	001C2000	0384
			|---*---|	|**|

* Twenty-one (21) address bits relative to the start address.
** Ten (10) bits video offset.

RAMDAC Addressing

Addresses XXBFFFF0$_H$ through XXBFFFF7$_H$ are used for programming and controlling the RAMDACs. The RAMDACs function as a byte-write, write-only type memory.

Interrupt Reset

Writing to address XXBFFFF8$_H$ resets the IRQ11/ or IRQ15/ interrupt to the PS/2.

Control Registers

There are eight 16-bit Control Registers. They are defined as follows:

- XXEFF000H - XXEFF007H are Cursor Control Registers (four 16-bit words).
- XXEFFFF8H - XXEFFFFFH are SDA Control Registers (four 16-bit words).

Control Register Layout

The table below describes the Control Registers.

Address	Data Bits F E D C B A 9 8 7 6 5 4 3 2 1 0	Function
XXEFF000H	unused | Off-set (10 Bits)	Cursor horizontal offset
XXEFF002H	unused | Size (9 Bits)	Cursor horizontal half-size
XXEFF004H	unused | Off-set (10 Bits)	Cursor vertical offset
XXEFF006H	unused | Type | Size (8 Bits)	Cursor Type* and vertical half-size
XXEFFFF8H	Address (16 LSBs)	Frame Start
XXEFFFFAH	unused | Z. F. | unused | F.S.	Frame Start (3 MSBs) Zoom Factor (Bits 8 - A)
XXEFFFFCH	unused | Video Offset	Modulo 2048 of Frame Location (10 Bits)
XXEFFFFEH	G.M. | | Graph. Offset	Modulo 2048 of Frame Location (10 Bits) Bits C - F are Graphics Masks, bits A-B unused

* Cursor Types:
0 = Open Box   4 = Open Box
1 = Filled Box   5 = Transparent
2 = Star Wars   6 = Open Box with Cross Hair
3 = Transparent   7 = Cross Hair

RAMDAC Register Layout

A RAMDAC contains an image and a graphics color enhancement table. The image enhancement table contains 256 24-bit locations; the graphics enhancement table contains 16 24-bit locations. Though data can be read from or written to any location in any table in a RAMDAC, the readback capability in the SDA application is not implemented.

An enhancement table is programmed by first selecting the starting address and then writing three bytes of data. The RAMDAC interprets the first data byte as red data, the second byte as green data and the third byte as blue data. If consecutive locations are being loaded, the RAMDAC automatically increments the storage address after every third byte of data (blue value). Thus, only one start address is required for contiguous locations. As an example, if you want to program image enhancement table locations 100 and 101 (decimal), write 64_H (100 decimal) to address $XXBFFFF0_H$ to set the starting address. Then, write six bytes of data (red, green, blue, red, green, blue) to address $XXBFFFF1_H$ to program the two locations.

The table below shows the SDA memory mapped address for each RAMDAC register, its name and function.

SDA Address	Register Name	Register Function
XXBFFFF0	Write Address	Image palette start address
XXBFFFF1	Color Palette RAM	Image palette data
XXBFFFF2	Pixel Read Mask	not used in SDA
XXBFFFF3	RAM Read Address	not used in SDA
XXBFFFF4	Overlay Write Address	Graphics palette start address
XXBFFFF5	Overlay	Graphics palette data
XXBFFFF6	Reserved	
XXBFFFF7	Overlay Read Address	not used in SDA

Deleting the Diagnostics

You can remove all files associated with the Low Level Diagnostics from your hard drive, once you have finished using them. Insert the diagnostics disk in drive A. Then,

Type: A:REMOVSDA
Press: Enter

Use an optional drive argument if you did not install the Low Level Diagnostics on drive C. For example, type A:INSTSDA D: if you put the Low Level Diagnostics on the D: drive when you installed them.

Supplemental Data

This Supplemental Data section contains:

- an explanation of the PAL equation listings
- the symbols and abbreviations used in the equations
- two PAL listings
- Schematic drawing 6450-0639, Revision F, sheets 1 through 4
- Assembly drawing 6450-0640

PAL Equation Listings

The listings for the logic equations used in the Programmable Array Logic devices are explained below.

Title

Each device has a printout that begins with the title page. On this page the device is called out by its location and device name. The Title also includes a functional name for the PAL, the design engineer's initials and the revision date of the PAL logic.

Identification

The Identification section lists the device location and the type of PAL.

Module

The Module section gives the program module name.

Declarations

The Declarations section lists the logic conventions and format for the signal names. The signal names are in uppercase letters and numerals.

The first group of signal names refers to the power and ground rails. The respective pin numbers that make these connections are given in the line below the signal names.

The second group of signal names refers to the input signals received by the device. The respective pin numbers assigned to receive these inputs are given in the line below the signal names.

The third group of signal names refers to the outputs from the PAL. The respective pin numbers assigned to each output are given in the line below the signal names.

Equations

The logic equations used to generate signals are shown in their highest order form.

Symbols and Abbreviations

The following is an example of a PAL equation.

```
QA    :=    !S2 & S1 & !S0          "(A comment may be shown here.)
          # S2 & D3 & !D1 & !D0
          # S2 & D1 & !D0
```

Below is an explanation of the symbols and abbreviations used in the equation above and an explanation of address terms used in other equations found in this section.

Symbol	Explanation
!	The one's complement of, e.g. !WR = $(\overline{WR})$
:	A register latched signal; valid on the rising edge of the register clock
&	Logical AND
#	Logical OR
ABC_	An active low signal (PAL equations)
ABC/	An active low signal (Schematic Drawings)
Address=^hE1	Combined terms simplify to a value greater than or equal to E1 Hexadecimal *
Address==^hE1	Combined terms are equal to E1 Hexadecimal *

*** Terms combined to form address values are listed in the Equations Section, Address = line.**

The listings for Programmable Array Logic devices used in the SDA follow.

P22V10 Located at U1

Title

Interrupt and Upper Data Bus Latch
UW SSEC MADISON WI 10/29/91

Identification

Device Location	U1
Device Type	P22V10

Module

SDA1

Declarations

```
TRUE, FALSE = 1, 0
H, L = 1, 0
X, Z, CK = .X., .Z., .C.

GND, VCC
    PIN 12, 24

D8, D9, D10, D11, D12, D13, D14, D15, CDS_, CMD_, BDE_, FI
    PIN 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13

OD8, OD9, OD10, OD11, OD12, OD13, OD14, OD15, GSE_, IRQ15_
    PIN 14, 15, 16, 17, 18, 19, 20, 21, 22, 23

IDATA = [D15, D14, D13, D12, D11, D10, D9, D8]
ODATA = [OD15, OD14, OD13, OD12, OD11, OD10, OD9, OD8]
```

