NASA-CR-197001

FINAL
IN-60-CR
OCIT.
28466
54P

SODR Memory Control Buffer Control ASIC

Final Report

Principal Investigator: Dr. Robert F. Hodson

Department of Physics and Computer Science

Period: May 18, 1992 through July 31, 1994

CHRISTOPHER NEWPORT UNIVERSITY
NEWPORT NEWS, VIRGINIA 23606-2998

NASA Research Grant NUMBER NAG-1-1439

November 2, 1994

Table of Contents

I. Summary of Research

The SODR Memory Buffer Control ASIC Research Grant is part of an effort to build a Spacecraft Optical Disk Recorder (SODR). The SODR system is designed to be a state of the art mass storage system for future NASA spaceflight missions requiring high rate and large capacity storage systems. The system goal for the SODR project is to design an expandable system with a 1.2 TByte capacity, a 2.4 GBit/sec data transfer rate and 250 msec access time.

The Memory Buffer Control (MBC) ASIC is a component in the SODR system that performs two primary functions: (1) buffering data to prevent loss of data during disk access times

(2) converting data formats from a High Performance Parallel Interface Format (HIPPI) to a Small Computer Systems Interface Format (SCSI II).

In addition to designing a system to meet the functional specifications of the SODR system, developing an ASIC design capability at NASA, and the dissemination of information for potential commercial applications were projects goals.

In summary, the following results/activities/devices are a direct result of the SODR Memory Buffer Control ASIC research grant:

1) Ten 144 pin, 50 MHz CMOS ASICs were designed, fabricated and tested to implement the Memory Buffer Control Function.

2) Results of this research effort were published at two conferences:

Robert F. Hodson, Stephen Campbell, *An ASIC Memory Buffer Controller for a High Speed Disk System,* 5th NASA Symposium on VLSI Design.

Glenn D. Hines, Stephen G. Jurczyk, Robert F. Hodson, *A Spacecraft Mass Storage Optical Disk System,* Twelfth IEEE Symposium on Mass Storage Systems.

3) The standard cell ASIC design process, with NASA tools for front end design and United Silicon (US2) tools layout and fabrication, was worked through for the first time in NASA's Flight Electronics Branch.

II. Project Organization

The SODR Memory Buffer Controller ASIC project team consisted of several different organizations working together. The technical individuals involved, their affiliations, and primary contributions are listed below.

Name	Affiliations	Primary Contributions
Robert F. Hodson	CNU	Principal Investigator ASIC System Design Master Controller Design Memory Interface Design Functional Simulations Acceptance Testing
Stephen Campbell	CNU (grad. student)	Group Controller Interface Design Functional Simulations DAS Testing
Stephen Jurczyk	NASA	Technical Project Management
Gerry Tucker	NASA	Technical Project Management
Glenn Hines	NASA	System Level Modelling Board Level Design
Edward Naddeo	NASA (Co-op)	HIPPI Device Modelling
Lawerence Abga	NASA JOVE Program	Independent Design Review
Cathy McGowan	SAIC	HIPPI Interface Design SCSI Interface Design Functional Simulations
Brian Krieder	US2	Foundry Technical Contact ASIC Layout

III. System Design

The SODR System has a modular, expandable design which allows for multiple data I/O ports and multiple optical disk drives under control of a centralized System Controller. Figure 1 is a block diagram of a fully configured SODR System.

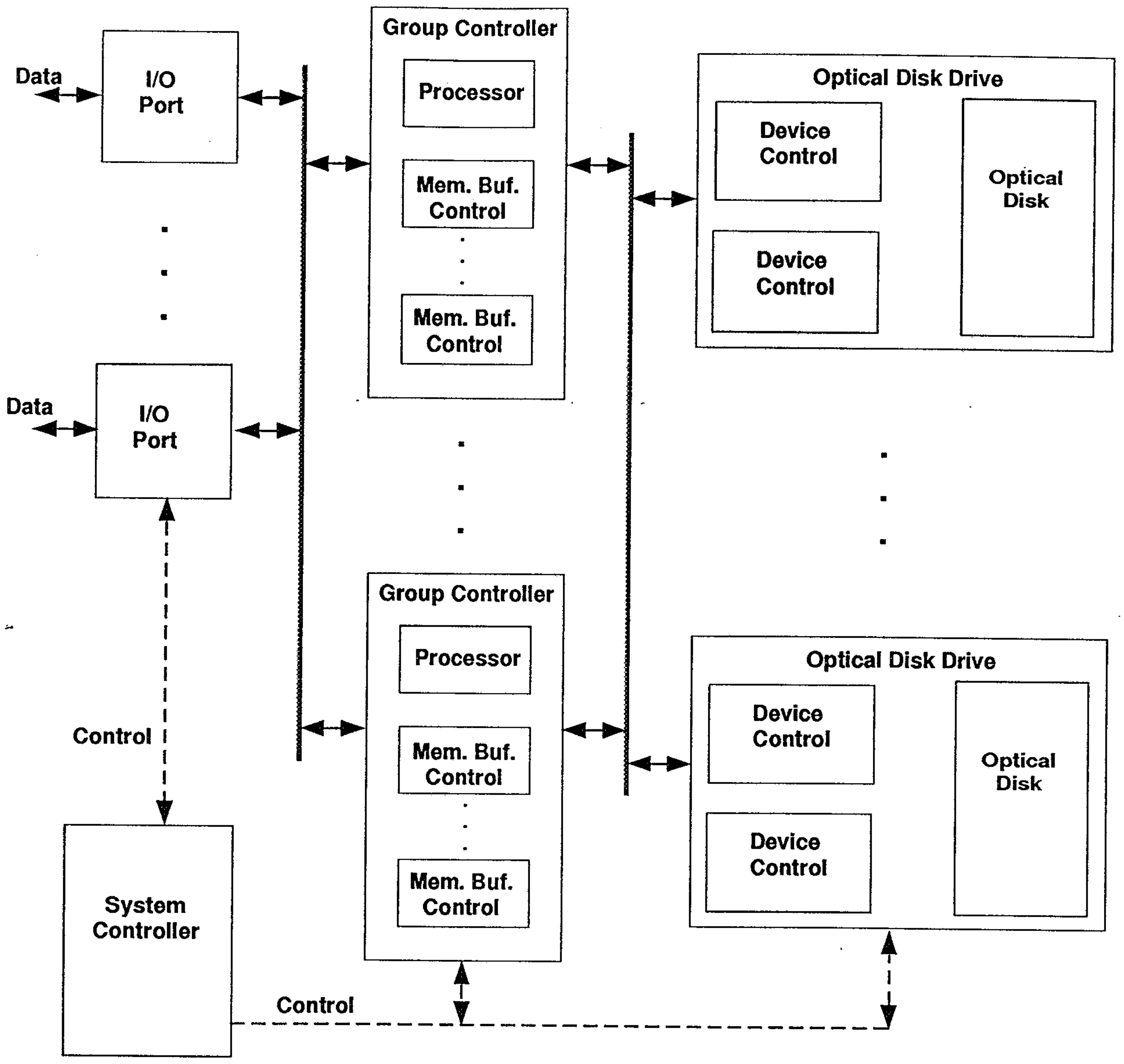


Figure 1. Fully Configured SODR System.

The basis of the SODR system is a high speed optical disk drive that supports 10 Gbytes capacity, 300 MBit/sec transfer rates, and 150 msec access times. Disks can be cascaded to achieve the appropriate storage requirement for a particular application. The system was designed with a standard disk interface (SCSI II) so that any SCSI II disks that meet capacity and data rate requirements could be used. This standardized disk interface allows for multiple disk vendors and opens the door for potential commercial applications.

The System Controller manages the overall system operation by responding to user commands and maintaining system status information. High level system commands like INITIALIZE SYSTEM, OPEN WRITE, OPEN READ and others are received by the System Controller. The System Controller allocates resources (Data Ports, Group Controllers and Disk Drives) to perform the details of the data transfer task.

The Group Controller performs the low level data transfer and buffering functions including the control of the Data Ports which use a HIPPI Protocol and the disk drive interface which uses a SCSI II Protocol. Each Group Controller is designed to interface with one I/O port and four disk drives. Data from the I/O Ports is striped across four disk drives to maintain the high I/O Port bandwidth of 600 MBits/sec.

Each group controller consists of a processor which receives commands from the system controller, a HIPPI I/O Port interface, a SCSI II Disk Interface, four Memory Buffer Control ASICs and 16 Mbytes of buffer memory.

IV. Memory Buffer Controller ASIC Design

The Group Controller consisting of an I/O Port, four Memory Buffer Control ASICs along with buffer memories (4MBytes each), four SCSI II disk drive interfaces and and the Group Controller processor is shown in Figure 2.

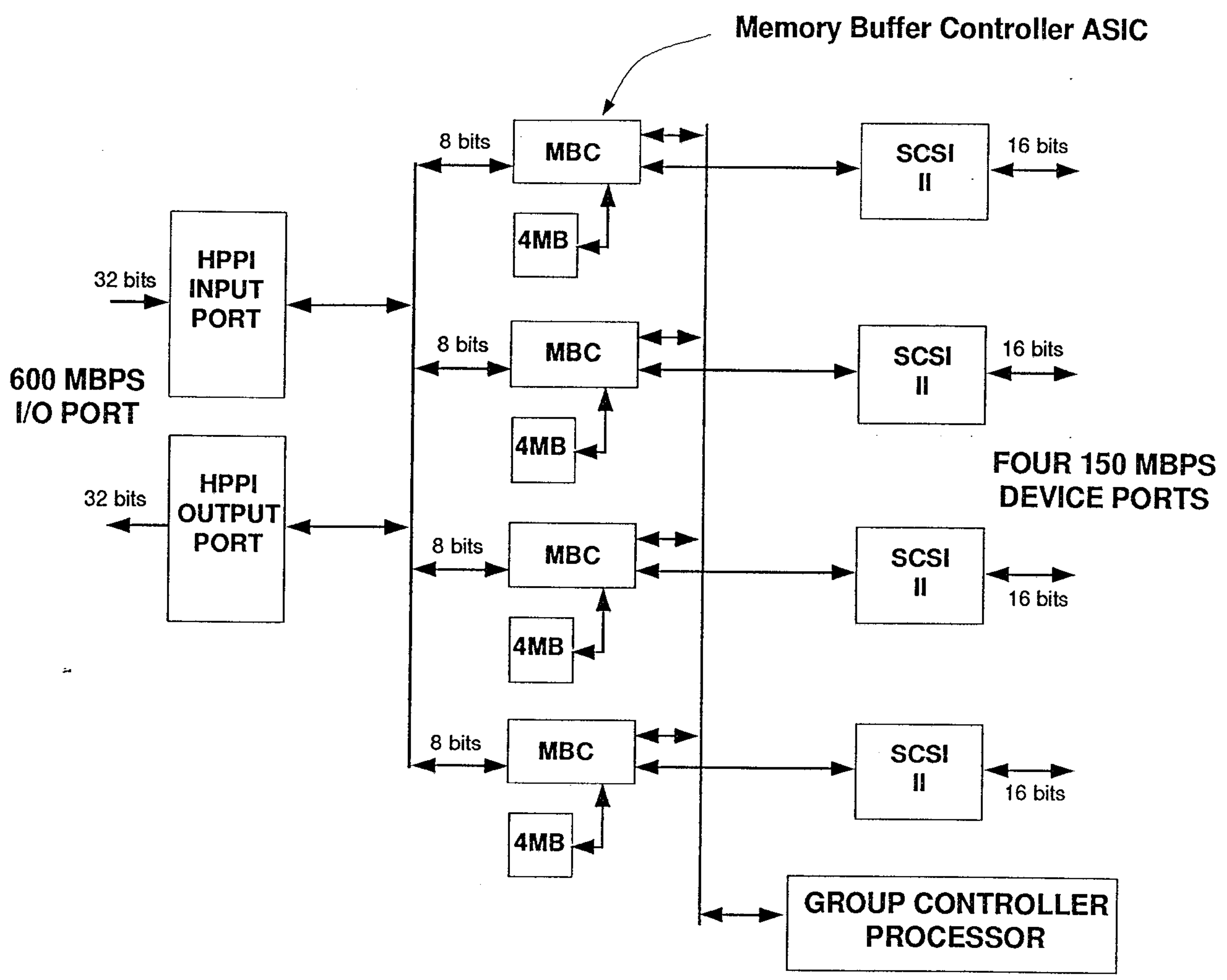


Figure 2. Group Controller

The MBC ASIC was designed to interface with a specific chip set consisting of:

1) AMCC's S2020/S2021 HIPPI Source/Destination Interface Circuits
2) EMULEX's FAS-366 SCSI Processor
3) Cypress IDT7MP4045 256K x 32 CMOS static RAM memories

The best/typical/worse case timing for these devices were used for functional testing of the MBC design. Pinout of the MBC was designed to interface with these devices with minimal glue logic.

Functionally the MBC ASIC performs data conversion and data buffering as shown in Figure 3. In HIPPI to SCSI Mode, 8-bits of a HIPPI data burst arrive from the AMCC HIPPI destination chip. The 8-bit data are buffered and de-multiplexed into 32-bit data before it is sent off-chip to a 4 MByte memory for buffering. Data are read back from memory and multiplexed into a 16-bit format before being sent off-chip to the EMULEX SCSI Processor. The data path is reversed in SCSI to HIPPI Mode. Data comes into the MBC ASIC from the SCSI Processor, it is de-multiplexed, sent to memory, multiplexed and sent out to the HIPPI source chip. The MBC ASIC can also function in a diagnostic mode where the Group Controller can read and write the 4 MByte memory through the ASIC. This mode is useful for testing memory and for checking out either the HIPPI or SCSI interfaces independently.

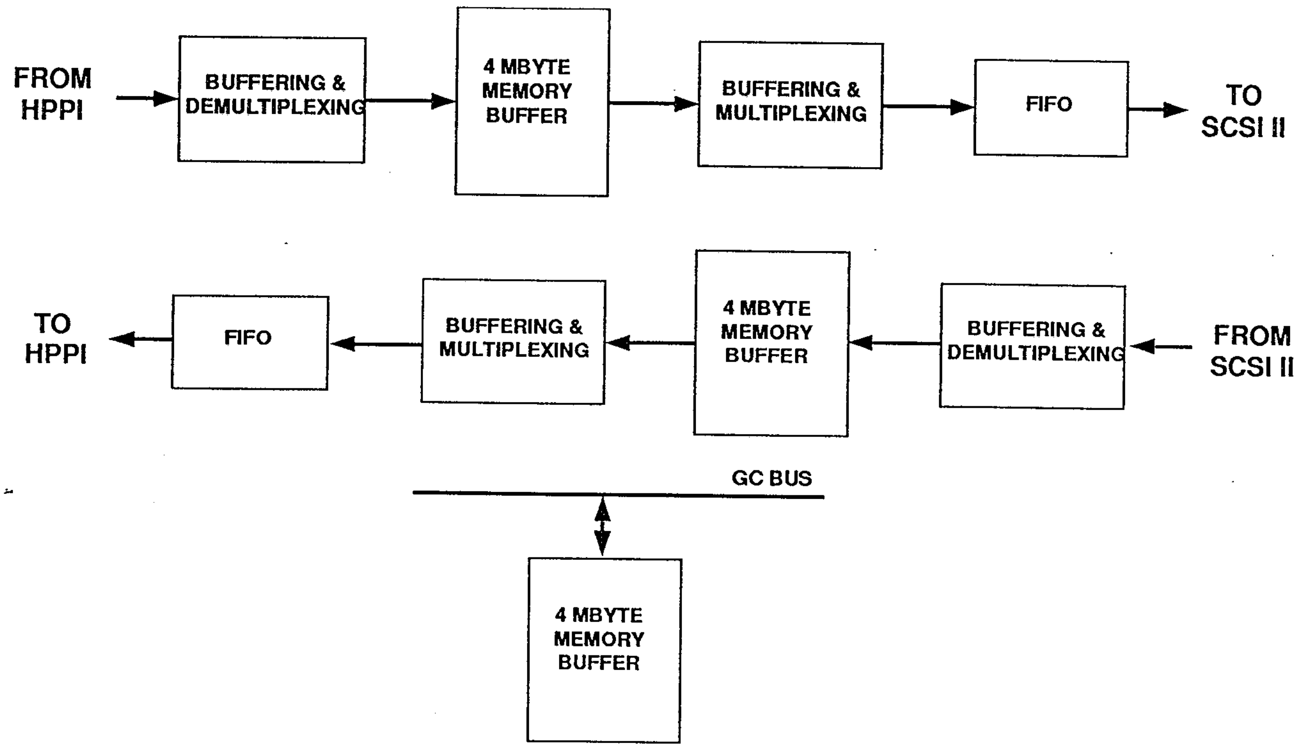


Figure 3. MBC ASIC Data Paths.

The MBC ASIC was designed as five major functional units as shown in Figure 4.

1) Memory Buffer Master Controller
2) HIPPI Interface.
3) SCSI II Interface.
4) Group Controller Interface.
5) Memory Interface.

Memory Buffer Master Controller (MBMC)

The Memory Buffer Master Controller (MBMC) is designed to provide centralized control over the various interfaces within the MBC ASIC. It is a large state machine the has the following operating modes:

Mode 0: Reset
Mode 1: HIPPI to SCSI
Mode 2: SCSI to HIPPI
Mode 3: GC to Memory
Mode 4: Memory to HIPPI
Mode 5: Memory to SCSI

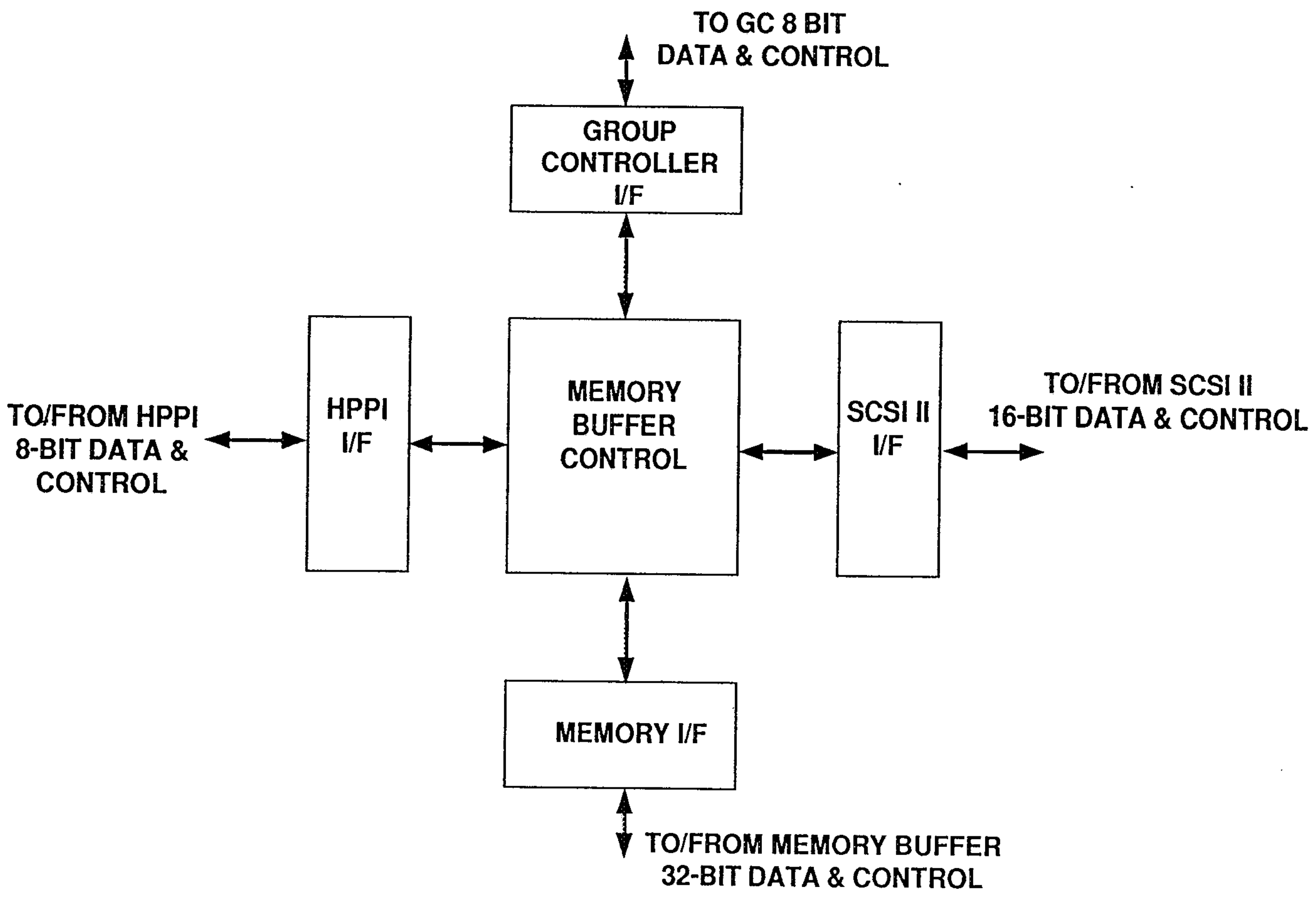

TO GC 8 BIT DATA & CONTROL
GROUP CONTROLLER I/F
TO/FROM HPPI 8-BIT DATA & CONTROL
HPPI I/F
MEMORY BUFFER CONTROL
SCSI II I/F
TO/FROM SCSI II 16-BIT DATA & CONTROL
MEMORY I/F
TO/FROM MEMORY BUFFER 32-BIT DATA & CONTROL

Figure 4.Functional View

For each mode the MBMC performs high level control over the appropriate data transfers. It provides information to the interfaces about the direction of transfer and controls the high level data transfer timing between the interfaces.

HIPPI Interface

The HIPPI interface communicates with the AMCC HIPPI source and destination chip set. These chips implement the High Performance Parallel Interface Standard. The chips perform the high level channel connect protocol as well as the low level data transfers. The HIPPI channel is 32-bits wide. This is 32-bit data path is striped across 4 MBC ASICs, each controlling the transfer of an 8-bit data path to independent SCSI II disk drives. The MBC ASIC only implements the low level control for the AMCC chips. It also performs the data transfer functions (burst transfers). The Connect/Disconnect protocol must be implemented with external logic with the exception of the I-field transfer. The timing between the Connect/Disconnect and the burst transfers is not critical and therefore was de-coupled to reduce pinout on the MBC ASIC.

The function of the HIPPI interface is to first issue an I-field during the HIPPI Connect phase and then handle the data transfer. This is accomplished by writing an I-field to the GC interface and placing the ASIC in Memory to HIPPI Mode. After a connect is achieved, the mode can be changed to SCSI to HIPPI and the HIPPI Interface will buffer data it receives from the Memory Interface into burst of 256 bytes. This is accomplished by using a 512x8 FIFO and monitoring its half full flag. As bursts become available, the data is sent to the AMCC source chip for transfer across the HIPPI channel. When the MBC ASIC is in HIPPI to SCSI Mode, the HIPPI interface transfers data in the opposite timing and control to the HIPPI source/destination chips are performed by the HIPPI Interface.

SCSI II Interface

The SCSI II interface communicates with the EMULEX FAS366 SCSI Processor. The MBC ASIC controls the handshaking for the DMA interface and provides a 16-bit bidirectional data path to the SCSI Processor. The microprocessor interface to the FAS366 is not controlled by the MBC ASIC and should be controlled by the Group Controller microprocessor. This interface conains a 256x16 FIFO for data rate matching between the Memory interface and the SCSI interface. Data is received from the Memory Interface in HIPPI to SCSI mode and sent to the Memory Interface in SCSI to HIPPI Mode. The interface contains several small state machines to control the data transfer timing.

Group Controller Interface

The Group Controller interface connects the MBC ASIC to the Group Controller microprocessor. Since the Group Controller microprocessor has not be chosen yet, a generic memory mapped interface design was implemented within the MBC ASIC. This interface has an 8-bit data path and five bits of register selects. Read, Write, and an ASIC Select signal are used to control the data transfers to this interface. Through the GC Interface the MBC ASIC is controlled by setting modes, checking status or transferring data. Details of the GC Interface's register mapping are given in ASIC Data Sheet Section of this report.

Memory Interface

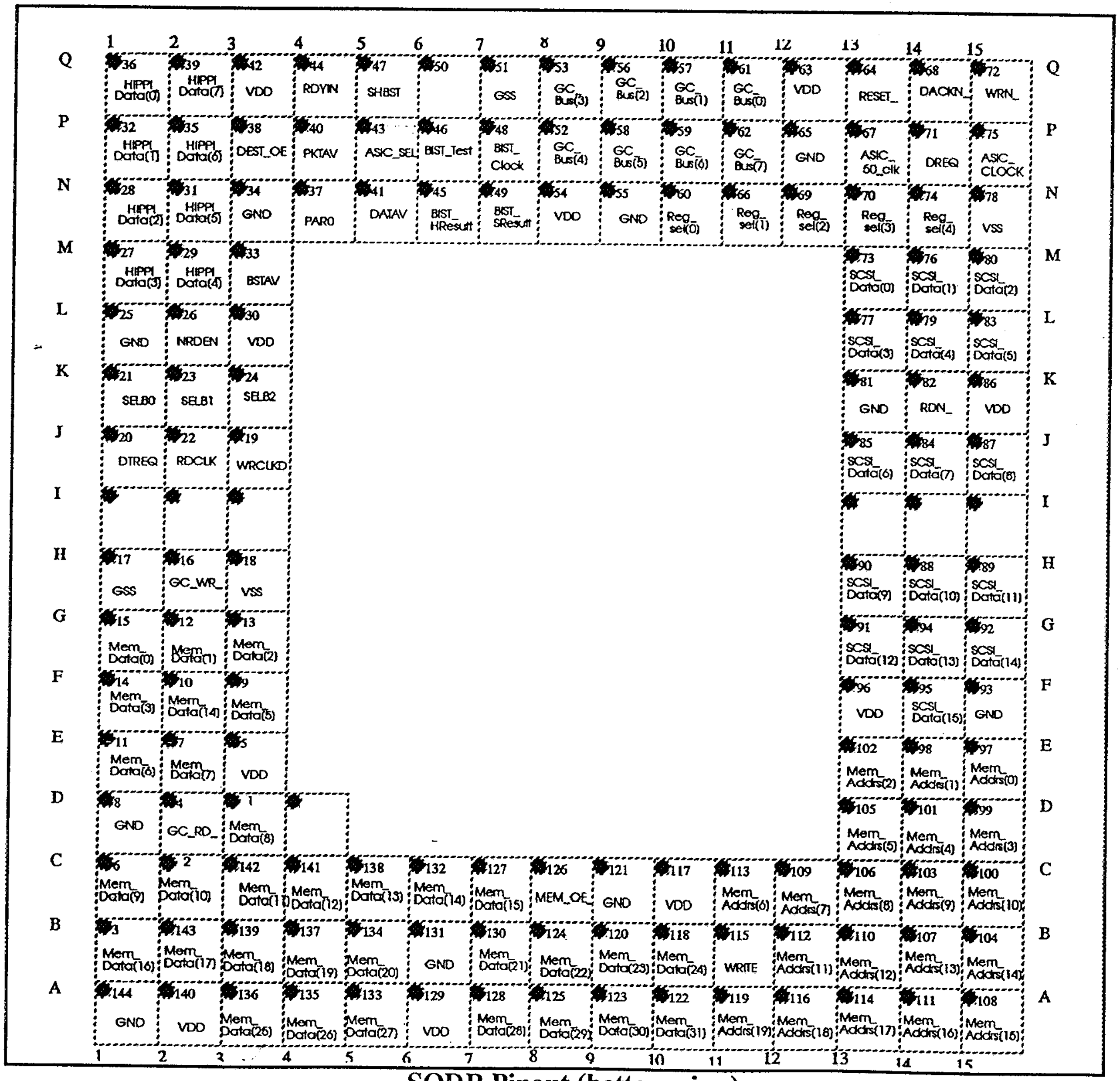
The Memory Interface controls the the low level timing of data transfers to or from the 4 MByte off-chip buffer memory. The memory is organized into a 1MByte by 32-bit memory. This requires 4 memory chips with an external decoder logic. The Memory Interface can control memory transfers to and from the HIPPI Interface, the SCSI Interface or the GC Interface depending on the operating mode of the MBC ASIC. The memory functions as a circular buffer using read and write address registers to point to head and tail of the buffer space. A memory full status is also maintained by this interface and can be read via the GC Interface.

V. SODR MBC Data Sheets

Description:	SODR Memory Buffer Controller ASIC
Package/Pins:	Ceramic PGA144
Material/Process:	CMOS 1.0 micron Technology
Ratings:	Industrial (-40 to 85C)
Power Consumption:	0.25W

Sheets included:

1) Pinout Diagram
2) Signal Description
3) ES2 Pinout Report
4) ES2 I/O Cell Specifications
5) ES2 I/O Cell Library Sheets
6) Register Map
7) SODR Operating Guide
8) US2 Power Calculation Worksheets



SODR Pinout (bottom view)