Equations

```
enable IRQ15_= FI & !BDE_       "TRI-STATE

IRQ15_ =    FALSE               "ACTIVE LOW INTERRUPT

!GSE_  =    !CDS_ & CMD_        "TRANSPARENT LATCH: SET
          # !GSE_ & !CMD_       "HOLD (CDS occurs 1 cycle early)
          # !GSE_ & !CDS_       "CATCH: TIES THE MINTERMS

OD15   =    D15 & !CMD_         "TRANSPARENT LATCH: SET
          # OD15 & CMD_         "HOLD
          # OD15 & D15          "CATCH: TIES THE MINTERMS

OD14   =    D14 & !CMD_         "TRANSPARENT LATCH: SET
          # OD14 & CMD_         "HOLD
          # OD14 & D14          "CATCH: TIES THE MINTERMS

OD13   =    D13 & !CMD_         "TRANSPARENT LATCH: SET
          # OD13 & CMD_         "HOLD
          # OD13 & D13          "CATCH: TIES THE MINTERMS
```

```
OD12  =  D12 & !CMD_        "TRANSPARENT LATCH: SET
       # OD12 & CMD_        "HOLD
       # OD12 & D12         "CATCH: TIES THE MINTERMS

OD11  =  D11 & !CMD_        "TRANSPARENT LATCH: SET
       # OD11 & CMD_        "HOLD
       # OD11 & D11         "CATCH: TIES THE MINTERMS

OD10  =  D10 & !CMD_        "TRANSPARENT LATCH: SET
       # OD10 & CMD_        "HOLD
       # OD10 & D10         "CATCH: TIES THE MINTERMS

OD9   =  D9 & !CMD_         "TRANSPARENT LATCH: SET
       # OD9 & CMD_         "HOLD
       # OD9 & D9           "CATCH: TIES THE MINTERMS

OD8   =  D8 & !CMD_         "TRANSPARENT LATCH: SET
       # OD8 & CMD_         "HOLD
       # OD8 & D8           "CATCH: TIES THE MINTERMS
```

P22V10 Located at U30

Title

FIFO Load and Lower MC Data Bus Latch
UW SSEC MADISON WI 10/29/91

Identification

```
Device Location   U30
Device Type       P22V10
```

Module

SDA2

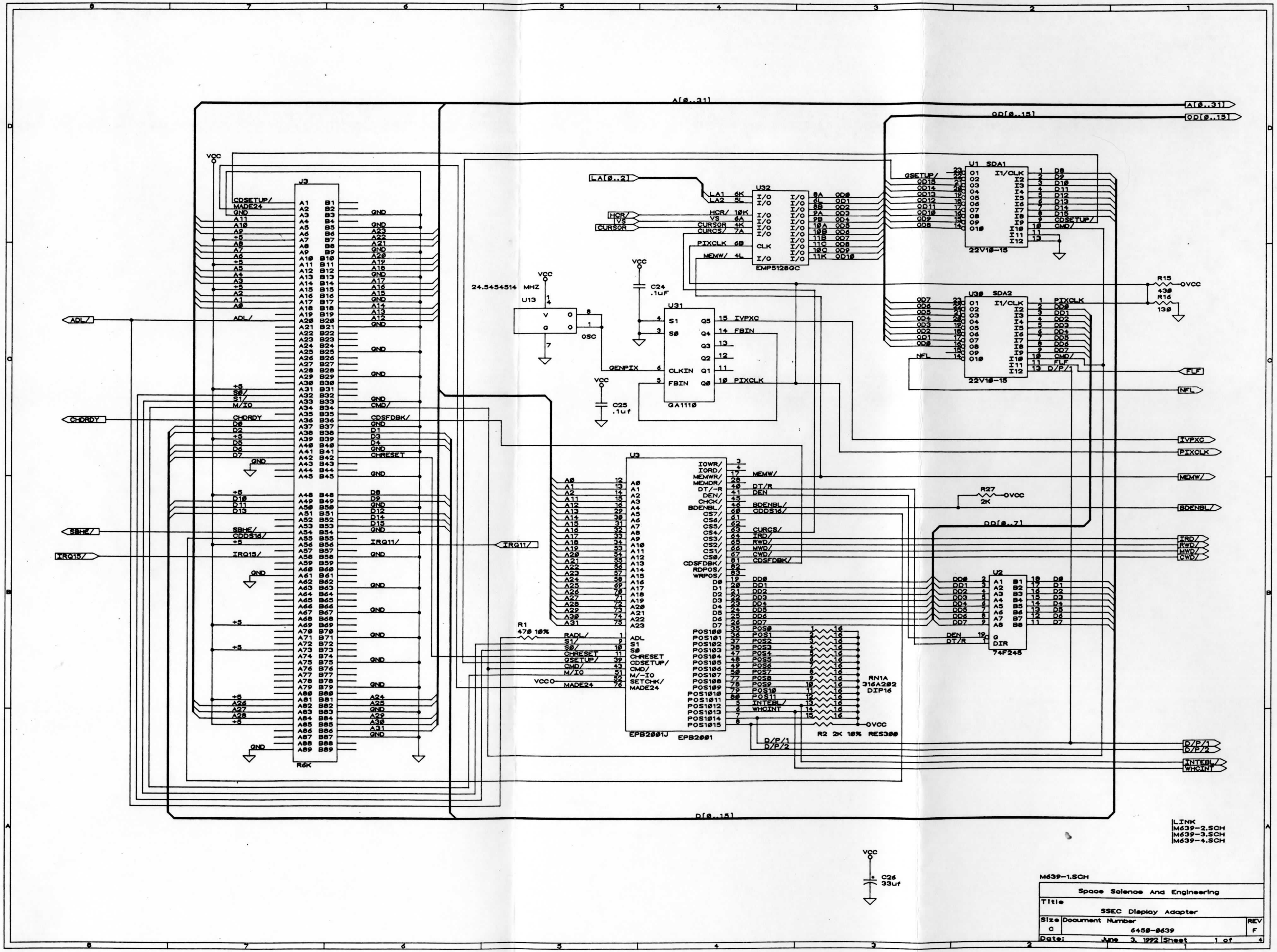
Declarations

```
TRUE, FALSE = 1, 0
H, L = 1, 0
X, Z, CK = .X., .Z., .C.

GND, VCC
    PIN 12, 24

CLK, D0, D1, D2, D3, D4, D5, D6, D7, CMD_, FLF, DPG_
    PIN 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13

IFL_, OD0, OD1, OD2, OD3, OD4, OD5, OD6, OD7
    PIN 14, 16, 17, 18, 19, 20, 21, 22, 23

IDATA = [D7, D6, D5, D4, D3, D2, D1, D0]
ODATA = [OD7, OD6, OD5, OD4, OD3, OD2, OD1, OD0]
```

Equations

```
!IFL_   =   !CLK & FLF           "FIFO LOAD

OD0     =   D0 & !CMD_           "TRANSPARENT LATCH: SET
          # OD0 & CMD_           "HOLD
          # OD0  & D0            "CATCH: TIES THE MINTERMS

OD1     =   D1 & !CMD_           "TRANSPARENT LATCH: SET
          # OD1 & CMD_           "HOLD
          # OD1  & D1            "CATCH: TIES THE MINTERMS

OD2     =   D2 & !CMD_           "TRANSPARENT LATCH: SET
          # OD2 & CMD_           "HOLD
          # OD2  & D2            "CATCH: TIES THE MINTERMS

OD3     =   D3 & !CMD_           "TRANSPARENT LATCH: SET
          # OD3 & CMD_           "HOLD
          # OD3  & D3            "CATCH: TIES THE MINTERMS

OD4     =   D4 & !CMD_           "TRANSPARENT LATCH: SET
          # OD4 & CMD_           "HOLD
          # OD4  & D4            "CATCH: TIES THE MINTERMS
```

```
OD5    =   D5 & !CMD_          "TRANSPARENT LATCH: SET
         # OD5 & CMD_          "HOLD
         # OD5  & D57          "CATCH: TIES THE MINTERMS

OD6    =   D6 & !CMD_          "TRANSPARENT LATCH: SET
         # OD6 & CMD_          "HOLD
         # OD6  & D67          "CATCH: TIES THE MINTERMS

enable OD7 = DPG_              "TRISTATE OD7 DURING
                               "LCA CONFIG.

OD7    =   D7 & !CMD_          "TRANSPARENT LATCH: SET
         # OD7 & CMD_          "HOLD
         # OD7  & D7           "CATCH: TIES THE MINTERMS
```



LINK
M639-2.SCH
M639-3.SCH
M639-4.SCH

M639-1.SCH

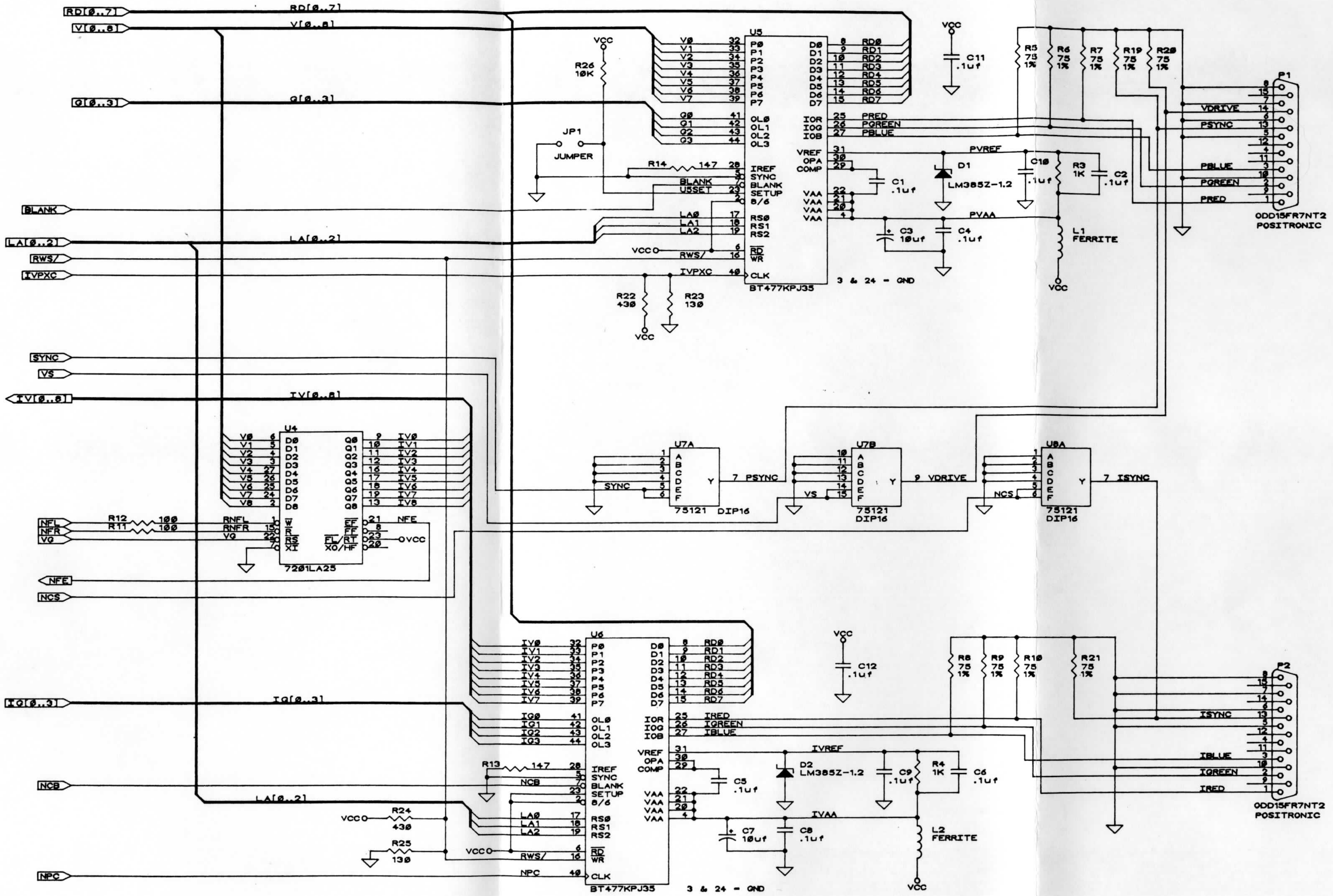
Space Science And Engineering

Title

SSEC Display Adapter

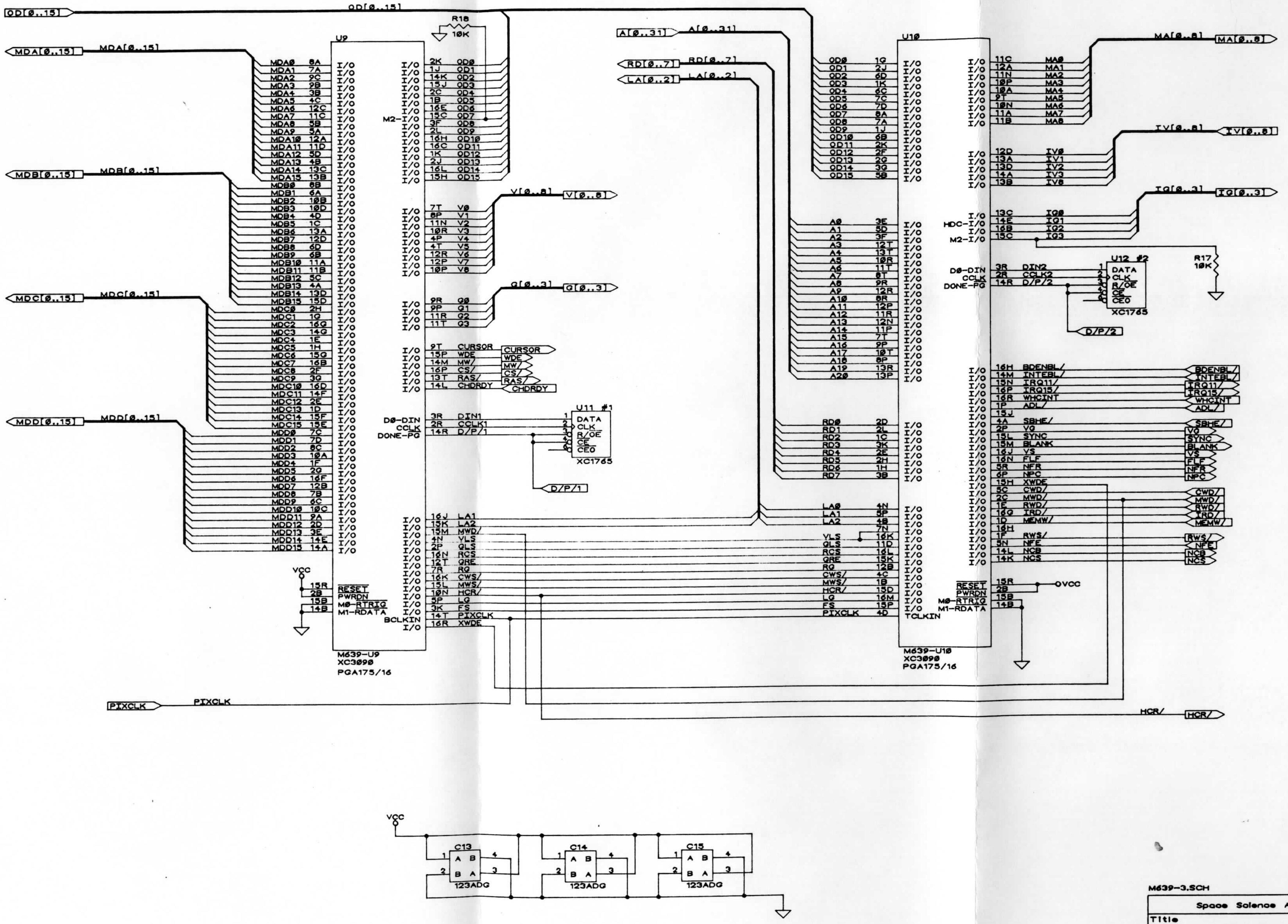
Size	Document Number	REV
C	645Ø-Ø639	F

Date: June 3, 1992 Sheet 1 of 4



M639-2.SCH

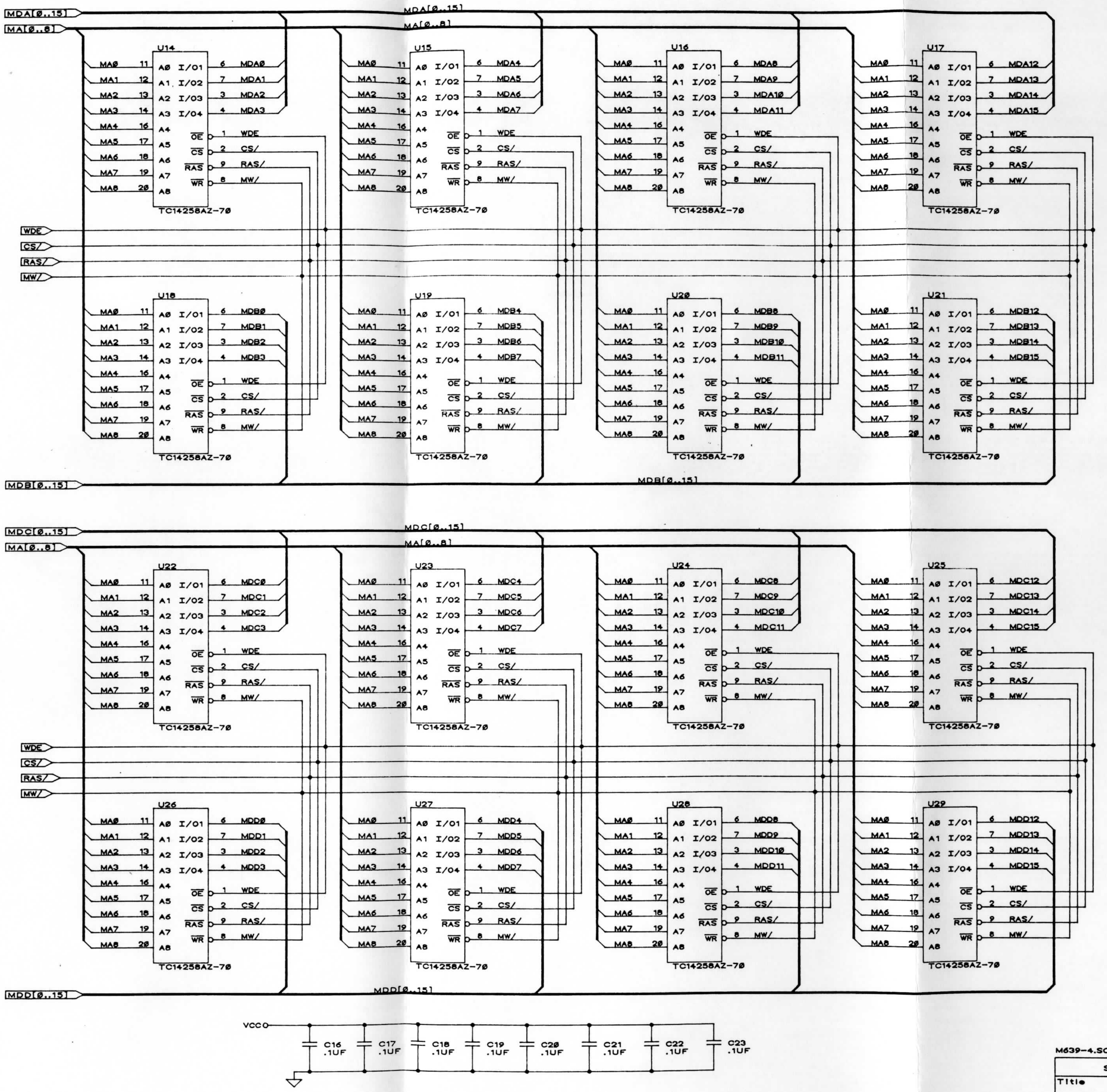
Space Science And Engineering		
Title SSEC Display Adapter		
Size C	Document Number 6450-0639	REV F
Date: June 3, 1992	Sheet 2 of 4	



M639-3.SCH

Space Science And Engineering

Title	SSEC Display Adapter	
Size C	Document Number 645Ø-Ø639	REV F
Date: June 3, 1992	Sheet 3 of 4	



M639-4.SCH