Signal Description

PIN NAME	: I/O	: #	: Type	: Description
Clock25	: I	: 1	: CMOS	: 25MHz Asic Clock
Reset_	: I	: 1	: TTL	: Chip Reset (active low)
BIST_Test	: I	: 1	: TTL	: BIST Test input
BIST_HResult	: O	: 1	: TTL	: BIST Test Result (HIPPI)
BIST_SResult	: O	: 1	: TTL	: BIST Test Result (SCSI)
BIST_Clock	: I	: 1	: TTL	: BIST TEST Clock
GC I/F:				
Asic_SEL_	: I	: 1	: TTL	: Asic Select (active low)
GC_Bus[7:0]	: I/O	: 8	: TTL	: GC Data Bus
GC_WR_	: I	: 1	: TTL	: GC Write (active low)
GC_RD_	: I	: 1	: TTL	: GC Read (active low)
Reg_sel[4:0]	: I	: 5	: TTL	: GC Register Select
HIPPI I/F:				
HPPI_Data[7:0]	: I/O	: 8	: TTL	: HIPPI Data Bus
DTREQ	: I	: 1	: TTL	: HIPPI Data Request
NRDEN	: I	: 1	: TTL	: HIPPI NOT Read Enabled
RDCLK	: I	: 1	: TTL	: HIPPI Read Clock
SELB0	: I	: 1	: TTL	: HIPPI SELB
SELB1	: I	: 1	: TTL	: HIPPI SELB
SELB2	: I	: 1	: TTL	: HIPPI SELB
WRCLKD	: I	: 1	: TTL	: HIPPI Write Clock
BSTAV	: O	: 1	: TTL	: HIPPI Burst Available
DEST_OE_	: O	: 1	: TTL	: HIPPI Destination Output Enable
PAR0	: O	: 1	: TTL	: HIPPI Parity 0
PKTAV	: O	: 1	: TTL	: HIPPI Packet Available
RDYIN	: O	: 1	: TTL	: HIPPI Ready In
SHBST	: O	: 1	: TTL	: HIPPI Short Burst
Sync	: I	: 1	: TTL	: HIPPI Synchronized True
MEM I/F:				
Mem_Data[31:0]	: I/O	: 32	: TTL	: Memory Data Bus
Mem_Addrs[19:0]	: O	: 20	: Ttl	: Memory Address
WRITE_	: O	: 1	: TTL	: Memroy Write (active low)
MEM_OE_	: O	: 1	: TTL	: Memory Output Enable (active low)
SCSI I/F:				
SCSI_Data[15:0]	: I/O	: 16	: TTL	: SCSI Data Bus
ASIC_50_clk	: I	: 1	: TTL	: 50MHz SCSI Clock
DREQ	: I	: 1	: TTL	: SCSI Data Request
DACKN_	: O	: 1	: TTL	: SCSI Data Acknowledge (active low)
RDN_	: O	: 1	: TTL	: SCSI Read (active low)
WRN_	: O	: 1	: TTL	: SCSI Write (acitve low)

```
*****************************************************************************
*   :***  :***  ***:  *    S I L I C O N     W O R K     O R D E R S       *
*   :*    ***:  :***  ******************************************************
*   ****  ****  ****  *     D E V I C E    P I N - O U T    S H E E T      *
*****************************************************************************
*     DATE:            *  PACKAGE TYPE:       *  PRODUCT No:               *
*     30/11/1993       *     BGA144S          *    10248                   *
*                      *                      *                            *
*---------------------------------------------------------------------------*
*BND:PAD:FT :PT :  PIN : ES2 DATA               :   PIN CUST      :PAD CENTRE *
*PIN:PIN:PIN:PIN: FUNC :  SHEET                 :     NAME        :COORDINATES*
*---------------------------------------------------------------------------*
*  1:   :   :   : I/O  : IOR1P                  : Mem_Data<8>     : 3279,-2827*
*  2:   :   :   : I/O  : IOR1P                  : Mem_Data<10>    : 3279,-2664*
*  3:   :   :   : I/O  : IOR1P                  : Mem_Data<16>    : 3279,-2461*
*  4:   :   :   : I    : IPS8G                  : GC_RD_          : 3279,-2329*
*  5:   :   :   : VDDX : PWRPY                  : pwr             : 3279,-2197*
*  6:   :   :   : I/O  : IOR1P                  : Mem_Data<9>     : 3279,-2065*
*  7:   :   :   : I/O  : IOR1P                  : Mem_Data<7>     : 3279,-1933*
*  8:   :   :   : GNDX : GNDPY                  : gnd             : 3279,-1801*
*  9:   :   :   : I/O  : IOR1P                  : Mem_Data<5>     : 3279,-1669*
* 10:   :   :   : I/O  : IOR1P                  : Mem_Data<4>     : 3279,-1537*
* 11:   :   :   : I/O  : IOR1P                  : Mem_Data<6>     : 3279,-1405*
* 12:   :   :   : I/O  : IOR1P                  : Mem_Data<1>     : 3279,-1273*
* 13:   :   :   : I/O  : IOR1P                  : Mem_Data<2>     : 3279,-1141*
* 14:   :   :   : I/O  : IOR1P                  : Mem_Data<3>     : 3279,-1009*
* 15:   :   :   : I/O  : IOR1P                  : Mem_Data<0>     : 3279,-877 *
* 16:   :   :   : I    : IPS8G                  : GC_WR_          : 3279,-745 *
* 17:   :   :   : GNDC : GNDCO                  : gnd             : 3279,-613 *
* 18:   :   :   : VDDC : PWRCO                  : pwr             : 3279,-481 *
* 19:   :   :   : I    : IPS8G                  : WRCLKD          : 3279,-349 *
* 20:   :   :   : I    : IPS8G                  : DTREQ           : 3279,442  *
* 21:   :   :   : I    : IPS8G                  : SELB0           : 3279,574  *
* 22:   :   :   : I    : IPS8G                  : RDCLK           : 3279,706  *
* 23:   :   :   : I    : IPS8G                  : SELB1           : 3279,838  *
* 24:   :   :   : I    : IPS8G                  : SELB2           : 3279,970  *
* 25:   :   :   : GNDX : GNDPY                  : gnd             : 3279,1102 *
* 26:   :   :   : I    : IPS8G                  : NRDEN           : 3279,1234 *
* 27:   :   :   : I/O  : IOS1P                  : HPPI_Data<3>    : 3279,1366 *
* 28:   :   :   : I/O  : IOS1P                  : HPPI_Data<2>    : 3279,1498 *
* 29:   :   :   : I/O  : IOS1P                  : HPPI_Data<4>    : 3279,1630 *
* 30:   :   :   : VDDX : PWRPY                  : pwr             : 3279,1762 *
* 31:   :   :   : I/O  : IOS1P                  : HPPI_Data<5>    : 3279,1894 *
* 32:   :   :   : I/O  : IOS1P                  : HPPI_Data<1>    : 3279,2033 *
* 33:   :   :   : O    : OPR1U                  : BSTAV           : 3279,2189 *
* 34:   :   :   : GNDX : GNDPY                  : gnd             : 3279,2361 *
* 35:   :   :   : I/O  : IOS1P                  : HPPI_Data<6>    : 3279,2547 *
* 36:   :   :   : I/O  : IOS1P                  : HPPI_Data<0>    : 3279,2832 *
* 37:   :   :   : O    : OPS4U                  : PAR0            : 2843,3242 *
* 38:   :   :   : O    : OPR1U                  : DEST_OE         : 2688,3242 *
* 39:   :   :   : I/O  : IOS1P                  : HPPI_Data<7>    : 2467,3242 *
*****************************************************************************
* PIN DISTRIBUTION:                  | PIN FUNCTION:     GND  : Digital gnd    *
*                                    | GNDC : Core gnd   GNDX : Periphery gnd  *
* GND  : 0    GNDC : 2    GNDX : 11  | VDD  : Dig. pwr   VDDX : Periphery pwr  *
* VDD  : 0    VDDC : 2    VDDX : 10  | VDDC : Core power VDDP : Power-on Rst   *
* VDDP : 0    VDCP : 0    AVDR : 0   | VDCP : Core POR   AI/O : Analog bidir   *
* AVDD : 0    AGND : 0    AGNR : 0   | AVDD : Analog pwr AGND : Analog gnd     *
* I    : 22   O    : 33   I/O  : 64  | AVDR : Ref. high  AGNR : Ref. low       *
* AIN  : 0    AOUT : 0    AI/O : 0   | AIN  : Analog in  AOUT : Analog output  *
* TRI  : 0    VDXP : 0    NC   : 0   | I    : Input      O    : Output         *
*                                    | I/O  : Bidir      TRI  : Tristate       *
*  ES2 DATA SHEET: FOR INTERNAL USE BY ES2.                                    *
*****************************************************************************
```

```
*************************************************************************
*   :***   :***   ***:  *    S I L I C O N     W O R K     O R D E R S   *
*   :*     ***:   :***  **************************************************
*   ****   ****   ****  *     D E V I C E     P I N - O U T    S H E E T  *
*************************************************************************
*      DATE:            *  PACKAGE TYPE:        *  PRODUCT No:            *
*      30/11/1993       *      BGA144S          *    10248                *
*                       *         -             *                         *
*-----------------------------------------------------------------------*
*BND:PAD:FT :PT :  PIN  : ES2 DATA              :   PIN CUST      :PAD CENTRE *
*PIN:PIN:PIN:PIN: FUNC  :  SHEET                :     NAME        :COORDINATES*
*-----------------------------------------------------------------------*
* 40:   :   :   : O     : OPR1U                 : PKTAV           : 2335,3242 *
* 41:   :   :   : GNDX  : GNDPYPOR              : gnd             : 2203,3242 *
* 42:   :   :   : VDDX  : PWRPY                 : pwr             : 2071,3242 *
* 43:   :   :   : I     : IPS8G                 : ASIC_SEL_       : 1939,3242 *
* 44:   :   :   : O     : OPR1U                 : RDYIN           : 1807,3242 *
* 45:   :   :   : O     : OPR1U                 : BIST_HResult    : 1675,3242 *
* 46:   :   :   : I     : IPS8G                 : BIST_Test       : 1543,3242 *
* 47:   :   :   : O     : OPR1U                 : SHBST           : 1411,3242 *
* 48:   :   :   : I     : IPS8G                 : BIST_Clock      : 1279,3242 *
* 49:   :   :   : O     : OPR1U                 : BIST_SResult    : 1147,3242 *
* 50:   :   :   : I     : IPS8G                 : sync            : 1015,3242 *
* 51:   :   :   : GNDC  : GNDCO                 : gnd             :  883,3242 *
* 52:   :   :   : I/O   : IOR1P                 : GC_Bus<4>       :  751,3242 *
* 53:   :   :   : I/O   : IOR1P                 : GC_Bus<3>       :  619,3242 *
* 54:   :   :   : VDDX  : PWRPY                 : pwr             : -172,3242 *
* 55:   :   :   : GNDX  : GNDPY                 : gnd             : -304,3242 *
* 56:   :   :   : I/O   : IOR1P                 : GC_Bus<2>       : -436,3242 *
* 57:   :   :   : I/O   : IOR1P                 : GC_Bus<1>       : -568,3242 *
* 58:   :   :   : I/O   : IOR1P                 : GC_Bus<5>       : -700,3242 *
* 59:   :   :   : I/O   : IOR1P                 : GC_Bus<6>       : -832,3242 *
* 60:   :   :   : I     : IPS8G                 : Reg_sel<0>      : -964,3242 *
* 61:   :   :   : I/O   : IOR1P                 : GC_Bus<0>       :-1096,3242 *
* 62:   :   :   : I/O   : IOR1P                 : GC_Bus<7>       :-1228,3242 *
* 63:   :   :   : VDDX  : PWRPY                 : pwr             :-1360,3242 *
* 64:   :   :   : I     : IPS8G                 : RESET_          :-1492,3242 *
* 65:   :   :   : GNDX  : GNDPY                 : gnd             :-1624,3242 *
* 66:   :   :   : I     : IPS8G                 : Reg_sel<1>      :-1756,3242 *
* 67:   :   :   : I     : IPS8G                 : ASIC_50_clk     :-1888,3242 *
* 68:   :   :   : O     : OPS1U                 : DACKN_          :-2020,3242 *
* 69:   :   :   : I     : IPS8G                 : Reg_sel<2>      :-2152,3242 *
* 70:   :   :   : I     : IPS8G                 : Reg_sel<3>      :-2315,3242 *
* 71:   :   :   : I     : IPS8G                 : DREQ            :-2687,3242 *
* 72:   :   :   : O     : OPS1U                 : WRN_            :-2862,3242 *
* 73:   :   :   : I/O   : IOR1P                 : SCSI_Data<0>    :-3279,2830 *
* 74:   :   :   : I     : IPS8G                 : Reg_sel<4>      :-3279,2546 *
* 75:   :   :   : I     : IPS8G                 : ASIC_CLOCK      :-3279,2361 *
* 76:   :   :   : I/O   : IOR1P                 : SCSI_Data<1>    :-3279,2189 *
* 77:   :   :   : I/O   : IOR1P                 : SCSI_Data<3>    :-3279,2032 *
* 78:   :   :   : VDDC  : PWRCO                 : pwr             :-3279,1893 *
*************************************************************************
* PIN DISTRIBUTION:                   | PIN FUNCTION:      GND  : Digital gnd     *
*                                     | GNDC : Core gnd    GNDX : Periphery gnd   *
* GND  : 0    GNDC : 2    GNDX : 11   | VDD  : Dig. pwr    VDDX : Periphery pwr   *
* VDD  : 0    VDDC : 2    VDDX : 10   | VDDC : Core power  VDDP : Power-on Rst    *
* VDDP : 0    VDCP : 0    AVDR : 0    | VDCP : Core POR    AI/O : Analog bidir    *
* AVDD : 0    AGND : 0    AGNR : 0    | AVDD : Analog pwr  AGND : Analog gnd      *
* I    : 22   O    : 33   I/O  : 64   | AVDR : Ref. high   AGNR : Ref. low        *
* AIN  : 0    AOUT : 0    AI/O : 0    | AIN  : Analog in   AOUT : Analog output   *
* TRI  : 0    VDXP : 0    NC   : 0    | I    : Input       O    : Output          *
*                                     | I/O  : Bidir       TRI  : Tristate        *
*  ES2 DATA SHEET: FOR INTERNAL USE BY ES2.                                       *
*************************************************************************
```

```
*************************************************************************
*   :***  :***  ***:  *   S I L I C O N    W O R K     O R D E R S     *
*   :*    ***:  :***  ***************************************************
*   ****  ****  ****  *    D E V I C E     P I N - O U T    S H E E T   *
*************************************************************************
*     DATE:               *  PACKAGE TYPE:      *  PRODUCT No:          *
*     30/11/1993          *     BGA144S         *    10248              *
*                         *                     *                       *
*-----------------------------------------------------------------------*
*BND:PAD:FT :PT :  PIN : ES2 DATA            :   PIN CUST      :PAD CENTRE *
*PIN:PIN:PIN:PIN: FUNC :  SHEET              :     NAME        :COORDINATES*
*-----------------------------------------------------------------------*
* 79:   :   :   : I/O   : IOR1P              : SCSI_Data<4>    :-3279,1761 *
* 80:   :   :   : I/O   : IOR1P              : SCSI_Data<2>    :-3279,1629 *
* 81:   :   :   : GNDX  : GNDPY              : gnd             :-3279,1497 *
* 82:   :   :   : O     : OPS1U              : RDN_            :-3279,1365 *
* 83:   :   :   : I/O   : IOR1P              : SCSI_Data<5>    :-3279,1233 *
* 84:   :   :   : I/O   : IOR1P              : SCSI_Data<7>    :-3279,1101 *
* 85:   :   :   : I/O   : IOR1P              : SCSI_Data<6>    :-3279,969  *
* 86:   :   :   : VDDX  : PWRPY              : pwr             :-3279,837  *
* 87:   :   :   : I/O   : IOR1P              : SCSI_Data<8>    :-3279,705  *
* 88:   :   :   : I/O   : IOR1P              : SCSI_Data<10>   :-3279,573  *
* 89:   :   :   : I/O   : IOR1P              : SCSI_Data<11>   :-3279,441  *
* 90:   :   :   : I/O   : IOR1P              : SCSI_Data<9>    :-3279,-350 *
* 91:   :   :   : I/O   : IOR1P              : SCSI_Data<12>   :-3279,-482 *
* 92:   :   :   : I/O   : IOR1P              : SCSI_Data<14>   :-3279,-614 *
* 93:   :   :   : GNDX  : GNDPY              : gnd             :-3279,-746 *
* 94:   :   :   : I/O   : IOR1P              : SCSI_Data<13>   :-3279,-878 *
* 95:   :   :   : I/O   : IOR1P              : SCSI_Data<15>   :-3279,-1010*
* 96:   :   :   : VDDX  : PWRPY              : pwr             :-3279,-1142*
* 97:   :   :   : O     : OPR1U              : Mem_Addrs<0>    :-3279,-1274*
* 98:   :   :   : O     : OPR1U              : Mem_Addrs<1>    :-3279,-1406*
* 99:   :   :   : O     : OPR1U              : Mem_Addrs<3>    :-3279,-1538*
*100:   :   :   : O     : OPR1U              : Mem_Addrs<10>   :-3279,-1670*
*101:   :   :   : O     : OPR1U              : Mem_Addrs<4>    :-3279,-1802*
*102:   :   :   : O     : OPR1U              : Mem_Addrs<2>    :-3279,-1934*
*103:   :   :   : O     : OPR1U              : Mem_Addrs<9>    :-3279,-2066*
*104:   :   :   : O     : OPR1U              : Mem_Addrs<14>   :-3279,-2198*
*105:   :   :   : O     : OPR1U              : Mem_Addrs<5>    :-3279,-2330*
*106:   :   :   : O     : OPR1U              : Mem_Addrs<8>    :-3279,-2462*
*107:   :   :   : O     : OPR1U              : Mem_Addrs<13>   :-3279,-2653*
*108:   :   :   : O     : OPR1U              : Mem_Addrs<15>   :-3279,-2828*
*109:   :   :   : O     : OPR1U              : Mem_Addrs<7>    :-2865,-3242*
*110:   :   :   : O     : OPR1U              : Mem_Addrs<12>   :-2675,-3242*
*111:   :   :   : O     : OPR1U              : Mem_Addrs<16>   :-2372,-3242*
*112:   :   :   : O     : OPR1U              : Mem_Addrs<11>   :-2151,-3242*
*113:   :   :   : O     : OPR1U              : Mem_Addrs<6>    :-2019,-3242*
*114:   :   :   : O     : OPR1U              : Mem_Addrs<17>   :-1887,-3242*
*115:   :   :   : O     : OPR1U              : WRITE_          :-1755,-3242*
*116:   :   :   : O     : OPR1U              : Mem_Addrs<18>   :-1623,-3242*
*117:   :   :   : VDDX  : PWRPY              : pwr             :-1491,-3242*
*************************************************************************
* PIN DISTRIBUTION:                  | PIN FUNCTION:     GND  : Digital gnd     *
*                                    | GNDC : Core gnd    GNDX : Periphery gnd   *
* GND  : 0   GNDC : 2   GNDX : 11   | VDD  : Dig. pwr    VDDX : Periphery pwr   *
* VDD  : 0   VDDC : 2   VDDX : 10   | VDDC : Core power  VDDP : Power-on Rst    *
* VDDP : 0   VDCP : 0   AVDR : 0    | VDCP : Core POR    AI/O : Analog bidir    *
* AVDD : 0   AGND : 0   AGNR : 0    | AVDD : Analog pwr  AGND : Analog gnd      *
* I    : 22  O    : 33  I/O  : 64   | AVDR : Ref. high   AGNR : Ref. low        *
* AIN  : 0   AOUT : 0   AI/O : 0    | AIN  : Analog in   AOUT : Analog output   *
* TRI  : 0   VDXP : 0   NC   : 0    | I    : Input       O    : Output          *
*                                    | I/O  : Bidir       TRI  : Tristate        *
*  ES2 DATA SHEET: FOR INTERNAL USE BY ES2.                                    *
*************************************************************************
```

```
***************************************************************************
*   :***  :***  ***:  *   S I L I C O N    W O R K     O R D E R S       *
*   :*    ***:  :***  *****************************************************
*   ****  ****  ****  *    D E V I C E    P I N - O U T    S H E E T      *
***************************************************************************
*     DATE:               *  PACKAGE TYPE:       *  PRODUCT No:            *
*     30/11/1993          *     BGA144S          *    10248                *
*                         *                      *                         *
*-------------------------------------------------------------------------*
*BND:PAD:FT :PT :  PIN  : ES2 DATA           :   PIN CUST        :PAD CENTRE *
*PIN:PIN:PIN:PIN: FUNC  :  SHEET             :     NAME          :COORDINATES*
*-------------------------------------------------------------------------*
*118:   :   :   : I/O   :  IOR1P             : Mem_Data<24>      :-1359,-3242*
*119:   :   :   : O     :  OPR1U             : Mem_Addrs<19>     :-1227,-3242*
*120:   :   :   : I/O   :  IOR1P             : Mem_Data<23>      :-1095,-3242*
*121:   :   :   : GNDX  :  GNDPY             : gnd               : -963,-3242*
*122:   :   :   : I/O   :  IOR1P             : Mem_Data<31>      : -831,-3242*
*123:   :   :   : I/O   :  IOR1P             : Mem_Data<30>      : -699,-3242*
*124:   :   :   : I/O   :  IOR1P             : Mem_Data<22>      : -567,-3242*
*125:   :   :   : I/O   :  IOR1P             : Mem_Data<29>      : -435,-3242*
*126:   :   :   : O     :  OPR1U             : MEM_OE_           : -303,-3242*
*127:   :   :   : I/O   :  IOR1P             : Mem_Data<15>      : -171,-3242*
*128:   :   :   : I/O   :  IOR1P             : Mem_Data<28>      :  620,-3242*
*129:   :   :   : VDDX  :  PWRPY             : pwr               :  752,-3242*
*130:   :   :   : I/O   :  IOR1P             : Mem_Data<21>      :  884,-3242*
*131:   :   :   : GNDX  :  GNDPY             : gnd               : 1016,-3242*
*132:   :   :   : I/O   :  IOR1P             : Mem_Data<14>      : 1148,-3242*
*133:   :   :   : I/O   :  IOR1P             : Mem_Data<27>      : 1280,-3242*
*134:   :   :   : I/O   :  IOR1P             : Mem_Data<20>      : 1412,-3242*
*135:   :   :   : I/O   :  IOR1P             : Mem_Data<26>      : 1544,-3242*
*136:   :   :   : I/O   :  IOR1P             : Mem_Data<25>      : 1676,-3242*
*137:   :   :   : I/O   :  IOR1P             : Mem_Data<19>      : 1808,-3242*
*138:   :   :   : I/O   :  IOR1P             : Mem_Data<13>      : 1940,-3242*
*139:   :   :   : I/O   :  IOR1P             : Mem_Data<18>      : 2072,-3242*
*140:   :   :   : VDDX  :  PWRPY             : pwr               : 2204,-3242*
*141:   :   :   : I/O   :  IOR1P             : Mem_Data<12>      : 2336,-3242*
*142:   :   :   : I/O   :  IOR1P             : Mem_Data<11>      : 2468,-3242*
*143:   :   :   : I/O   :  IOR1P             : Mem_Data<17>      : 2707,-3242*
*144:   :   :   : GNDX  :  GNDPY             : gnd               : 2863,-3242*
***************************************************************************
* PIN DISTRIBUTION:                | PIN FUNCTION:     GND  : Digital gnd    *
*                                  | GNDC : Core gnd   GNDX : Periphery gnd  *
* GND  : 0   GNDC : 2   GNDX : 11  | VDD  : Dig. pwr   VDDX : Periphery pwr  *
* VDD  : 0   VDDC : 2   VDDX : 10  | VDDC : Core power VDDP : Power-on Rst   *
* VDDP : 0   VDCP : 0   AVDR : 0   | VDCP : Core POR   AI/O : Analog bidir   *
* AVDD : 0   AGND : 0   AGNR : 0   | AVDD : Analog pwr AGND : Analog gnd     *
* I    : 22  O    : 33  I/O  : 64  | AVDR : Ref. high  AGNR : Ref. low       *
* AIN  : 0   AOUT : 0   AI/O : 0   | AIN  : Analog in  AOUT : Analog output  *
* TRI  : 0   VDXP : 0   NC   : 0   | I    : Input      O    : Output         *
*                                  | I/O  : Bidir      TRI  : Tristate       *
*  ES2 DATA SHEET: FOR INTERNAL USE BY ES2.                                  *
***************************************************************************
```

1.2 Family Specifications

These specifications apply to the following processes: ECPD15, ECPD12 and ECPD10.