Space Science And Engineering

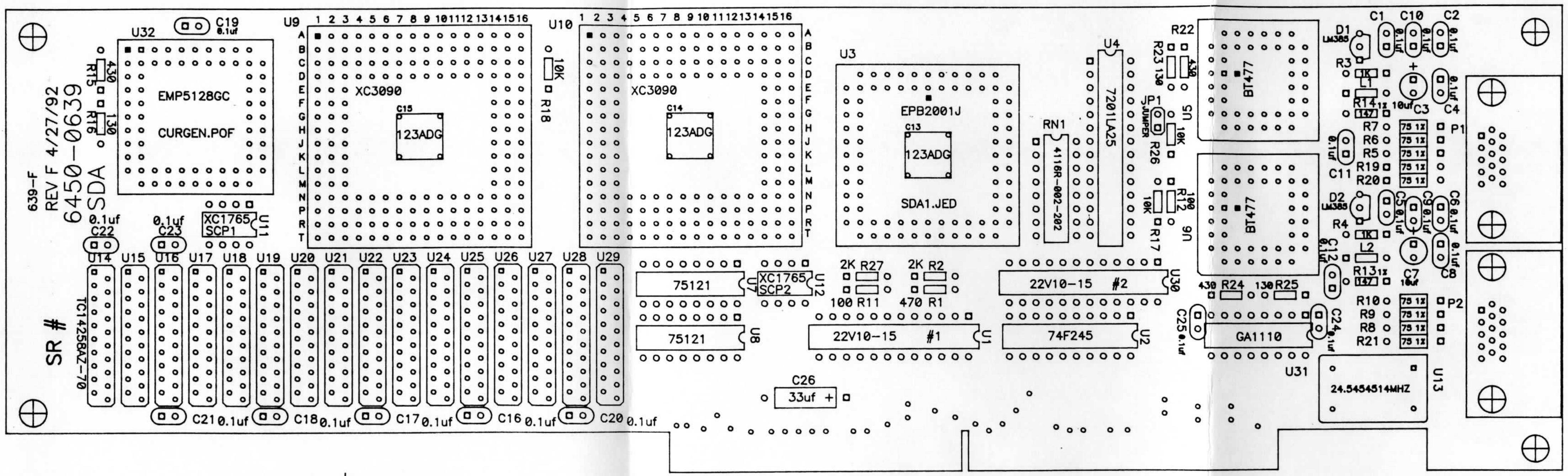
Title

SSEC Display Adapter

Size	Document Number	REV
C	6450-0639	F

Date: June 3, 1992 Sheet 4 of 4

TOP ASSY M639
TOP SILK M639



SPACE SCIENCE AND ENGINEERING
SSEC DISPLAY ADAPTER
6450-0640   4/27/92 REV F

Appendix A

PS/2 Architecture

The PS/2 Model 70 or 80 computer has a number of hardware blocks that are interconnected by the microprocessor's address, data and control lines. Collectively, these lines are called the Micro Channel. Each hardware block is assigned a unique block of memory and/or I/O address space. The PS/2 hardware blocks are:

- ROM (Read Only Memory)
- DMA (Direct Memory Access)
- Math Coprocessor (optional)
- RAM (Random Access Memory)
- Diskette Control
- Video Graphics Control
- RS-232 Serial Port
- Parallel Port
- Interrupt Controller
- Timer (Three Channels)
- Keyboard and Mouse Control
- Clock/Calender
- Spare Micro Channel Connectors (3 and 8 for Models 70 and 80, respectively)

Micro Channel Characteristics

The PS/2 uses an Intel 80386 32-bit microprocessor. For backward compatibility with less powerful microprocessors, the 80386 can operate with 8-, 16-, 24- or 32-bit data words (1 - 4 bytes). Further, the 80386 can transfer data to or from other devices over a 16- or 32-bit wide data bus. Bus width control is accomplished via an input pin controlled by the memory or I/O mapped hardware block being addressed (i.e., SDA card).

80386 Data Lines				16-Bit Micro Channel Bus Signals			
(Upper Bus) Byte 4 D31-24	Byte 3 D23-16	(Lower Bus) Byte 2 D15-8	Byte 1 D7-0	BHE *	A1	A0	Comments
			S	1	0	0	
		S		0	0	1	
	S			1	1	1	
S				0	1	1	
		D	D	0	0	0	
	D	D		0	0	1	Double Bus Cycle: Byte 2, then Byte 3
D	D			0	1	0	
	T	T	T	0	0	0	Double Bus Cycle: Bytes 1-2, then Byte 3
T	T	T		0	0	1	Double Bus Cycle: Byte 2, then bytes 3-4
Q	Q	Q	Q	0	0	0	Double Bus Cycle: Bytes 1-2 then 3-4

S = active byte in single-byte operands
D = active bytes in double-byte operands
T = active bytes in triple-byte operands
Q = active bytes in quadruple-byte operands

* BHE is asserted (active) when zero (0)

Table A-1. Micro Channel Data Transfers

The data bus component of the Micro Channel is 32 bits wide for maximum internal data throughput. However, devices connected to the Micro Channel may be either 16-bit (such as the SDA card) or 32-bit devices. When a data word longer than 16 bits is transferred to or from a 16-bit device, the 80386 performs multiple bus cycles.

The 80386 has 30 address lines (A2 - A31) and four byte enable lines (BE0 - BE3). BE0 - 3 are generated by internally decoding A0 and A1. BE0 - 3 identify the active bytes on the 32-bit Micro Channel data bus. Via hardware external to the 80386, A0 and A1 are reconstructed and are part of the Micro Channel's address signals. Also, BE0 - 3 are part of the Micro Channel's control signals. Therefore, the Micro Channel's address bus is 32 bits wide (A0 - A31), and its lower two bits (A0 - A1) function as a byte address for memory reads and writes.

For added flexibility and memory utilization, two additional features are provided. First, multiple byte operands may be split between two physical words in the memory array. For example, two 16-bit and four 8-bit operands (eight bytes total) can be stored in two physical words (8 byte capacity) in the memory array. Thus, an operand can start with any byte within the physical word provided all bytes in the multiple byte operand are contiguous. The 24- and 32-bit modes are not used in the workstation application. Second, operands or portions of operands that reside only in the upper half of a 32-bit physical word are duplicated on the lower 16 bits of the Micro Channel's data bus. This results in a higher data throughput when the Micro Channel is forced into the 16-bit mode because data does not have to be translated to the lower half of the bus by the slave device.

Table A-1 shows each valid operand location within a physical word. Byte High Enable (BHE) is a Micro Channel signal that, together with A0 and A1, describes which data bytes are read from or written to the Micro Channel device while in the 16-bit mode. When operands extend across the upper/lower bus boundary, the 80386 sends or receives the active lower bus portion first. Then it performs an additional bus cycle to send or receive the upper bus portion.

As an example of the multiple bus cycle concept, consider the first triple-byte operand in Table A-1. The BHE, A1 and A0 code (0, 0 and 0) is the same as the code for the first double-byte and quadruple-byte operands. The code is the same because both lower bus bytes are active for each operand. That is, the code shown for each double bus cycle operand is the code for the first cycle. Once the first bus cycle is completed, a second bus cycle is executed if the operand transfer is not complete. In the example triple-byte operand, the BHE, A1 and A0 for the second bus cycle are 1, 1 and 0, respectively. Note in Table A-1 that this code is the same as the code for a single-byte operand located on the 80386's D16 - D23 data lines.

The data bus component of the Micro Channel is 32 bits wide for maximum internal data throughput. However, devices connected to the Micro Channel may be either 16-bit (such as the 8DA card) or 32-bit devices. When a data word longer than 16 bits is transferred to or from a 16-bit device, the 80386 performs multiple bus cycles.

The 80386 has 30 address lines (A2 - A31) and four byte enable lines (BE0 - BE3). BE0 - 3 are generated by internally decoding A0 and A1. BE0 - 3 identify the active bytes on the 32-bit Micro Channel data bus. Via hardware external to the 80386, A0 and A1 are reconstructed and are part of the Micro Channel's address signals. Also, BE0 - 3 are part of the Micro Channel's control signals. Therefore, the Micro Channel's address bus is 32 bits wide (A0 - A31), and its lower two bits (A0 - A1) function as a byte address for memory reads and writes.

For added flexibility and memory utilization, two additional features are provided. First, multiple byte operands may be split between two physical words in the memory array. For example, two 16-bit and four 8-bit operands (eight bytes total) can be stored in two physical words (8 byte capacity) in the memory array. Thus, an operand can start with any byte within the physical word provided all bytes in the multiple byte operand are contiguous. The 24- and 32-bit modes are not used in the workstation application. Second, operands or portions of operands that reside only in the upper half of a 32-bit physical word are duplicated on the lower 16 bits of the Micro Channel's data bus. This results in a higher data throughput when the Micro Channel is forced into the 16-bit mode because data does not have to be translated to the lower half of the bus by the slave device.

Table A-1 shows each valid operand location within a physical word. Byte High Enable (BHE) is a Micro Channel signal that, together with A0 and A1, describes which data bytes are read from or written to the Micro Channel device while in the 16-bit mode. When operands extend across the upper/lower bus boundary, the 80386 sends or receives the active lower bus portion first. Then it performs an additional bus cycle to send or receive the upper bus portion.