1.2.1 Electrical Characteristics

1.2.1.1 Recommended Operating Conditions

This is the range for which ES2's library cells have been characterised. Operation of a device outside this range may result in the device failing to meet some of its specifications.

SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT	CONDITIONS
VDD	DC supply voltage	2.5	5.0	5.5	V	
VI	DC input voltage	0		VDD	V	
VO	DC output voltage	0		VDD	V	
TA	Operating free air temp range	- 55 - 40		+125 + 85	C C	Military Industrial
TR	Input rise time			300	ns	10%-90% CMOS and TTL
TR	Input rise time			No Limit		triggers CMOS and TTL
TF	Input fall time			300	ns	90%-10% CMOS and TTL
TF	Input fall time			No Limit		triggers CMOS and TTL

See the characterisation information in Section 1.2.2 for definitions of the conditions used in this table.

1.2.1.2 Absolute Maximum Ratings

Operation of a device outside this range may cause permanent damage to the device and/or affect reliability.

SYMBOL	PARAMETER	MIN	MAX	UNIT	CONDITIONS
VDD	DC supply voltage	-0.5	7.0	V	
VI	DC input voltage	-0.5	VDD+0.5	V	Or see +-IIk
VO	DC output voltage	-0.5	VDD+0.5	V	Or see +-IOk
+-IIk	DC input diode current		10	mA	VI < -0.5V VI > VDD+0.5V
+-IOk	DC output diode current		20	mA	VO < -0.5V VO > VDD+0.5V Output tristate
IOLMAX	Continuous * output current		IOLX2.0 IOLX1.5	mA mA	Industrial Military
IOHMAX	Continuous * output current		IOHX2.0 IOHX1.5	mA mA	Industrial Military
TSG	Storage temperature	- 65	+150	C	
TSH	Time of outputs shorted		5	sec	
TA	Operating free air temp range	- 55 - 40	+125 + 85	C C	Military Industrial

* Applies to external output or bi-directional pads. See datasheets for IOL and IOH.

See the characterisation information in Section 1.2.2 for definitions of the conditions used in this table.

1.2.1.3 Input/Output Characteristics (Padlib2 Library Only)

(Industrial temperature range, VDD = 4.5 to 5.5V, TA = –40 to +85 C)

SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT	CONDITIONS
VOH	High level output voltage	VDD-0.05			V	IOH = 0 mA
VOL	Low level output voltage			0.05	V	IOL = 0 mA
VOH	High level output voltage	VDD-0.5			V	IOH = rated output current
VOL	Low level output voltage			0.5	V	IOL = rated output current
IOZ	High impedance output leakage current (bidir cells)			2.0 1.0	µA µA	VDD = 5.5V VDD = 3.0V
VIH	High level input voltage	70%VDD			V	CMOS
VIL	Low level input voltage			30%VDD	V	CMOS
VIH	High level input voltage	2.00V			V	TTL
VIL	Low level input voltage			0.80	V	TTL
IIH	Input leakage w/o pull-up			1.00 0.50	µA µA	VIN=VDD=5.5V VIN=VDD=3.0V
IIL	Input leakage w/o pull-up			1.00 0.50	µA µA	VIN=0 VDD=5.5V VIN=0 VDD=3.0V

See the characterisation information in Section 1.2.2 for definitions of the conditions used in this table.

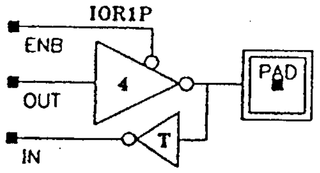
ES2 PadLib2

process ECPD1Ø
REV.1.1

FEB 91

Bidirectional I/O cell
4mA tri-state output buffer with half di/dt
TTL inverting input buffer

IOR1P



PARAMETER	VALUE	UNIT
Size	13Ø.5*399.8	um2
C_PAD	4	pF
Cin_ENB	Ø.Ø86	pF
Cin_OUT	Ø.Ø85	pF
Fanout_IN	Ø.91	pF
Fanout_PAD	1ØØ.Ø	pF
total cap	6.73	pF
transistors	3Ø	

PARAMETER	CONDITIONS	MILITARY MIN	MILITARY MAX	INDUSTRIAL MIN	INDUSTRIAL MAX	UNIT
IIH	VI=VCC, temp=full range	−1Ø	1Ø	−1Ø	1Ø	uA
IIL	VI=Ø, temp=full range	−1Ø	1Ø	−1Ø	1Ø	uA
VIL	VCC worst case VCC=4.5V	–	Ø.8	–	Ø.8	V
VIH	VCC worst case VCC=5.5V	2.Ø	–	2.Ø	–	V
IOZ	VOZ= VCC or ØV	−1Ø	1Ø	−1Ø	1Ø	uA
VOL	IOL=+5.5mA & VCC=4.5V	–	–	–	Ø.5	V
VOL	IOL=+4.ØmA & VCC=4.5V	–	Ø.5	–	–	V
VOH	IOH=−5.5mA & VCC=4.5V	–	–	4.Ø	–	V
VOH	IOH=−4.ØmA & VCC=4.5V	4.Ø	–	–	–	V

PARAMETER	FROM	TO	MIN	TYP	MAX	MIL	UNIT
tplh	OUT	PAD	1.4Ø	3.5Ø	7.71	9.29	ns
tphl	OUT	PAD	1.11	2.79	6.13	7.38	ns
tphz	ENB	PAD	1.58	3.96	8.71	1Ø.5	ns
tplz	ENB	PAD	1.3Ø	3.24	7.13	8.59	ns
tpzl	ENB	PAD	1.Ø8	2.7Ø	5.94	7.15	ns
tplz	ENB	PAD	1.32	3.3Ø	7.26	8.74	ns
tplh	PAD	IN	Ø.Ø73	Ø.18	Ø.4Ø	Ø.48	ns
tphl	PAD	IN	Ø.Ø42	Ø.1Ø	Ø.23	Ø.28	ns
Δtplh	OUT	PAD	Ø.Ø15	Ø.Ø36	Ø.Ø8Ø	Ø.Ø96	ns/pF
Δtphl	OUT	PAD	Ø.Ø16	Ø.Ø4Ø	Ø.Ø89	Ø.11	ns/pF
Δtplh	PAD	IN	Ø.41	1.Ø4	2.28	2.75	ns/pF
Δtphl	PAD	IN	Ø.11	Ø.27	Ø.59	Ø.71	ns/pF

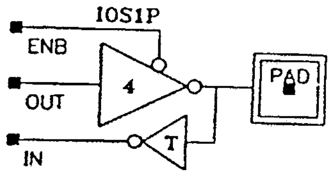
ES2 PadLib2

process ECPD1Ø.
REV.1.1

FEB 91

Bidirectional I/O cell
4mA tri-state output buffer
TTL inverting input buffer

IOS1P



PARAMETER	VALUE	UNIT
Size	13Ø.5*399.8	um2
C_PAD	4	pF
Cin_ENB	Ø.Ø86	pF
Cin_OUT	Ø.Ø85	pF
Fanout_IN	Ø.91	pF
Fanout_PAD	1ØØ.Ø	pF
total cap	6.76	pF
transistors	3Ø	

PARAMETER	CONDITIONS	MILITARY MIN	MILITARY MAX	INDUSTRIAL MIN	INDUSTRIAL MAX	UNIT
IIH	VI=VCC, temp=full range	−1Ø	1Ø	−1Ø	1Ø	uA
IIL	VI=Ø, temp=full range	−1Ø	1Ø	−1Ø	1Ø	uA
VIL	VCC worst case VCC=4.5V	–	Ø.8	–	Ø.8	V
VIH	VCC worst case VCC=5.5V	2.Ø	–	2.Ø	–	V
IOZ	VOZ= VCC or ØV	−1Ø	1Ø	−1Ø	1Ø	uA
VOL	IOL=+5.5mA & VCC=4.5V	–	–	–	Ø.5	V
VOL	IOL=+4.ØmA & VCC=4.5V	–	Ø.5	–	–	V
VOH	IOH=−5.5mA & VCC=4.5V	–	–	4.Ø	–	V
VOH	IOH=−4.ØmA & VCC=4.5V	4.Ø	–	–	–	V

PARAMETER	FROM	TO	MIN	TYP	MAX	MIL	UNIT
tplh	OUT	PAD	Ø.63	1.57	3.45	4.16	ns
tphl	OUT	PAD	Ø.61	1.54	3.38	4.Ø7	ns
tphz	ENB	PAD	Ø.65	1.63	3.59	4.32	ns
tplz	ENB	PAD	Ø.58	1.45	3.19	3.84	ns
tpzl	ENB	PAD	Ø.48	1.21	2.66	3.2Ø	ns
tplz	ENB	PAD	Ø.54	1.36	2.99	3.6Ø	ns
tplh	PAD	IN	Ø.Ø73	Ø.18	Ø.4Ø	Ø.48	ns
tphl	PAD	IN	Ø.Ø42	Ø.1Ø	Ø.23	Ø.28	ns
Δtplh	OUT	PAD	Ø.Ø12	Ø.Ø31	Ø.Ø68	Ø.Ø82	ns/pF
Δtphl	OUT	PAD	Ø.Ø13	Ø.Ø33	Ø.Ø73	Ø.Ø88	ns/pF
Δtplh	PAD	IN	Ø.41	1.Ø4	2.28	2.75	ns/pF
Δtphl	PAD	IN	Ø.11	Ø.27	Ø.59	Ø.71	ns/pF

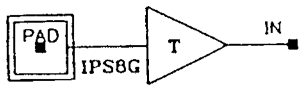
ES2 PadLib2

process ECPD10
REV.1.1

FEB 91

TTL input buffer

IPS8G



PARAMETER	VALUE	UNIT
Size	130.5*399.8	um2
C_PAD	3	pF
Fanout_IN	3.24	pF
total cap	2.59	pF
transistors	13	

PARAMETER	CONDITIONS	MILITARY MIN	MILITARY MAX	INDUSTRIAL MIN	INDUSTRIAL MAX	UNIT
IIH	VI=VCC, temp=full range	−10	10	−10	10	uA
IIL	VI=0, temp=full range	−10	10	−10	10	uA
VIL	VCC worst case VCC=4.5V	–	0.8	–	0.8	V
VIH	VCC worst case VCC=5.5V	2.0	–	2.0	–	V

PARAMETER	FROM	TO	MIN	TYP	MAX	MIL	UNIT
tplh	PAD	IN	0.10	0.25	0.56	0.67	ns
tphl	PAD	IN	0.23	0.56	1.24	1.49	ns
Δtplh	PAD	IN	0.13	0.33	0.73	0.88	ns/pF
Δtphl	PAD	IN	0.10	0.26	0.57	0.69	ns/pF

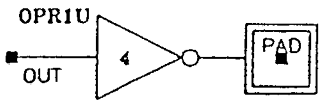
ES2 PadLib2

process ECPD10
REV.1.1

FEB 91

4mA output buffer with half di/dt

OPR1U



PARAMETER	VALUE	UNIT
Size	130.5*399.8	um2
C_PAD	4	pF
Cin_OUT	0.085	pF
Fanout_PAD	100.0	pF
total cap	5.44	pF
transistors	14	

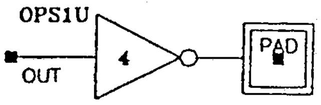
PARAMETER	CONDITIONS	MILITARY MIN	MILITARY MAX	INDUSTRIAL MIN	INDUSTRIAL MAX	UNIT
VOL	IOL=+5.5mA & VCC=4.5V	–	–	–	0.5	V
VOL	IOL=+4.0mA & VCC=4.5V	–	0.5	–	–	V
VOH	IOH=–5.5mA & VCC=4.5V	–	–	4.0	–	V
VOH	IOH=–4.0mA & VCC=4.5V	4.0	–	–	–	V

PARAMETER	FROM	TO	MIN	TYP	MAX	MIL	UNIT
tplh	OUT	PAD	1.46	3.65	8.03	9.67	ns
tphl	OUT	PAD	1.13	2.83	6.23	7.50	ns
Δtplh	OUT	PAD	0.015	0.037	0.081	0.098	ns/pF
Δtphl	OUT	PAD	0.017	0.041	0.091	0.11	ns/pF

FEB 91

4mA output buffer

OPS1U



PARAMETER	VALUE	UNIT
Size	130.5*399.8	um2
C_PAD	4	pF
Cin_OUT	0.085	pF
Fanout_PAD	100.0	pF
total cap	5.18	pF
transistors	12	

PARAMETER	CONDITIONS	MILITARY MIN	MILITARY MAX	INDUSTRIAL MIN	INDUSTRIAL MAX	UNIT
VOL	IOL=+5.5mA & VCC=4.5V	–	–	–	0.5	V
VOL	IOL=+4.0mA & VCC=4.5V	–	0.5	–	–	V
VOH	IOH=–5.5mA & VCC=4.5V	–	–	4.0	–	V
VOH	IOH=–4.0mA & VCC=4.5V	4.0	–	–	–	V

PARAMETER	FROM	TO	MIN	TYP	MAX	MIL	UNIT
tplh	OUT	PAD	0.46	1.14	2.51	3.02	ns
tphl	OUT	PAD	0.49	1.23	2.71	3.26	ns
Δtplh	OUT	PAD	0.013	0.032	0.070	0.084	ns/pF
Δtphl	OUT	PAD	0.014	0.034	0.075	0.090	ns/pF

Register Map

Description: This file lists the names and addresses for the registers used in the SODR Memory Buffer Controller ASIC.

Dec. Addrs	Hex Addrs	Register	Type	Comments
0	00	None	N/A	NOT USED
1	01	HPPI STATUS	Read	Tri-state outputs
2	02	HPPI CONTROL	Write	Register outputs
3	03	HPPI Interrupt	Read	Tri-state outputs
4	04	IFIELD	Write	Register outputs
5	05	Transfer Counter [LSB]	Read/Write	
6	06	Transfer Counter	Read/Write	
7	07	Transfer Counter	Read/Write	
8	08	Transfer Counter	Read/Write	
9	09	Transfer Counter	Read/Write	
10	0A	Transfer Counter [MSB]	Read/Write	
11	0B	System Status Read	Read	Tri-state/ASIC clock
12	0C	System Mode Write	Write	Register outputs
13	0D	Address to G.C. [LSB]	Write	Register outputs
14	0E	Address to G.C.	Write	Register outputs
15	0F	Address to G.C. [MSB]	Write	Register outputs
16	10	Mem to GC Data [LSB]	Read	Tri-state/ASIC clock
17	11	Mem to GC Data	Read	Tri-state/ASIC clock
18	12	Mem to GC Data	Read	Tri-state/ASIC clock
19	13	Mem to GC Data [MSB]	Read	Tri-state/ASIC clock
20	14	GC to Mem Data [LSB]	Write	Register outputs
21	15	GC to Mem Data	Write	Register outputs
22	16	GC to Mem Data	Write	Register outputs
23	17	GC to Mem Data [MSB]	Write	Register outputs
24	18	GC Address load	Signal	F/F ASIC clock
25	19	GC Address read	Signal	F/F ASIC clock
26	1A	GC Address write	Signal	F/F ASIC clock
27	1B	None	N/A	NOT USED
28	1C	None	N/A	NOT USED
29	1D	None	N/A	NOT USED
30	1E	None	N/A	NOT USED
31	1F	None	N/A	NOT USED

Operational Description

The following section describes the steps required to setup the various operating modes of the MBC ASIC.

HIPPI to SCSI Transfer:

1) Write Reset to the Mode Register.
2) Write a starting address to the memory address register (if desired). This consists of writing registers 13, 14, 15 with data and writing 24 to load.
3) Establish a HIPPI connection by externally controlling the destination chip (refer to HIPPI data sheet).
4) Write HIPPI_to_SCSI to the Mode Register. This causes the transfer to begin.
5) Disconnect HIPPI channel (if desired).

SCSI to HIPPI Transfer:

1) Write Reset to the Mode Register.
2) Write a starting address to the memory address register (if desired). This consists of writing registers 13, 14, 15 with data and writing 24 to load.
3) Write transfer counter with (2^{48} - number of bytes) to be transferred.
4) Write the I-Field to register 4.
5) Write Memory_to_HIPPI to the Mode Register.
6) Establish a HIPPI connection by externally controlling the source chip.
7) Write SCSI_to_HIPPI to Mode Register. This causes the transfer to begin.
8) Disconnect HIPPI channel (if desired).

GC to SCSI Transfer:

1) Write Reset to the Mode Register.
2) Write GC_to_Memory to the Mode Register.
3) Write Address to registers 13, 14, 15 and load with a write to register 24.
4) Write Data to registers 20, 21, 22, 23 and load with a write to register 26. (At this point data will be written to memory at the specified address.)
5) Repeat step 3 and 4 for each word written.
6) Write Memory_to_SCSI to the Mode Resister. This will start the data transfer from memory to the SCSI Processor. Note: the data will be inverted.

GC to HIPPI Transfer:

1) Write Reset to the Mode Register.
2) Write GC_to_Memory to the Mode Register.

3) Write Address to registers 13, 14, 15 and load with a write to register 24.
4) Write Data to registers 20, 21, 22, 23 and load with a write to register 26. (At this point data will be written to memory at the specified address.)
5) Repeat step 3 and 4 for each word written.
6) Write transfer counter with (2^{48} - number of bytes) to be transferred.
7) Write the I-Field to register 4.
8) Write Memory_to_HIPPI to Mode Register.
9) Establish a HIPPI Connection by externally controlling the source chip. Data will be sent from the MBC ASIC to the HIPPI Source chip. Note the data will be the inversion of what was written to memory.
10) Disconnect HIPPI channel (if desired).

HIPPI to GC Transfer:

1) Write Reset to the Mode Register.
2) Write GC_to_Memory to the Mode Register.
3) Write starting address to the memory address register. This consists of writing registers 13, 14, 15 with data and writing 24 to load.
4) Establish a HIPPI Connection by externally controlling the destination chip.
5) Write HIPPI_to_SCSI to the Mode Register. This causes the transfer to begin. Data will be transferred from the HIPPI interface to memory.
6) To read the data out of memory via the GC, write GC_to_Memory to the Mode Register.
7) Write the starting address by writing registers 13, 14, and 15 with data and writing register 24 to load.
8) Load the data into the GC interface data by writing register 24.
9) Read the data from registers 16, 17, 18, 19. Note: this data will be the inversion of what is sent from the HIPPI destination chip.
10) Repeat steps 7, 8 and 9 to read each word of memory.
11) Disconnect HIPPI channel (if desired).

SCSI to GC Transfer:

1) Write Reset to the Mode Register.
2) Write GC_to_Memory to the Mode Register
3) Write starting address to the memory address register. This consists of writing registers 13, 14, 15 with data and writing 24 to load.
4) Write SCSI_to_HIPPI to the Mode Register. The MBC ASIC is now ready to accept data from the SCSI Processor. The data will be transferred from the SCSI interface to the memory.
5) To read the data out of memory via the GC, write GC_to_Memory to the Mode Register.

6) Write the starting address by writing registers 13, 14, and 15 with data and writing register 24 to load.
7) Load the data into the GC interface data by writing register 24.
8) Read the data from registers 16, 17, 18, 19. Note: this data will be the inversion of what is sent from the SCSI Processor.
9) Repeat steps 6, 7 and 8 to read each word of memory.

GC to Memory Transfer:

1) Write Reset to the Mode Register.
2) Write GC_to_Memory to the Mode Register.
3) Write an address to registers 13, 14, 15 and load with a write to register 24.
4) Write data to registers 20, 21, 22, 23 and load with write to register 26. (At this point data will be written to memory at the specified address.)
5) Repeat step 3 and 4 for each word written.
6) To read data back, write the address again to registers 13, 14, 15 and load with a write to register 24.
7) Perform a memory read by writing register 25.
8) The data can then be read back via registers 16, 17, 18, 19.
9) Repeat steps 6, 7 and 8 to read each word.

Reading MBC ASIC Status:

1) Read register 11.

bit 0: Transfer Counter = 0.
bit 1: HIPPI FIFO Empty Flag.
bit 2: Memory Full.
bit 3. Memory Empty.

U S 2 Customer Design Signoff Document

Static Power Calculation Sheet

Spec No: AG8-HN17 Revision A

Customer Design Name: SOPR I/F

US2 Product Code: ______ Technology Used: ECPD10

US2 Databook used: ES2 ECPD10 Version: E01A09

Supply voltage: VDD = 5.0 V Temperature range: -40 to 85 C INDUS

Number of gates in the design:		12K	
Total leakage current for gates: 12 pA/1K	IGA =	14.4	uA
Number of input cells in the design:		22	@200nA
Total leakage current for input cells:	IINP =	4.4	uA
Number of output cells in the design:		33	
Total leakage current for output cells:	IOUT =	6.6	uA
Number of bi-directional I/O cells in the design:		64	
Total leakage current for bi-dir I/O cells:	IBDR =	12.8	uA
Number of compiled megacells:		2	
Megacell type: 256 x 16	Leakage current:	0.8	uA
Megacell type: 512	Leakage current:	0.9	uA
Megacell type:	Leakage current:		uA
Megacell type:	Leakage current:		uA
Total leakage current for megacells:	IMG =	1.7	uA
Number of analogue cells:			
Analogue cell:	Leakage current:		uA
Analogue cell:	Leakage current:		uA
Analogue cell:	Leakage current:		uA
Analogue cell:	Leakage current:		uA
Total leakage current for analogue cells:	IAN =		uA
Total leakage current for device: ITOT = IGA + IINP + IOUT + IBDR + IMG + IAN	ITOT =	39.9	uA
Average static output current per output cell: (6K)	ISOUT =	0.2	uA
Static power dissipation per output cell: (6K) PSOUT = ISOUT * ISOUT * VOL / IOL * 0.001	PSOUT =	3.64	mW
Total static power dissipation for output cells:	PTOUT =	120.12	mW
Total static power dissipation: PSTA = VDD * ITOT * 0.001 + PTOUT	PSTA =	120.32	mW

Note that minimum leakage is only achieved if:
* All internal tri-states nodes are biased.
* All PLAs are in standby mode and all oscillators are turned off.
* All inputs with pullup are at VDD, all other inputs are at 0 or VDD.

Report generated by: GLENN HINES Signature: [illegible] 8-18-93 Date

US2 approval by: ______ Signature: ______ Date

U S 2 Customer Design Signoff Document

Dynamic Power Calculation Sheet

Spec No: AG8-HN17 Revision A

Customer Design Name: SODR I/F

US2 Product Code: ______________ Technology Used: ECPD10

P = 9 uW/gate/MHz for ECPD15 P = 5 uW/gate/MHz for ECPD10 (all at VDD = 5V)
P = 7 uW/gate/MHz for ECPD12 P = 4 uW/gate/MHz for ECPD07

Supply voltage: VDD = 5.0 V Operating frequency: F = 25 MHz

Number of gates in the design:	G = 12 K	
Estimated fraction of gates switching simultaneously (typical 0.20)	S = 0.20	
Dynamic power dissipation by gates: PGA = P * F * S * G * 0.001	PGA = 300	mW
Total dynamic power dissipation for compiled megacells (consult generated datasheets): ±20%	PMG = 576	mW
Total dynamic power dissipation in core: PCOR = PGA + PMG	PCOR = 876	mW
Average output capacitance load:	CL = 15	pF
Estimated number of output pads (including bi-dir I/Os) switching simultaneously:	SP = 40	(42%)
Operating frequency for outputs:	FP = 25	MHz
Total output dynamic power dissipation: POUT = VDD * VDD * CL * SP * FP / 2000	POUT = 187.5	mW
Total input dynamic power dissipation: (5k) (typical 0 if input cells are included in gate count)	PINP = 4	mW
Total dynamic power dissipation for analogue cells: (consult analogue datasheets)	PAN = –	mW
Total dynamic power dissipation: PDYN = (PCOR + POUT + PINP + PAN) * 0.001	PDYN = 0.12	W
Total static power dissipation: (from US2 Static Power Calculation Sheet)	PSTA = 120	mW
Total device power dissipation: PTOT = PDYN + (PSTA * 0.001)	PTOT = 0.240	W
Ambient operating temperature: (5k)	TA = 85	C
Package type: PGA144S Theta JA of package:	TP = 35	C/W
Junction temperature: TJ = (PTOT * TP) + T	TJ = 93.54	C

Report generated by: ______________ Signature: ______________
Date

US2 approval by: ______________ Signature: ______________
Date

VI. Functional Testing

Extensive simulations were performed to verify correct logical operation of the MBC ASIC. These tests were performed for minimum, typical and maximum propagation delays. All tests were performed before and after layout. The post layout tests included extracted capacitance values used to estimate wire delays for more accurate simulations. All simulations used the US2 standard cell library and the Verilog-XL simulator.

Test Limitations

All simulations are limited by the accuracy of the timing models for the standard cells and for the wire delay estimations. We have a relatively high level of confidence in these models since they have been extensively used by US2. We did, however, find some problems with the timing of one of the I/O cells and the flag timing for the FIFO Models. The I/O cell timing was corrected by US2 and we worked around the FIFO problem by designing our own FIFO flag logic.

The HIPPI source and destinations chips were modelled in Verilog XL by a NASA co-op. These models are relatively complex. These models are used to provide stimulus to the MBC ASIC model for testing of the HIPPI interface. The timing of HIPPI signals is critical for correct operation. Although these signals were carefully examined by several individuals there is some risk involved in this interface since this is the first time these HIPPI models have been used,and their timing was determined by interpreting the HIPPI Data Sheet.

Similar to the HIPPI interface, the SCSI interface timing was gleamed from data sheets and modelled in Verilog-XL. This process always involves risk but in this case the risk is minimal due to the relatively simple timing involved in the SCSI interface.

Another limitation of functional simulations is that the number of test patterns written must be limited keep execution time reasonable. For example, in testing the SCSI to HIPPI data path only a couple of data bursts were written. Certain test like reading and writing the entire memory were prohibitive due to extremely long execution times.

Simulation Organization.

The logical organization of the simulation is shown in Figure 5. Verilog models exist for the HIPPI Channels, The MBC ASIC, and the Memory. The Test Driver provides the stimulus to this circuit and monitors signal responses to verify correct operations. Detailed signal timing was also verified with the Cwaves program which graphically displays selected signals. A copy of the Test Driver is given Appendix B. This test driver shows the detailed timing of all the controls signals required for correct circuit operation. The Test Driver is organized into eight main tasks which are each described in the following sections.

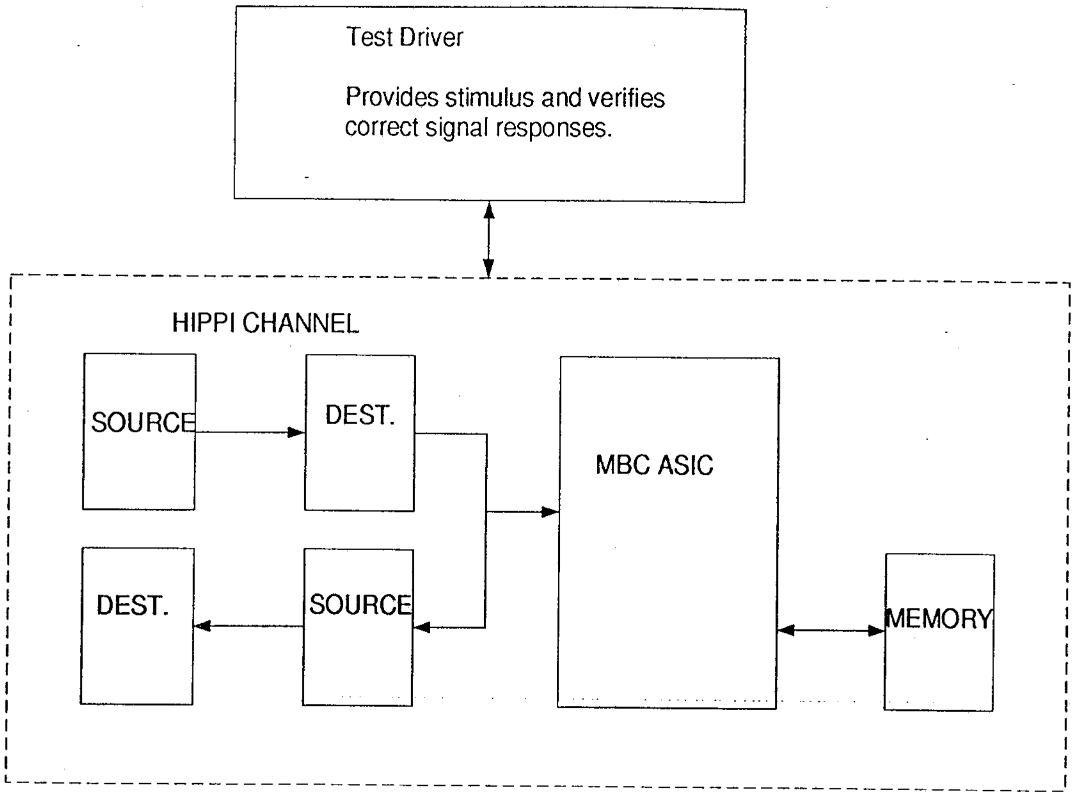


Figure 5. Functional test organization.

Test 1: GC to Memory Read/Write Test

The GC to Memory Test is designed to verify correct operation of the data path from the GC to the memory interface as well as the memory chip timing. This test writes three different data patterns to memory and then reads them back and verifies the patterns read back are correct. The verilog code which controls this test is fairly straight forward. All code is organized into tasks in a top down approach. The following code is an exerpt from the test driver.

```
begin
$display("-------------------------------------------\n");
$display("Starting GC to Memory Read/Write Test - test 1\n");
$display("-------------------------------------------\n");

 write_GC(mode_reg, RESET_mode);
 write_GC(mode_reg, GCtoM_mode);

 write_data(pattern1,t1_start_addr,t1_end_addr);
 check_mem_data(pattern1,t1_start_addr,t1_end_addr, noninvert);
 $display(" First pattern test complete.\n");
```

```
  write_data(pattern2,t1_start_addr,t1_end_addr);
  check_mem_data(pattern2,t1_start_addr,t1_end_addr, noninvert);
  $display(" Second pattern test complete.\n");

  write_data(pattern3,t1_start_addr,t1_end_addr);
  check_mem_data(pattern3,t1_start_addr,t1_end_addr, noninvert);
  $display(" Third pattern test complete.\n");

 $display("----------------------------------------\n");
 $display("GC to Memory Read/Write Test Complete.\n");
 $display("----------------------------------------\n");

 end
```

Chip timing constraints can also be verified within the test driver or within verilog models. The following code segment shows how timing for the memory chips was verified by testing the timing requirements given in the memory data sheet. If a setup, width, or recovery time violation occurs and error message is reported by the test.

```
specify
        specparam

        Tcw = 40,    //chip select to end of write
        Taw = 40,    //address valid to end of write
        Twp = 35,    //write pulse width
        Twr = 2,     //write recovery time
        Tdw = 25,    //data to end of write
        Tasu = 5,    //derived timing req., address to start of write,
                     //insures address doesn't change during write
        Taa = 45,    //address access time for read
        Tacs = 40;   //cs access time

        $setup(cs1_, posedge oe_, Tacs);
        $setup(cs2_, posedge oe_, Tacs);
        $setup(cs3_, posedge oe_, Tacs);
        $setup(cs4_, posedge oe_, Tacs);

        $width(negedge cs1_,  Tcw);
        $width(negedge cs2_,  Tcw);
        $width(negedge cs3_,  Tcw);
        $width(negedge cs4_,  Tcw);
        $setup(addr, posedge write_, Taw);

        $width(negedge write_, Twp);

        $recovery(posedge write_, addr, Twr);

        $setup(data, posedge write_, Tdw);

        $setup(addr, negedge write_, Tasu);
endspecify
```

Test 2: HIPPI to SCSI Test

The HIPPI to SCSI test verifies correct operation of the data path from the HIPPI to SCSI interfaces, using the memory as a buffer. This test sends a single packet of data with a single burst (256 bytes). First the MBC ASIC is reset and the address registers to the memory are initialized. The high level control to connect the HIPPI channel is performed. The connect function is performed independent of the MBC ASIC and simply establishes a HIPPI channel from the HIPPI Source to the HIPPI Destination. After a connection is established, an incrementing pattern is written to the HIPPI Source from the Test Driver. This pattern is transferred through the HIPPI channel and stimulates the HIPPI interface of the MBC ASIC. Data goes through the MBC ASIC to memory and is read back from memory and sent to the SCSI interface. The Test Driver performs the necessary handshake to acquire the data from the SCSI interface. The data is then verified as correct. Finally, a HIPPI disconnect is performed closing the HIPPI channel and the test terminates.

Test 3: SCSI to HIPPI Test

The SCSI to HIPPI test verifies correct operation of the data path from the SCSI interface, through the HIPPI interface using the memory and a buffer. This test first resets the MBC ASIC and then initializes the memory address registers. A connection is then established on the HIPPI channel. The connection requires writing the transfer counter and the I-Field to the MBC ASIC and performing the high level HIPPI control. The transfer counter is setup to transfer two bursts of data (512 bytes). An incrementing pattern is then written to the SCSI interface from the Test Driver. This data goes through the MBC ASIC and then through the HIPPI Channel. Data is acquired by the Test Driver at the HIPPI Destination chip and verified as correct. A HIPPI disconnect is then performed and the test is terminated.

Test 4: GC to SCSI Test

The GC to SCSI test writes a data pattern to memory while in GC_to_Memory mode and then transfers that data to the SCSI interface by changing to Memory_to_SCSI mode. This test simulates how the MBC ASIC might be used to check out the SCSI interface with no HIPPI device connected. The test acquires the data from the SCSI interface and verifies that it is correct. Several data patterns are written in this test to verify correct operation independent of data.

After multiple patterns have been written, read and verified, the test writes enough data to fill the SCSI FIFO and then tests that the FIFO full flag is set properly. This condition can occur if the SCSI drive is blocking or not handshaking properly with the MBC ASIC.

Test 5: SCSI to GC Test

The SCSI to GC test writes several data patterns into the SCSI interface of the MBC ASIC. These patterns are then written into memory and then are read back out via the GC interface. The data read out is verified as correct and the test terminates. This test simulates a setup that could check out the SCSI interface in the absence of a HIPPI drive.

Test 6: GC to HIPPI Test

The GC to HIPPI test verifies correct operation of the MBC ASIC from the GC interface to the HIPPI Interface. Data is also transferred through the simulated HIPPI channel so all the high level connect signals are also simulated. This test first writes a data pattern to memory while in GC_to_Memory mode. A connection on the HIPPI channel is established. This connection includes writing the transfer counter and the I-Field to the MBC ASIC and switching to Memory_to_HIPPI mode. A request_verify task acquires data at the HIPPI destination after data travels through the MBC ASIC and the HIPPI channel. Data is verified as correct and the test terminates.