As an example of the multiple bus cycle concept, consider the first triple-byte operand in Table A-1. The BHE, A1 and A0 code (0, 0 and 0) is the same as the code for the first double-byte and quadruple-byte operands. The code is the same because both lower bus bytes are active for each operand. That is, the code shown for each double bus cycle operand is the code for the first cycle. Once the first bus cycle is completed, a second bus cycle is executed if the operand transfer is not complete. In the example triple-byte operand, the BHE, A1 and A0 for the second bus cycle are 1, 1 and 0, respectively. Note in Table A-1 that this code is the same as the code for a single-byte operand located on the 80386's D16 - D23 data lines.

Appendix B

Video Scan Timing Characteristics

This appendix describes the U. S. TV timing standards (interlaced scan) and progressive scan characteristics.

Interlaced Timing Characteristics

U.S. TV timing characteristics are the result of standards adopted by the NTSC (National Television Standards Committee). These standards are tightly adhered to by the TV broadcasting industry. A TV frame (one complete image) consists of 525 horizontal lines. Frames are generated every 1/30 second for a black and white TV and every 1/29.97 second for a color TV.

TV frames are interlaced, i.e., they consist of two fields designated odd and even. Each field has 262-1/2 horizontal lines. The even field consists of even lines only; the odd field consists of odd lines only. A field is transmitted in one-half the frame time (i.e., 1/60 second for black/white or 1/59.94 for color). The first field is written to the display, scanning every other line. Then the remaining field is written to the display, filling in between the lines of the first field. This results in an effective flicker of a 60 Hz frame rate while requiring a bandwidth of a 30 Hz frame rate.

TV horizontal and vertical sweep circuits must be synchronized to the broadcast signal. This is accomplished with horizontal and vertical synchronization signals that are transmitted as part of the TV signal. When the scan reaches the right side or bottom of the screen, the sweep circuits must return the electron beam rapidly back to the left side or top of the screen. During these return times (called flyback or retrace), the electron beam must be turned off. This is accomplished with horizontal and vertical blanking signals which shut off the beam during retrace and blank the synchronization pulse periods.

The vertical retrace time is approximately 21 horizontal lines per field. Thus, about 42 horizontal lines are required for each frame for the vertical retrace (2 fields/frame x 21 lines/field = 42). This leaves 525 lines minus 42 lines for a total of 483 visible horizontal lines. The horizontal blanking period is approximately 17% of the total horizontal period.

Progressive Scan Timing Characteristics

Progressive scan frames are noninterlaced. Thus, a frame and a field are the same thing. To prevent flicker, the frame rate is doubled from 30 Hz for interlaced TV frames to 60 Hz for the progressive scan. This requires doubling the horizontal scan rate. Instead of 15,734 horizontal lines/second (29.97 frames/second x 525 lines/frame = 15,734), the SDA card generates 31,468 horizontal lines/second. Thus, to maintain the same horizontal pixel resolution, data must be written twice as fast for a progressive scan as it is for the interlaced scan. Vertical and horizontal blanking ratios are maintained at 42 horizontal lines and 17% of each horizontal line, respectively.

Appendix C

Programmable Gate Array Background

The Master Controller is based on the Xilinx Programmable Gate Array logic XC30XX family. Each chip in this family contains several thousand gates which make up several hundred Configurable Logic Blocks (CLB) organized as an array. The SDA card uses two XC3090 gate arrays. Each XC3090 has 9000 gates that make up 320 CLBs arranged in a 20 by 16 array.

Surrounding the CLB array is a ring of programmable I/O Blocks (IOBs), each connecting to one of the chip's external pins. Each IOB pin can be independently programmed as an input, an output (tri-state) or a bidirectional pin.

A programmable interconnect network connects the IOBs to the CLBs. This network consists of two layers of lines. One layer runs horizontally between rows of CLBs; the other layer runs vertically between columns of CLBs. At the intersections of these two layers are programmable bidirectional switches that allow programmable interconnections between the horizontal and vertical lines. Each IOB and CLB has programmable unidirectional switches that allow its inputs and outputs to be connected to the horizontal and vertical lines adjacent to it. Thus, by proper programming, any CLB input or output can be interconnected with any IOB or other CLB input or output.

The high degree of flexibility provided by configurable logic and interconnects allows groups of CLBs to be interconnected to form complex internal logic blocks without consuming large numbers of chip pins for external interconnections. Thus, IOB pins are only required to input signals required by this chip and output signals required by external circuitry.

Each interconnect switch, each IOB and CLB connection switch, and the configurable logic in each IOB and CLB are controlled by external configuration program data. During power-up, this data is downloaded into static configuration RAM in the chip. The XC30XX family of chips can be downloaded from an EEPROM, EPROM, ROM, floppy disk or hard disk in any of several modes. For simplicity, the SDA card uses a compatible serial PROM designed and manufactured by Xilinx especially for its Programmable Gate Arrays. During power-up, this data is serially clocked into the chip by an internal load clock. The XC3090 requires 64,200 bits to fully program its configurable logic and interconnects.