Test 7: HIPPI to GC Test

In this test data are sent to the HIPPI source chip from the Test Driver. Data travels across the HIPPI Channel to the destination chip. Data then goes into the HIPPI interface of the MBC and into memory. Data is read out of memory via the GC interface. Memory address registers are first initialized and a HIPPI connection is established on the channel. An incrementing data pattern is written to the HIPPI Source chip while in HIPPI_to_SCSI mode. The data burst is transferred to memory by the MBC ASIC. The mode is then changed to GC_to_Memory mode and the data as addressed and read from memory. The data is then verified as correct and the test terminates.

Test 8: Memory Status Test

This test was design to write memory until it was full and read back the memory full flag to verify that the flag logic worked properly. Unfortunately writing 4MBytes of data to memory in the simulation would take days. Instead the Patch Tool was used to force a value in the memory word counter and the flag was verified. The memory word counter was also shown to increment and decrement correctly by graphically viewing its operations with CWaves.

Synchronization Test

Most of the functionality of the MBC ASIC can be tested by simulating a single device. One facet of the chip cannot, however, be tested stand alone. When data is travelling from SCSI to HIPPI, there is no guarantee the data arriving from multiple SCSI Drives will arrive in sync. In fact, it is highly unlikely that it will. Therefore, the MBC ASICs must sync up data from multiple SCSI Drives before sending it through the HIPPI channel. Remember that four MBC ASICs are used for one 32-bit HIPPI channel. In this configuration, one MBC ASIC is a master and controls the HIPPI signals while the other simply provides data. Figure 6 shows the organization of the Synchronization Test.

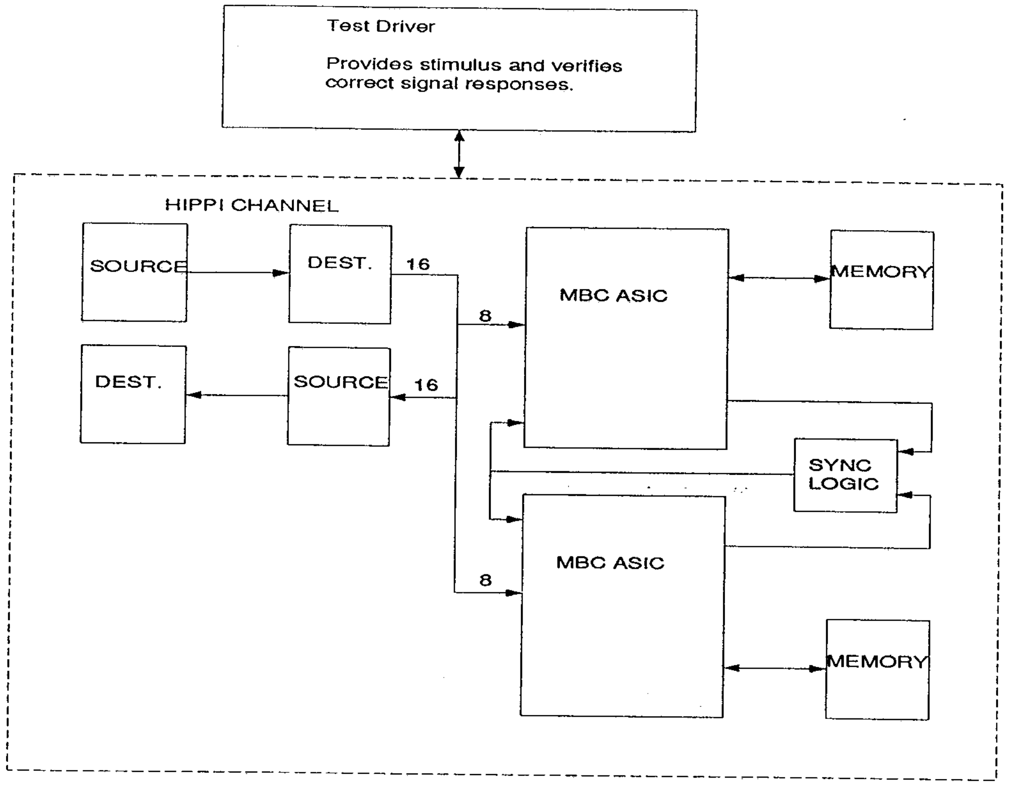


Figure 6. Synchronization Test Organization.

To verify that the synchronization logic was working properly, a separate test was devised with two MBC ASICs working in a master/slave mode and the external synchronization logic (a four input AND gate and a flip-flop). This test establishes a connection on HIPPI channel and sends data to both SCSI interfaces out of sync. The MBC ASICs must then synchronize the data and send it through the HIPPI channel. The data is verified as correct and then the test terminates.

Additional Miscellaneous Testing

In addtion to the testing stated, several other checks are performed to insure correct operation and timing of the MBC ASIC. Some of the addtional test are as follows:

1) HIPPI Timing checks (from data sheets).
2) SCSI Timing checks (from data sheets).
3) Memory Timing check (from data sheets).
4) HIPPI Errors test (monitoring of HIPPI source/destination errors).

VII. Acceptance Testing

Acceptance tests are post fabrication tests that are used to ensure that parts are fabricated without defects. These tests were designed by the NASA/CNU design team, but verified as correct by US2 at the foundry. All samples returned to NASA after fabrication passed acceptance testing.

Test Limitations

Ideally, acceptance testing guarantees a fault free design by providing test vectors which identify all possible stuck-at faults in a circuit. Unfortunately, since no fault analysis tools were available, another approach was taken. The acceptance tests were design to exercise all major data paths through the ASIC and verify correct data at the outputs.

Other limitations of the acceptance tests were:

1) The total number of test vectors was limited to 65,535.
2) The data rate was limited to 1MHz.
3) The data timing and sampling was limited as shown in Figure 7 below.

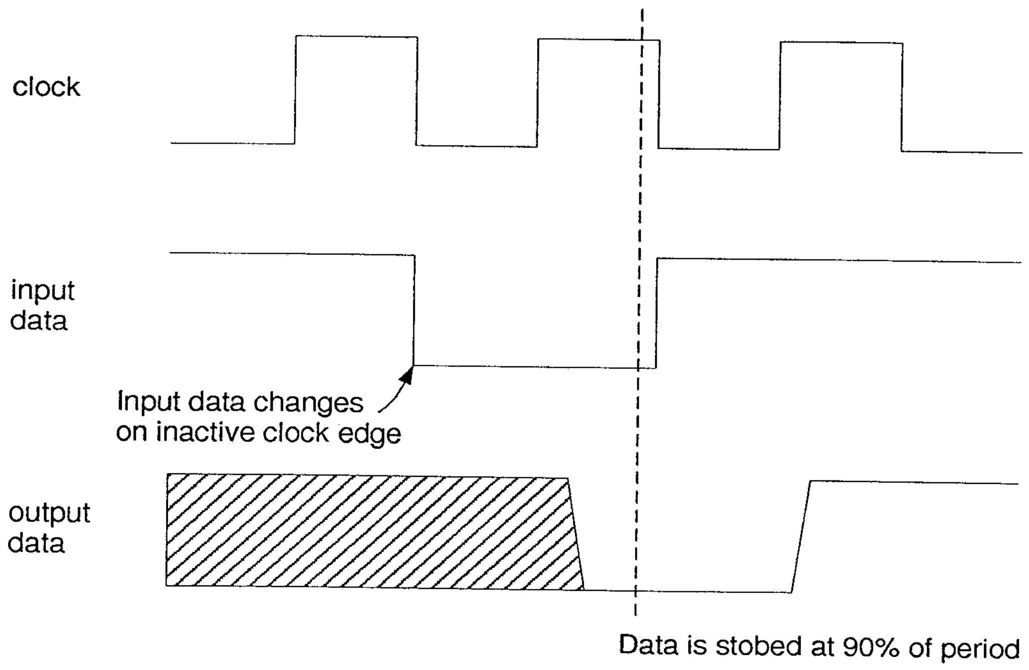

clock
input data
Input data changes on inactive clock edge
output data
Data is stobed at 90% of period

Figure 7. Test vector timing.

Tests

The Acceptance tests were written in Verilog -XL. Two tasks were provided by US2 to extract the test vectors and to verify that the extracted vectors would meet US2's timing requirements. These task are called $get_design and $check_design and are called from the test fixture. The result of running the Verilog Acceptance test is a file of test vectors with expected responses. A total of seven data path tests were performed along with BIST.

The Memory to SCSI test writes five different patterns to the memory while in GC_to_Memory mode and then transfer the data to the SCSI interface by switching to Memory_to_SCSI mode. The following data patterns (hex) are written: 33221100, FFFFFFFF, 00000000, AAAAAAAA, 55555555.

The GC to Memory test writes.four different test patterns to memory. These patterns are verified as correct at the memory interface. The patterns the test writes are: 55555555, AAAAAAAA, 00000000, 11111111.

The SCSI to Memory Test writes four different patterns to into the SCSI interface and these patterns are verified at the memory interface as correct. The patterns written were: 33221100, 00000000, 55555555, AAAAAAAA.

The Memory to GC test sets the data inputs to the memory interface to five different patterns. These patterns are read via the GC interface and verified as correct. The patterns used in this test were: 99887766, 00000000, FFFFFFFF, AAAAAAAA, 55555555.

The HIPPI to Memory test writes four different patterns to the HIPPI interface and verifies that the data are correct at the memory interface. The patterns written are: FF, 00, AA, 55.

The Transfer Counter test writes five different data patterns to the transfer counter and then reads the data back. The patterns written were: FF00AABBCC55, FFFFFFFFFFFF, 000000000000, AAAAAAAAAAAA, 555555555555.

The Memory to HIPPI test sets the data inputs on the memory interface to five different test patterns. The system is put into Memory_to_HIPPI mode and test patterns are read from the HIPPI interface and verified as correct. The patterns write were: FFFFFFFF, 00000000, AAAAAAAA, 55555555, FF00AA55.

In addition to the data path checking performed, all FIFOs were designed with Built in Self Test (BIST). The BIST test was exercised at the foundry and all chips provided by US2 passed BIST for both the HIPPI and SCSI FIFOs.

VIII. DAS Testing

In order to test the new ASICs, NASA/Langley purchased a Tektronix DAS9200. The DAS9200 is a Digital Acquisition System which can provide input stimulus and output acquisition capabilities to a complex digital system or chip such as the SODR ASIC. The DAS can be configured in a variety of ways depending on the acquisition and pattern generation cards installed. The DAS at NASA/Langley is equipped with:

1 92A96 96-channel by 8K, 100MHz acquisition card

1 92A16 16-channel by 4K, 200MHz acquisition card

1 92S16 16-channel by 1K, 50MHz pattern generator card

1 92S32T 32-channel by 8K, 50MHz pattern generator card.

These cards can be grouped together to behave as one piece of equipment operating synchronously (known as a cluster), or each card may be started or stopped individually. This SODR setup uses the 92A96, and the 92S32T cards as a single cluster.

Pattern generator cards supply inputs to the Device Under Test (DUT), in this case a single SODR MBC ASIC along with a 4MB buffer memory. The 92S16 card is an algorithmic card which can be programmed to follow a user-defined set of instructions. The 92S32T, on the other hand, supplies vectors (patterns of logical 1's and 0's) to the DUT. These vectors are generated external to the DAS, configured for the DAS, and then imported to the DAS. These vectors are available in two formats: Swave and Ewave. The Swave format provides a vector for every input at a regular interval such as every edge of the 50MHz clock used by the SODR ASIC. The Ewave format provides a new vector only when a single input changes or when multiple inputs change.

The DAS9200 can be used stand alone with a 92020XT X-terminal as a front end, but it's uses are limited. The primary limitation is that the Programmatic Command Language (PCL) tools are not accessible. This means input stimulus and machine configuration files (Swave and device map files) must be entered by hand. Other control and configuration capabilities include connection to a RS-232 host, a LAN-connected host, or a GPIB-connected host. Preliminary investigation into setting up the DAS indicated a need for a network connection for the DAS and 92LANP software (not originally purchased) in order to communicate and connect to the DAS using the CAEDE Lab's Sun Workstations. After a conference between Tektronix engineers and NASA personnel, purchase requests for a DAS network IP address, and the 92LANP software were issued. The network connected DAS can now be accessed and controlled from a SUN workstation on the CAEDE network. FTP is used to transfer files between the CAEDE workstation and the DAS.

Test Stand

In order to connect and test the SODR ASICs, a test stand or custom circuit board was required. Time constraints precluded the design and fabrication of a custom printed circuit board. Instead, a standard IBM-AT expansion card was selected as the basis for a test stand. This was primarily due to the large number and physical size of the connectors required to attach the various probes of the DAS. This particular ASIC has 12 power and ground pairs, an 8-bit GC data path, a 16-bit SCSI data path, an 8-bit HIPPI data path, and a 32-bit memory data path (with 20-bit address) along with the associated control and handshake signals. A full length card can hold the ASIC, stimulus and output connections for all of the pins, as well as a custom 4M memory daughter board which was also designed for the SODR ASIC teststand. A diagram showing the physical layout of the SODR ASIC test stand is shown in Figure TESTSTAND. The daughter board converts the non-standard 0.05" spacing of the 64-pin SIMM memory module sockets to standard 0.1" perf-board spacing, as well as providing external chip-select logic for the four 256k x 32b SIMM memories. This board was laid out and fabricated at NASA/Langley's Microelectronic Fab Shop by Vince Cowling. The board is connected to the test stand with a 96-pin VME-type connector. By using a memory daughter board, the testing process is greatly simplified, since real memory can be used instead of trying to make the DAS emulate memory. In addition to the memory daughter board, the test stand includes wire-wrap connections between a LIF (Low Insertion Force) socket for the ASICs and the input/output connectors for the GC, HIPPI, SCSI, and memory pins of the chip. This type of setup facilitates verification of all possible data paths by reconfiguring the DAS instead of rewiring the test stand.

Another reason for using a PC type card resulted from the development of a PC-interface which allows a PC to act as the group controller for the SODR ASIC. This interface enables a PC to pre-load the memory via the GC interface and then perform a block output through either the HIPPI or SCSI interface. This should prove to be extremely beneficial during subsequent testing.

Vector Generation

The attached samples of the vector generator (Appendix B DASgen.v) and the resultant output files (*.Swav) were produced using Cadence's Verilog-XL hardware descriptive language (HDL). Vector generation was broken down into modular, reusable tasks within DASgen.v. This allowed the functional tests for all possible data paths to be included within one file. Menu selection allows the generator user to select the appropriate vector file (data path) to be created. Statements within the generator produce vectors, headers, and other information to the desired output file. As can be seen in DASgen.v, there are seven major tests, as well as tasks to read and write data at each of the interfaces. The vector generator artificially emulates the required handshake and control timing necessary to interact with different interfaces of the SODR ASIC. Much of this test is an extension of the Verilog code used in the simulational testing of the SODR chip during the design process. This vector generator created the necessary vectors in the Tektronix Swave format, while also allowing the use of the Gwave graphical output tools of Composer to view the vectors and outputs as waveforms. The vectors themselves are created by sampling the inputs and outputs of DASgen.v at every edge of a 50MHz clock. This means that the finest separation between events is 10 time units(ns).

In addition to vector files, the DAS requires a Device Map file (DMF). Device map files instruct the DAS translator how associate vector signals with specific DAS generation and acquisition probes. A sample device map file and a sample portion of an Swave file for the H2S test are included. This file along with a vector file are then sent via FTP to the DAS.

CADTRANS

Once the necessary files have been transferred to the DAS, the DAS Programmatic Command Language translator CADTRANS is invoked. This translator preforms the mapping between the Device Map file and the Swave vector files as indicated earlier. CADTRANS generates several files stored and used within the DAS. After CADTRANS successfully compiles both the vector file and the device map file, the DAS user must set up the clock speed and viewing format for the DAS signals. The master clock period is adjustable from 10ns to 1ms. The input and output signals have been mapped internally by CADTRANS on the DAS, but they must also be defined for viewing on the DAS display terminal. The display format (binary/decimal/hex) and the relative grouping of both pattern generators and acquisition signals is done in this manner. This ability was crucial due to the number of different physical connections for each of the specific tests. Once the functionality of the memory had been verified, acquisition signals could be remapped to different duties monitoring other data and control signals used in testing the major functional data paths.