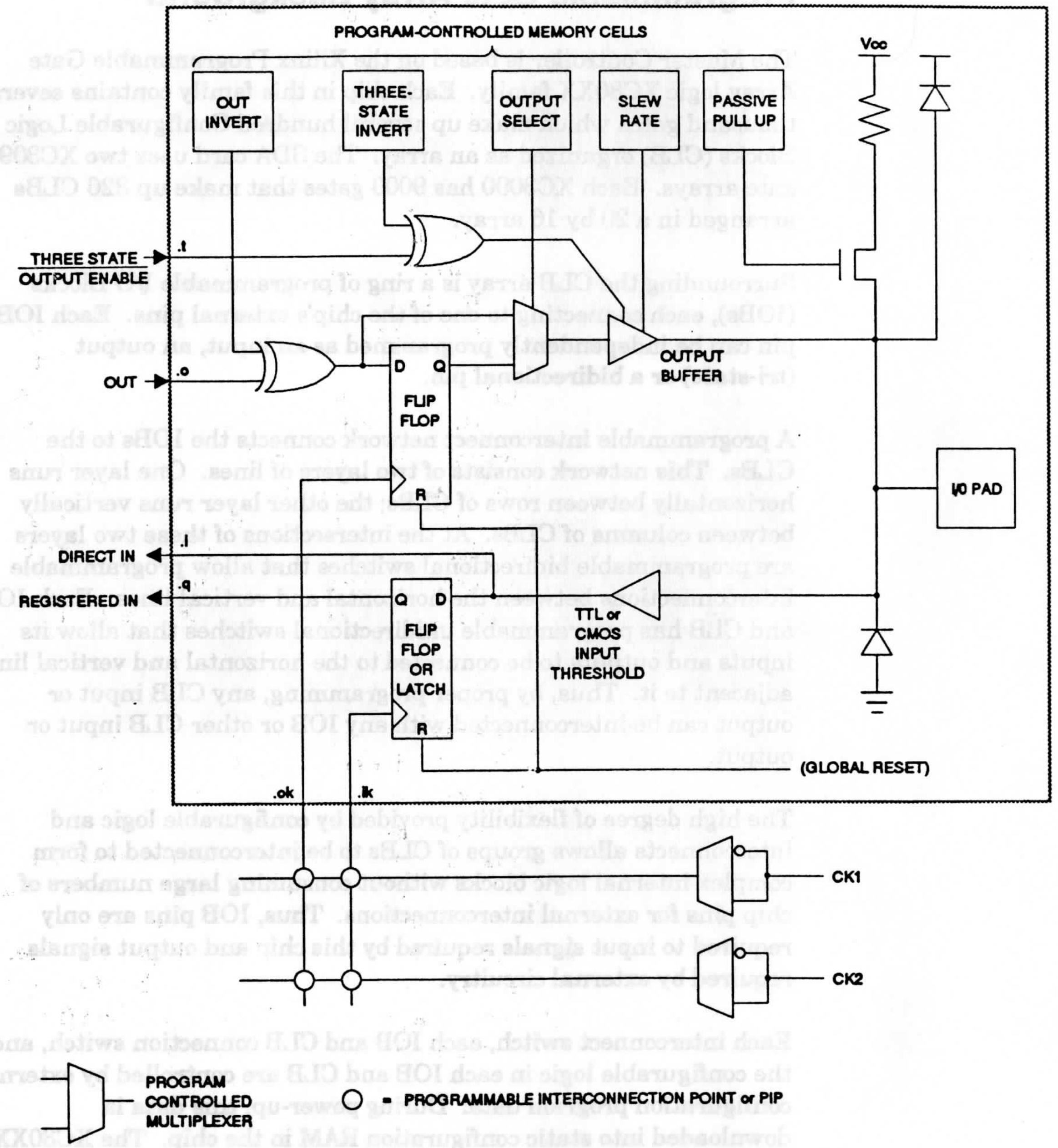


Figure Courtesy of XILINX, Inc.

Figure C-1. IOB Functional Block Diagram

Input/Output Block (IOB)

Refer to Figure C-1 on the adjacent page. The IOB provides an interface between the CLB and the external package pin. IOB input/output pins connect to two paths in the IOB, an input path and an output path. The input path contains a translation buffer that converts CMOS and TTL levels to internal logic levels. The output of the translation buffer drives a D-type latch. The output of the latch and the direct output of the translation buffer are output to the interconnect bus.

The output path passes CLB outputs to the external package pins. In the IOB logic, these signals first pass through an Exclusive-OR gate that functions as a programmable inverter (one of the Configuration bits drives the other gate input). Thus, depending on the Configuration bit value ("one" or "zero"), this gate either inverts the CLB output or passes it uninverted to a multiplexer and D-Type latch. This latch's output drives the other multiplexer input. The multiplexer's output is selected via one of the Configuration bits. Thus, the output of the multiplexer is the direct or inverted and latched or unlatched CLB output. The multiplexer's output drives a Configuration controlled tri-state buffer that drives the output pin. This buffer's tri-state control is driven by the interconnect bus. IOB logic and a configuration bit allow this input to enable the buffer with either a high or low input. Finally, the slew rate of the buffer can be controlled by a Configuration bit. Because CMOS logic loading is primarily capacitive, the output drive is a function of slew rate. Chip power consumption can be reduced by decreasing the slew rate of the noncritical outputs.

Configurable Logic Blocks (CLB)

Refer to Figure C-2 on the next page. The CLB consists of a five-input combinatorial logic block that drives output latches and multiplexers. The combinatorial logic block operates in one of three modes selected via Configuration bits. Mode F generates any function of five variables. Mode FG generates two functions of up to four variables each. Mode FGM (FG Multiplexed) uses four variables to generate two outputs of up to four variables each. These outputs drive a multiplexer whose output is controlled by the fifth input variable. Regardless of the mode, the combinatorial logic block has two outputs (F and G). If the selected mode generates only one function (i.e., Mode F or FGM), both logic block outputs are driven by the same function.

Programmable Interconnect Network

Refer to Figure C-3 on page C-5. The Interconnect Network connects the CLBs to other CLBs and IOBs.

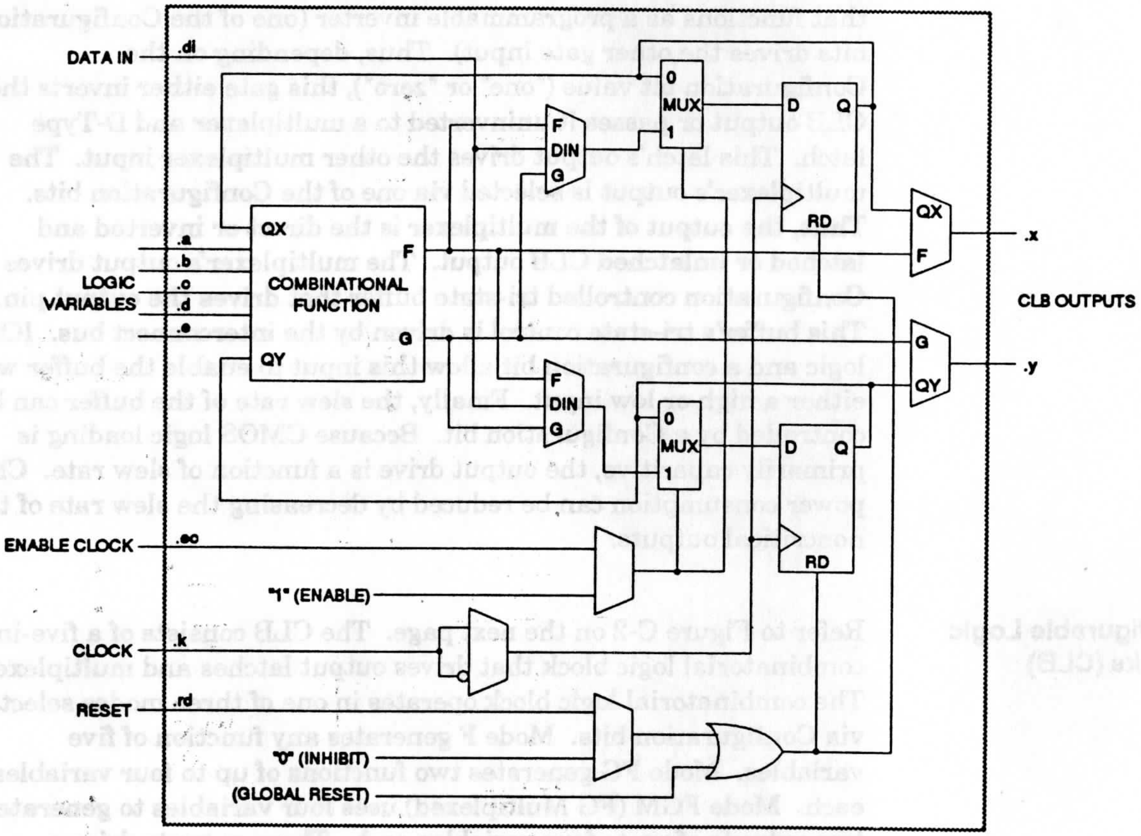


Figure Courtesy of XILINX, Inc.

Figure C-2. CLB Functional Block Diagram

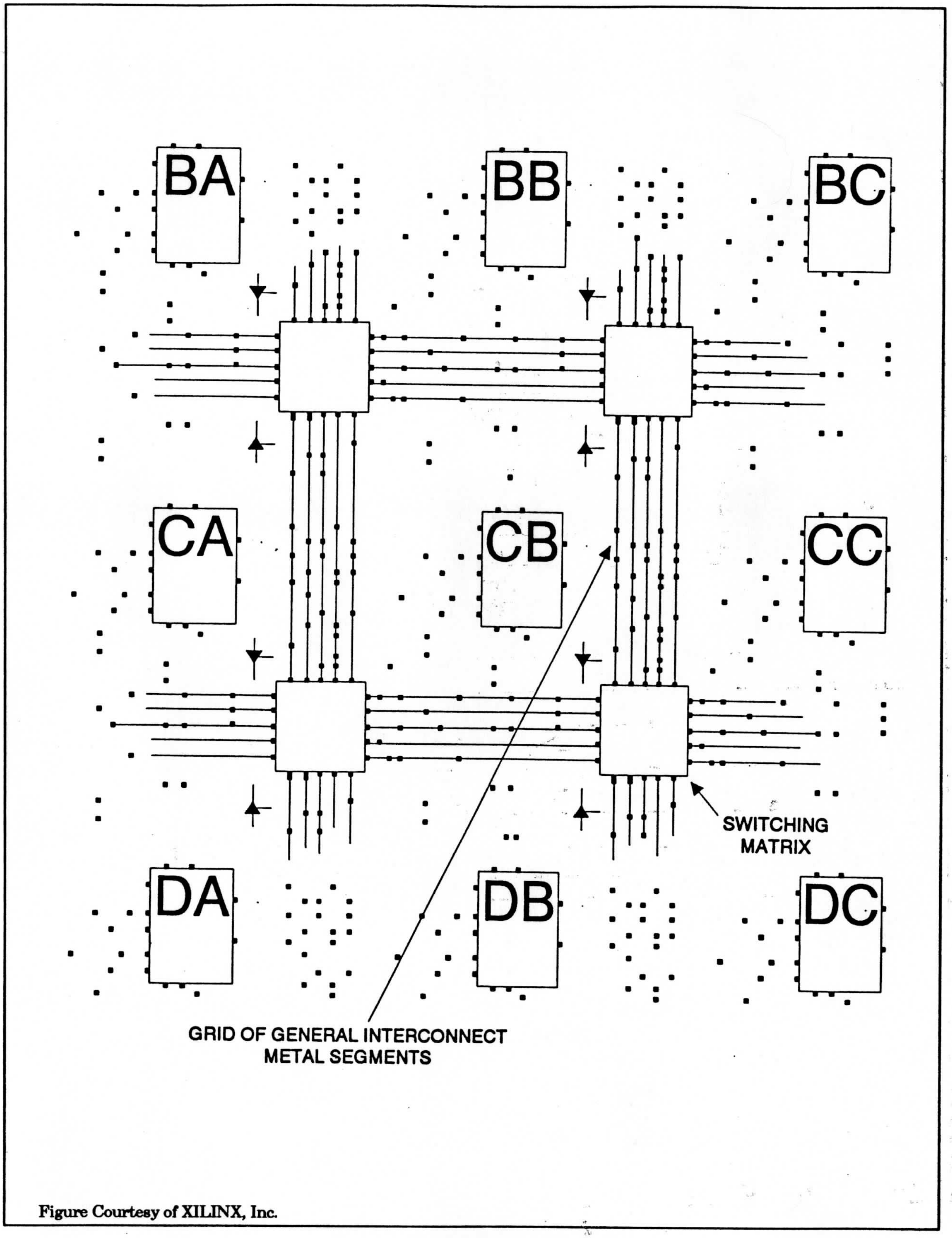


Figure Courtesy of XILINX, Inc.

Figure C-3. Interconnect Network

Appendix D

PS/2 Micro Channel Control Signals

The Micro Channel control signals are:

- ADL/ (Address Decode Latch)
- CDDS16/ (Card Data Size 16)
- SBHE/ (System Byte High Enable)
- MADE24 (Memory Address Enable 24)
- M/IO (Memory/ Input Output)
- S0/ S1/ (Status bits 0 and 1)
- CMD/ (Command)
- CDSFDBK/ (Card Selected Feedback)
- IRQ15/ (Interrupt Request 15)
- CDSETUP/ (Card Setup)
- CHRESET (Channel Reset)

ADL/

ADL/ (Address Decode Latch) is driven by the PS/2's microprocessor. It is used as a latch enable to latch valid addresses and status bits.

CDDS16/

CDDS16/ (Card Data Size 16) is an output of the SDA card and an input to the Micro Channel. This signal tells the Micro Channel whether the address currently on the address bus is addressing an 8- or 16-bit data location. CDDS16/ is active low (asserted) during 16-bit operations.

SBHE/

SBHE/ (System Byte High Enable) is an input to the SDA card and an output of the Micro Channel. This signal goes low when the Micro Channel is transferring data on the upper eight bits of the 16-bit data bus. It is used with address bus bits A0 and A1 to distinguish between the upper byte (D8-D15) and lower byte (D0-D7).

MADE24

MADE24 (Memory Address Enable 24) is an output of the Micro Channel and an input to the SDA card. MADE24 is inactive low for addresses above 16M. That is, MADE24 is low if one or more of the upper eight address lines (A24-A31) are high.

M/IO

M/IO (Memory/ Input Output) is an output from the Micro Channel and an input to the SDA card. It distinguishes a memory cycle from an I/O cycle. When M/IO is high, a memory cycle is in progress; when M/IO is low, an I/O cycle is in progress.

S0/ and S1/

S0/ and S1/ (Status bits 0 and 1) are cycle status outputs of the Micro Channel and inputs to the SDA card. These signals are used with M/IO to indicate memory and I/O read and write cycles. S0/ and S1/ indicate read or write; M/IO indicates memory or I/O. Table D-1 below shows the possible cycles.

M/IO	S0/	S1/	Function
0	0	0	undefined
0	0	1	I/O Write
0	1	0	I/O Read
0	1	1	undefined
1	0	0	undefined
1	0	1	Memory Write
1	1	0	Memory Read
1	1	1	undefined

Table D-1. I/O and Memory Transfer Controls

CMD/

CMD/ (Command) is an output from the Micro Channel and an input to the SDA card. It defines when data is valid on the Micro Channel's data bus. Write data must be valid throughout the CMD/ active period; read data goes valid after the leading falling edge of CMD/ and remains valid until after CMD/ goes inactive. The rising trailing edge of this signal also defines the end of the current bus cycle.

CDSFDBK/

CDSFDBK/ (Card Selected Feedback) is an output of the SDA card and an input to the Micro Channel. The SDA card pulls this signal low when it is addressed. It serves as an acknowledgement to the PS/2's microprocessor that the current address is a valid SDA address. Note, the SDA card does not drive this signal during the times when CDSETUP/ is low.

IRQ15/

IRQ15/ (Interrupt Request 15) is an output of the SDA card and an input to the Micro Channel. This line signals the PS/2's microprocessor that the SDA card has begun a vertical blanking period and is now able to accept data. This signal is driven by the Master Controller's FI output.

CDSETUP/

CDSETUP/ (Card Setup) is an output of the Micro Channel and an input to the SDA card. Each Micro Channel Adapter connector has its own CDSETUP/ line. The PS/2 activates the SDA's CDSETUP/ line during system configuration and error recovery. The PS/2 reads the SDA's ID code and writes configuration variables to the programmable sections of the Micro Channel Interface using this signal.

CHRESET

CHRESET (Channel Reset) is an output of the Micro Channel and an input to the SDA card. The PS/2 uses this signal to reset or initialize all adapter cards (i.e., SDA card) during a power-on sequence or low voltage.