Sample Device Map and Swave File Device Map Files

Device Map File

```
zeroDelayFilter = on;
bidir reference "PGCrd_" 0 pos {signal "BiGCBus" [7:0];          }
bidir reference "PDTREQ" 0 neg {signal "BiHIPPI" [7:0];          }
module "92S32-1"{
 type = s32;
 first_slot = 7;
 num_cards = 1;
 signal "BiGCBus"[7:0] {testchannel = 1 A [7:0];                 }
 signal "Clock25"      {testchannel = 1 B  8;                    }
 signal "DTREQ"        {testchannel = 1 C  8;                    }
 signal "NRDEN"        {testchannel = 1 D  8;                    }
 signal "PRegSel"[4:0} {testchannel = 1 B [4:0];                 }
 signal "PGCwr_"       {testchannel = 1 B  5;                    }
 signal "PGCrd_"       {testchannel = 1 B  6;                    }
 signal "PReset_"      {testchannel = 1 B  7;                    }
 signal "BiHIPPI" [7:0] {testchannel = 1 C [7:0];                }
 }

module "92A96-1"{
 type = a96;
 first_slot = 3;
 num_cards = 1;
 delay = 2ns;
 setup = 0ps;
 hold = 1ns;
 signal "BiGCBus"[7:0]{testchannel = 1 A0 [7:0];                 }
 signal "BiHIPPI"[7:0]  {testchannel = 1 D3 [7:0];               }
 }
```

Swave Vector File

```
version state 0 1 0 ;
signal Clock25 input;
signal PReset_ input;
signal PGCrd_ input;
signal PGCwr_ input;
signal PRegSel [4:0] input;
signal BiGCBus [7:0] bidir;
signal PDTREQ input;
signal PNRDEN input;
signal PDREQ output;
signal PSELB [2:0] input;
signal BiHIPPI [7:0] bidir;
  0:  0 1 1 1 xxxxx zzzzzzzz xxxxxxxx 0 1 0 101 zzzzzzzz xxxxxxxx;
  1:  1 1 1 1 xxxxx zzzzzzzz xxxxxxxx 0 1 0 101 zzzzzzzz xxxxxxxx;
  2:  0 1 1 1 xxxxx zzzzzzzz xxxxxxxx 0 1 0 101 zzzzzzzz xxxxxxxx;
  3:  1 1 1 1 xxxxx zzzzzzzz xxxxxxxx 0 1 0 101 zzzzzzzz xxxxxxxx;
  4:  0 1 1 1 xxxxx zzzzzzzz xxxxxxxx 0 1 0 101 zzzzzzzz xxxxxxxx;
  5:  1 1 1 1 xxxxx zzzzzzzz xxxxxxxx 0 1 0 101 zzzzzzzz xxxxxxxx;
  6:  0 1 1 1 xxxxx zzzzzzzz xxxxxxxx 0 1 0 101 zzzzzzzz xxxxxxxx;
  7:  1 1 1 1 xxxxx zzzzzzzz xxxxxxxx 0 1 0 101 zzzzzzzz xxxxxxxx;
  8:  0 1 1 1 xxxxx zzzzzzzz xxxxxxxx 0 1 0 101 zzzzzzzz xxxxxxxx;
  9:  1 1 1 1 xxxxx zzzzzzzz xxxxxxxx 0 1 0 101 zzzzzzzz xxxxxxxx;
 10: 0  1 1 1 xxxxx zzzzzzzz xxxxxxxx 0 1 0101 zzzzzzzzz xxxxxxxx;
```

Testing

Eight major tests were conducted. These tests checked four major data paths:

GC and Memory:	GC to Memory, Memory to GC, GC Read/Write, Transfer Counter
GC and SCSI:	GC to SCSI, SCSI to GC
SCSI and HIPPI:	SCSI to HIPPI
HIPPI and SCSI:	HIPPI to SCSI.

In order to simplify operational testing, sections of the chip were tested individually whenever possible. To test the operation of the memory daughter board (and to simplify subsequent tests) the GC and Memory interfaces were tested first. Once they had been proven functional, the SCSI I/F (working with the GC I/F) was tested. Lastly the HIPPI I/F was tested in combination with the SCSI I/F by means of HIPPI->SCSI and SCSI->HIPPI tests. Tests of GCtoHIPPI and HIPPItoGC were not conducted due to the size of the burst (256 words) which had to be sent through any HIPPI data transfer. The large number of vectors required to test this non critical path precluded further testing. The details and specifics of each test can be seen in the attached DASgen.v file.

GC / Memory Testing

The GC to Memory, Memory to GC, GC Read/Write, and Transfer Counter tests were done first in order to verify functional operation and speed limitations of the memory daughter board, the GC interface, and the memory interface. These tests write and read different data at a variety of addresses and address ranges. In addition, the operation of the status register and operation of the memory at page boundaries is tested. These tests showed that the memory and GC performed suitably at 50MHz in the following areas:

status register operation with respect to bits indicating memory full and memory not empty,
memory operation at page boundaries,
memory interface operation relative to use with the GC,
GC interface operations,*
transfer counter operation.

The GC interface is used to setup, control, and monitor the SODR chip. The interface is a memory mapped register design. A register map, regmap.sodr, (ASIC DataSheets) is included in this report. The chip can be reset by writing a reset code to the GC mode register, or by sending the Reset_ pin low. Memory is written by writing four 8-bit patterns to a 32-bit MtoGC data register, three 8-bit patterns to a 20-bit address register, and then issuing a code to write the contents of the memory data register at the memory address register specified. Figures GC2M[a:c] show sample read/write operations of the GC interface.

Figure GC2M[a] shows a hardware reset (Reset_), a preliminary read of the status register (RegSel -> 0x0B), three software resets (RegSel -> 0x0C), and then a memory write of data 0x04030201 (RegSel->0x14, 0x15, 0x16, 0x17) to address 0x00000 (RegSel->0x0D, 0x0E, 0x0F).

Figure GC2M[b] shows the last part of a write to address 0x00004, and then a read of address 0x00000. The memory address registers (0x0D, 0x0E, 0x0F) are loaded with address 0x00000, then a GC interface load code for the MtoGC register is issued (RegSel->0x19), and finally the four MtoGC data registers(RegSel->0x10, 0x11, 0x12, 0x13) are read out one by one. The data bus reflects inverted data values which have been stored in memory. This inversion is due to the fact that the SODR ASIC contains inverting output pads. This is not a problem when the chip is used in one of it's two primary configurations: HIPPI->SCSI or SCSI->HIPPI. In these two modes the memory is transparent and the inversion is not visible. When the GC is used as either a input or output path the data will appear reversed at the memory. This feature is not a problem since the use of the GC as a data port to access memory is primarily a debugging feature of this chip.

Figure GC2M[c] shows the write of data 0x04030201 at addresses 0x00000 through 0x00004. In addition to the address changing, the memory output enable (Mem_oe_) and the memory write (Mem_wr_) signals can also be seen toggling during each of the five writes in this figure.

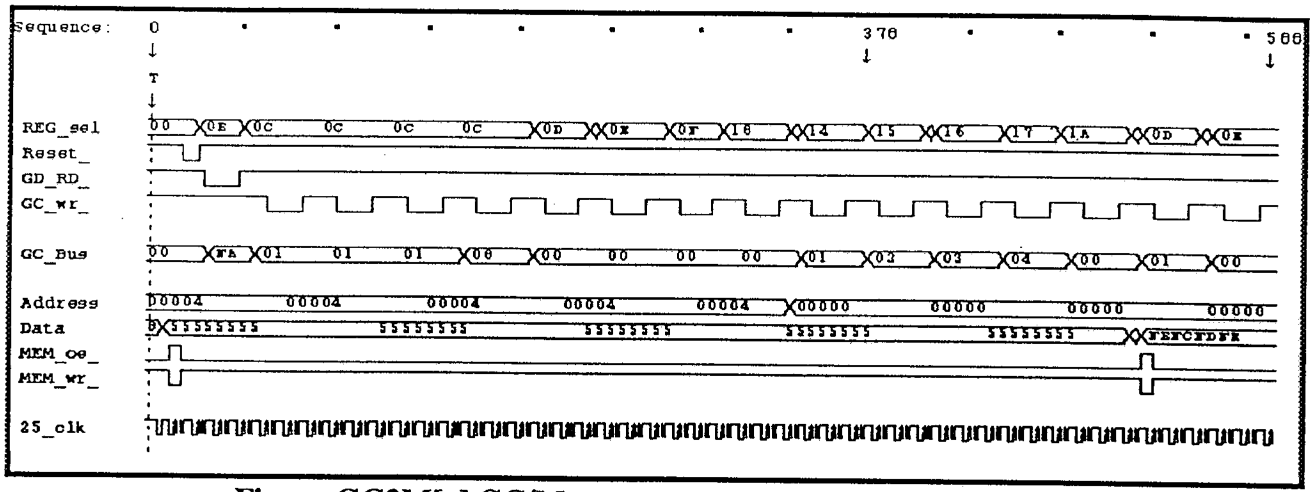


Figure GC2M[a] GC/Memory test startup and memory write

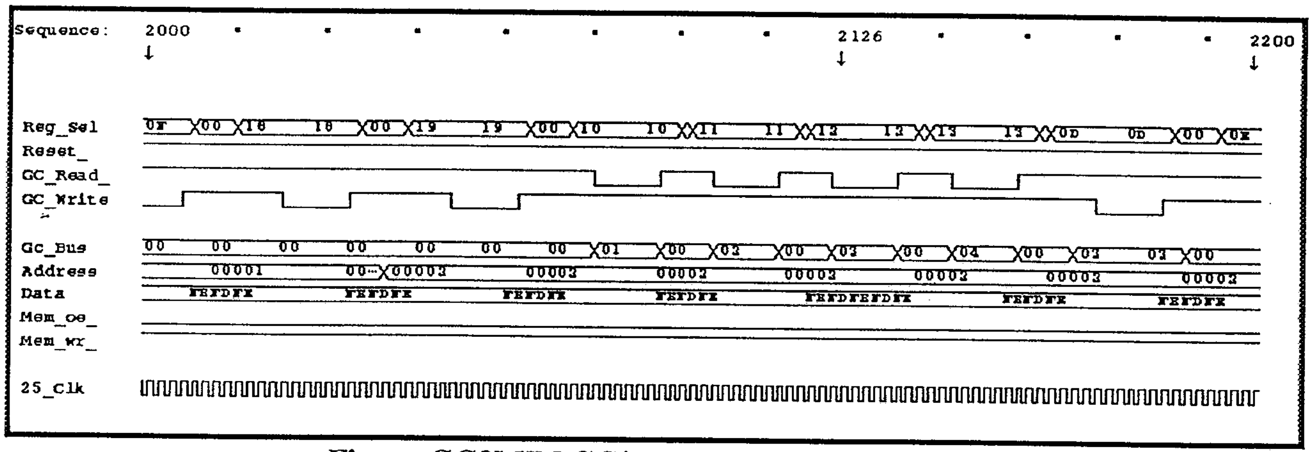


Figure GC2M[b] GC/Memory test data readout

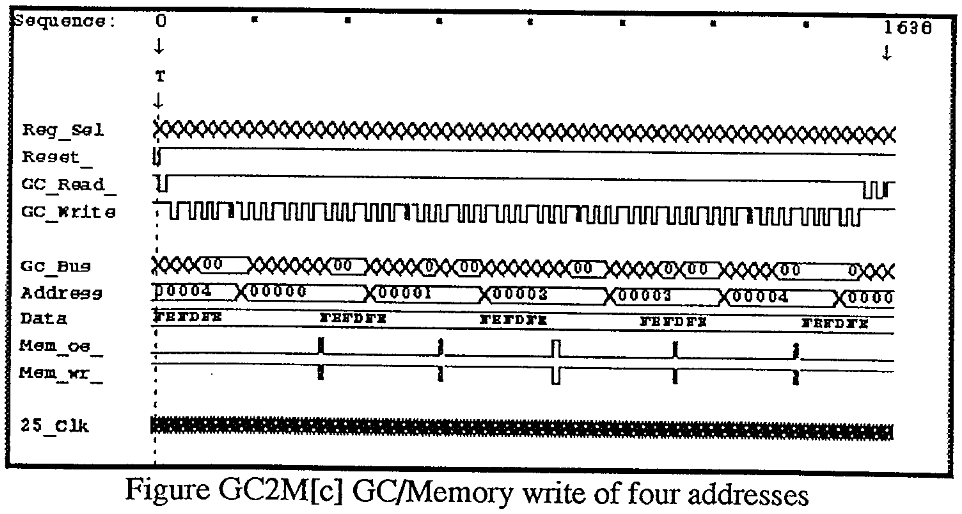


Figure GC2M[c] GC/Memory write of four addresses

Figures GC2M[a:c]

*Ringing problems were encountered intermittently during some READ operations of the GC interface's MemtoGC registers. These oscillations appear to be related to the test stand and the DAS. Exhaustive debugging was done to find the cause of this ringing. The noise seemed to occur primarily during a read of MemtoGC address 0x13, but was also observed during READ operations at MemtoGC addresses 0x10, 0x11, and 0x12 as well. Figures NOISE[a:b] show the results of the noise as captured on the DAS during a variety of tests. In Figure NOISE[a] the noise has caused data on GC_Bus to appear to be 0x00 when it should really be 0x76. The data pattern written was 0x76543210. Figure NOSIE[b] on the other hand, shows where the GC_Bus data is stable long enough to see the proper value 0x76 appear momentarily. Figures NOISE[c:f] show the noise as captured by the 200MHz digital storage scope. Figures NOISE[c:f] show the GC_RD_ signal on top and a bit of the GC_Bus on the bottom. These four low READ signals were captured during the read of the four M2GC registers (0x10-0x13). Note in Figures NOISE[c:d] how the noise occurs on both one READ (0x13) and all four READs. Figure NOISE[e] shows the oscillations as they were captured with the DAS in a static state, while Figure NOISE[f] shows four READs without any noise. Many hours of time were spent trying to resolve this problem. It was found that the chip would remain in this ringing state even with the chip and the DAS in a static mode (no inputs changing), and would continue to oscillate with practically all input and output probes removed from the teststand. Since the reading of data through the GC interface is primarily a debug feature of this chip and the data appeared to be intact within the affected registers, further detailed analysis of this phenomenon was not continued.

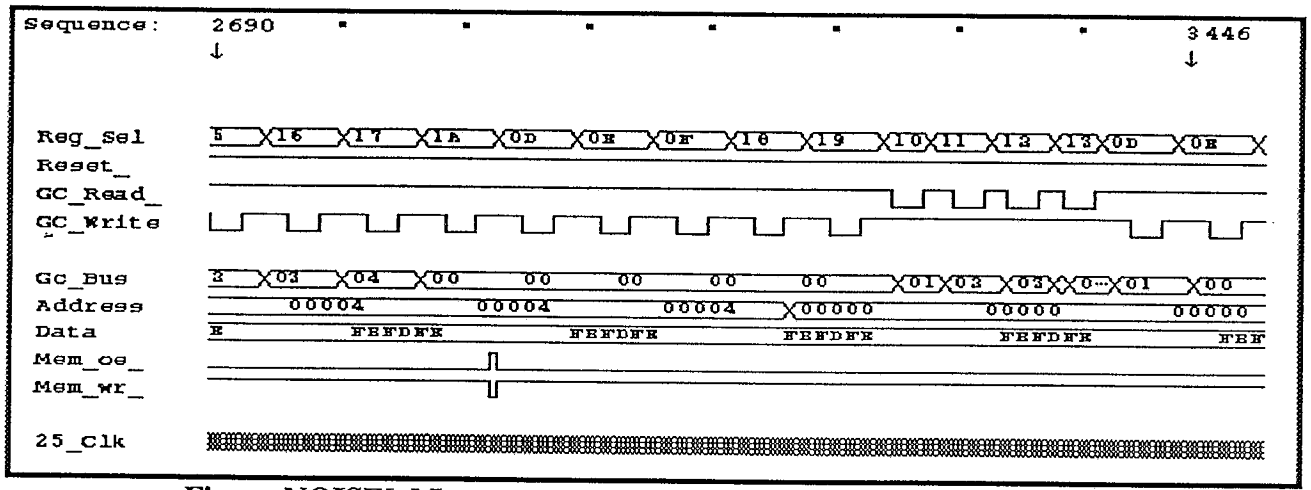


Figure NOISE[a] Incorrect data from register 0x13 (should be 0x76).

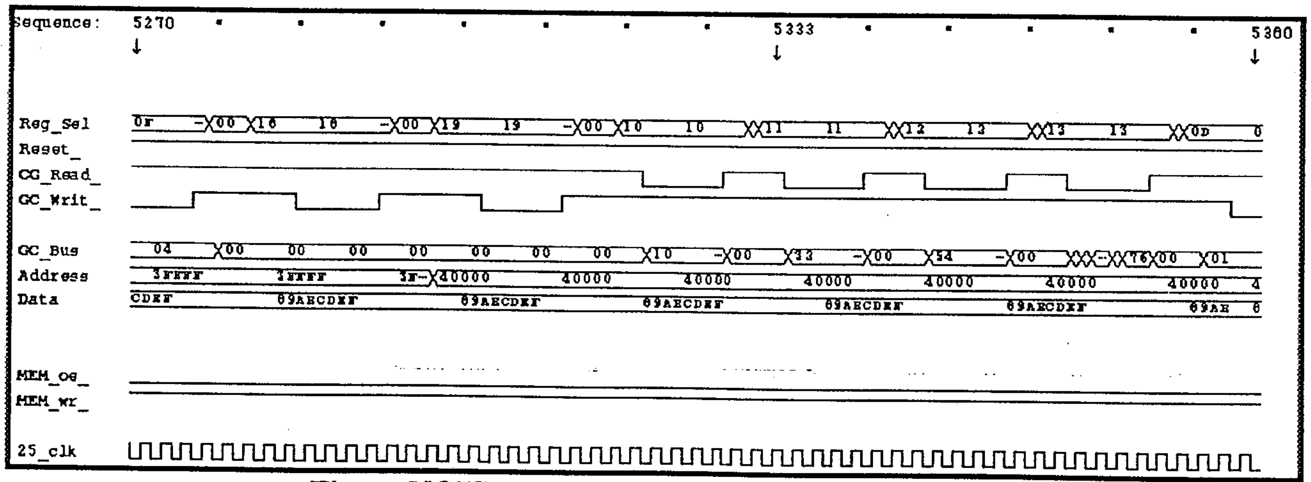


Figure NOISE[b] Correct data 0x76 for brief time.

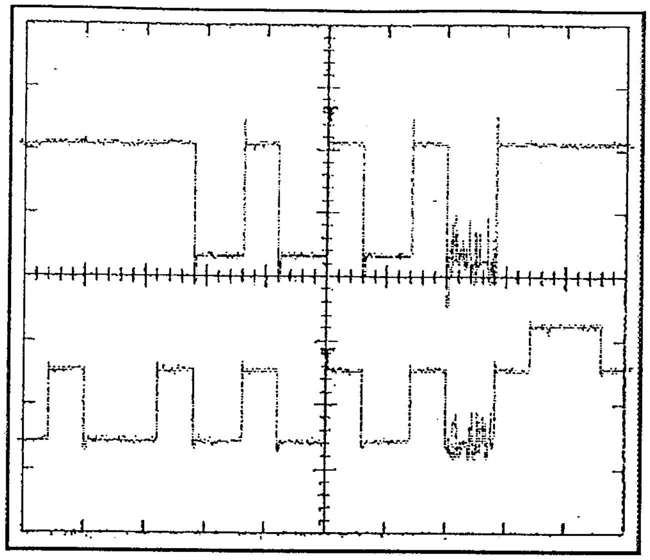


Figure NOISE[c]

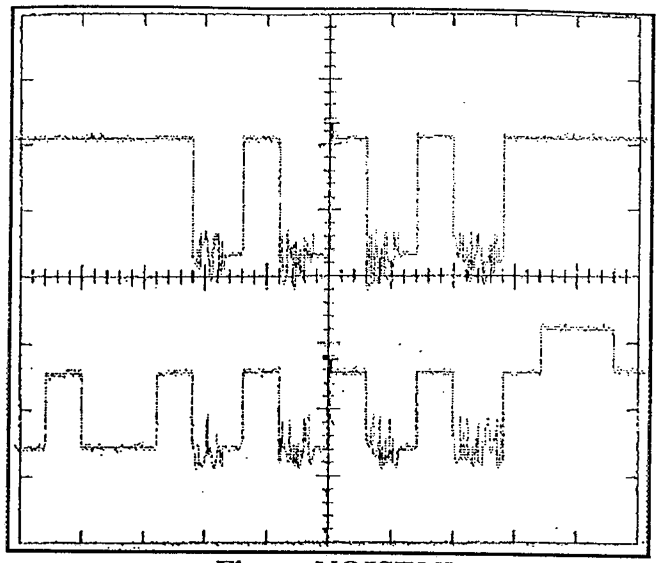


Figure NOISE[d]

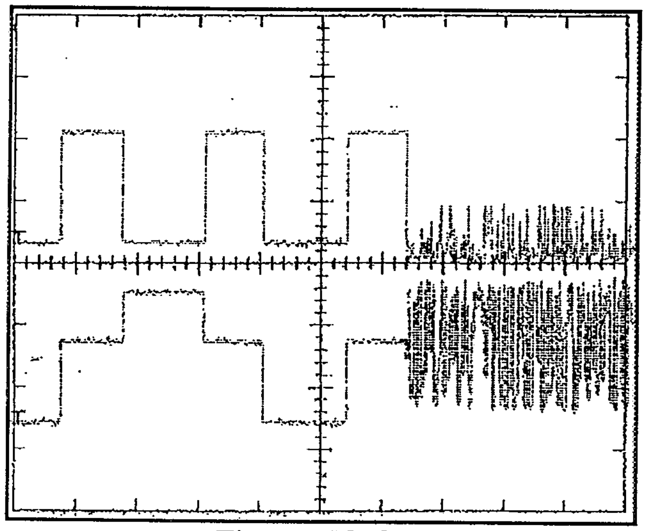


Figure NOISE[e]

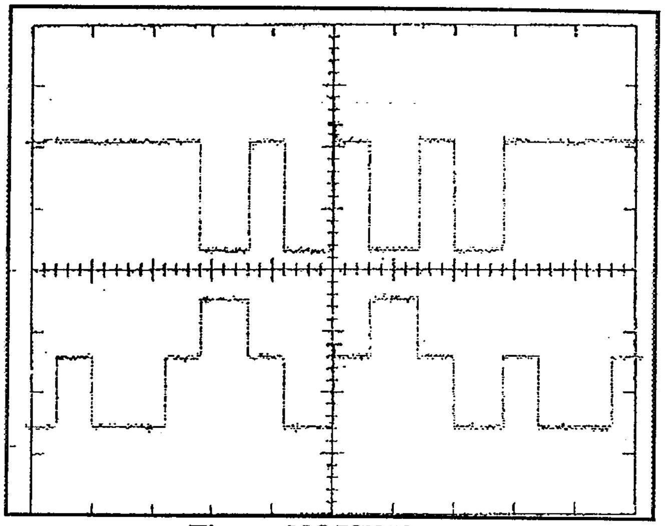


Figure NOISE[f]

Figure XFER[a] shows the WRITE and READ of the five transfer counter registers (RegSel-> 0x0A-0x05). The transfer counter keeps track of the number of words transferred during HIPPI->SCSI and SCSI->HIPPI data transfers. This counter was functional at speed with respect to writing, reading, and automatic incrementation.

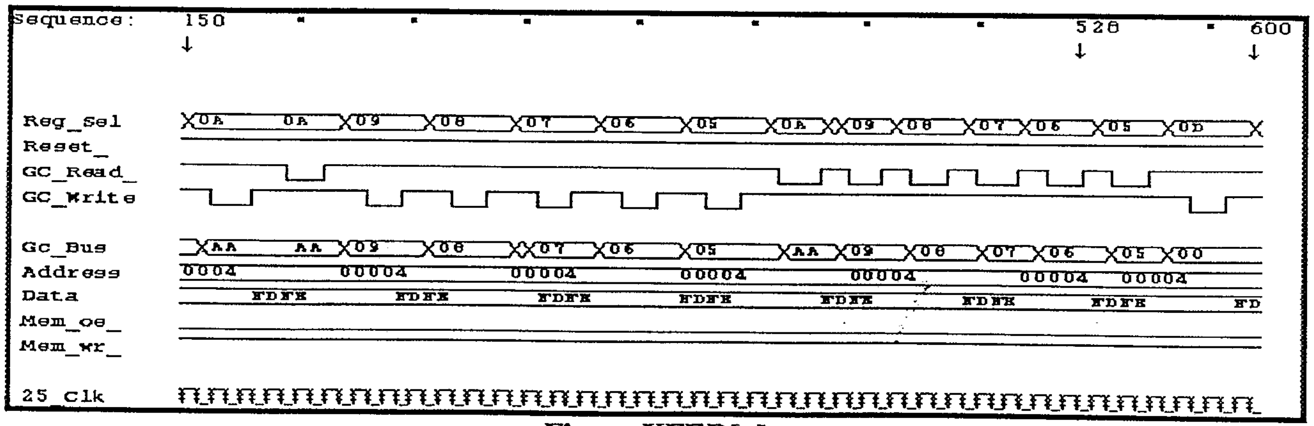


Figure XFER[a]

GC / SCSI Testing

After the functional operation of the memory and GC interfaces had been verified using the three GC/Memory tests, the GCtoSCSI test was developed. The relatively simple DMA-style interface used by the SCSI interface was the next logical feature to be tested. Instead of using the special single-bit, interfacing/handshaking pod for the DAS (which was delivered late to the project), the GCtoSCSI and SCSItoGC tests were written so that the tests reflected the response capabilities of the chip. By allowing the appropriate response time for acknowledgment signals (DACKN) to read and write requests (DREQ) from outside the chip, the read and write capabilities of the SCSI interface portion of the chip were verified. (Some noise problems were noted again during GC read operations.)

The GCtoSCSI test involved resetting the chip, loading data into memory, setting the chip into GCtoSCSI mode, and then monitoring the data subsequently output from the SCSI interface. Figure GC2S[a] shows the loading of data 0x04030201 (inverted 0xFBFCFDFE) at addresses 0x00000 through 0x00004, and the resultant SCSI output data along with the DREQ, DACKN, and WRN_ signals used by the SCSI interface is show in Figure GC2S[b]. An oscilloscope print-out showing a detail of the DREQ and DACKN timing at 1Mhz is included as Figure GC2S[c]. The GCtoSCSI test was functional at 50MHz other than the occasional loss of a half word during the read of the last word from the SCSI I/F. This may be due to the timing associated with the DREQ and DACKN signals which were not actually handshaking. Internal SCSI I/F FIFO access is directly controlled by these signals so failure to issue a DACKN for a data request (DREQ) would trigger the appropriate response from the SCSI DMA interface. This is not believed to be a functional problem.

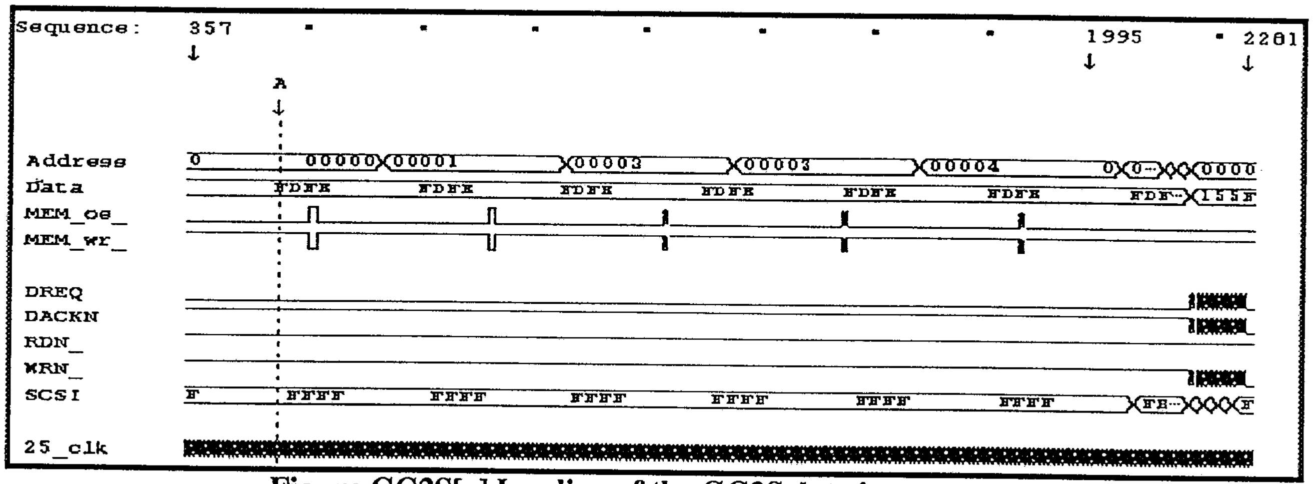


Figure GC2S[a] Loading of the GC2S data into memory.

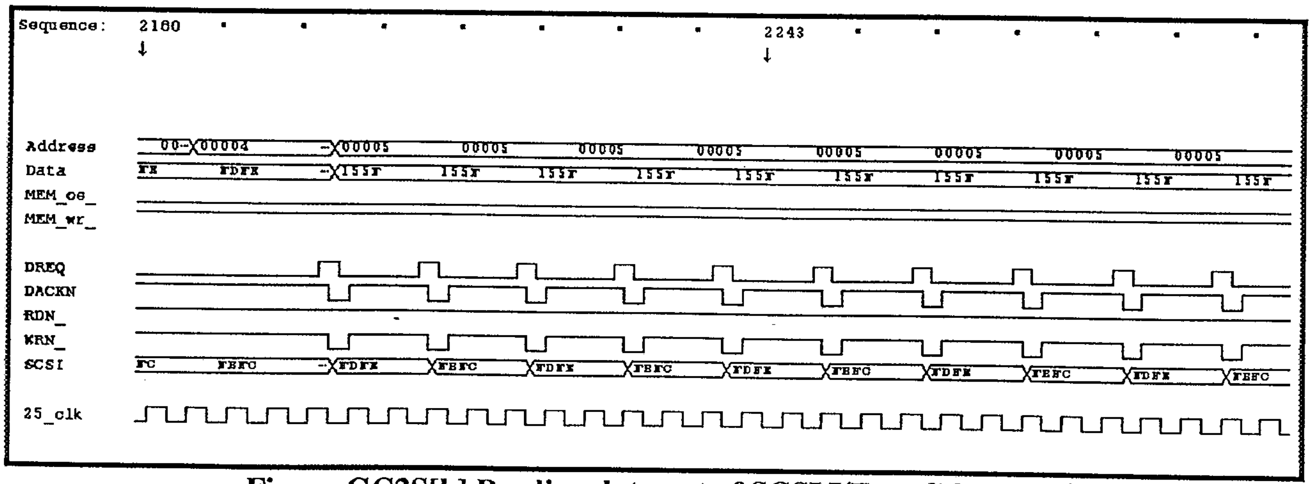


Figure GC2S[b] Reading data out of SCSI I/F on GC2S test.

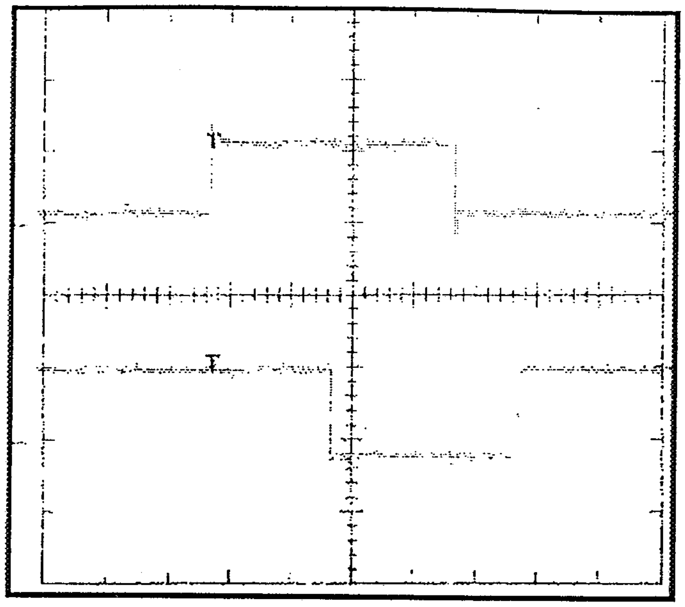


Figure GC2S[c] Details of the SCSI I/F DREQ (top) and DACKN (bottom) signals (oscilloscope).

The SCSItoGC test involved resetting the chip, writing the SCSI I/F with data, setting the chip into SCSItoGC mode, and then doing GC/Memory-style read operations from the memory. The SCSItoGC mode of operation places SCSI data in memory in anticipation of outputting the data through the HIPPI I/F. This test also displayed the same intermittent noise problems seen during the GC/Memory tests. Again it appeared that the data was successfully transferred despite the noise monitored both by the DAS and an oscilloscope during read-out of the SCSI data through the GC interface. Figure S2GC[a] shows the data (0x04030201) being written into memory through the SCSI I/F. Both the SCSI I/F signals (DREQ, DACKN, RDN_) and Memory I/F signals (Mem_oe_, Mem_wr_) can be viewed as the data travels into the SCSI I/F and out the Memory I/F. The Data signal reflects the lower half-word of the value written into memory.

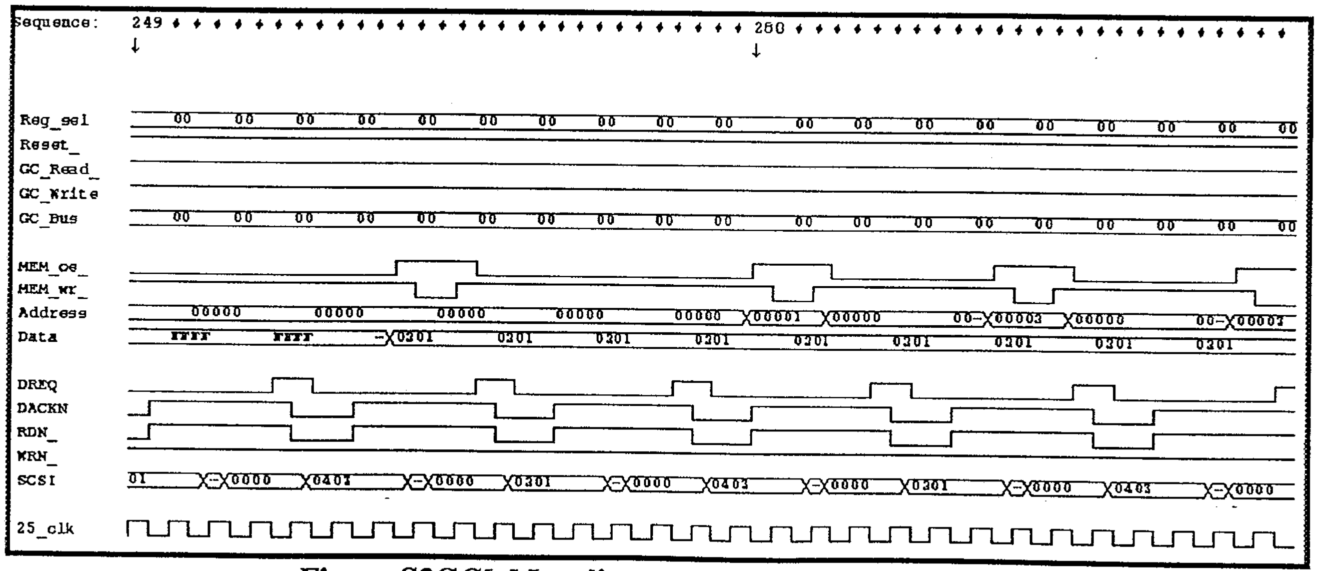


Figure S2GC[a] Loading memory during S2GC test.

HIPPI / SCSI Testing

Once the GC, Memory, and SCSI interfaces had been tested incrementally, all that remained was to implement a test of the HIPPI I/F and to integrate the tests to finally test out the two primary data paths used by the SODR ASIC (HIPPI->SCSI and SCSI->HIPPI). Both of these tests involve resetting the chip, loading the data through either the HIPPI or SCSI port and then setting the chip into the appropriate mode to read data at the other port.

The H2S test appears to be very sensitive to the shape of the two clock signals (25MHz ASIC clock and the 50MHz SCSI clock) and to the relative timing of the SELB, WRCLK, and the first HIPPI byte. These clock problems could probably be cleared up using a ground plane circuit board and a clock oscillator chip instead of the DAS clock supplied at the probes. The timing of the SELB and WRCLK signals would be provided by the actual HIPPI source and destination chips in an actual circuit. The resolution of the DAS vectors and the DAS acquisition timing may have played a role in problems encountered during this test. Finer resolution would have moved events off of clock edges. There were some successes with HIPPI->Mem and Mem->SCSI transfers, and although the first byte was occasionally dropped before getting to memory there were successful tests in which all of the data was successfully passed through the chip. Successful data transfers were obtained at clock speeds up to 1MHz. Again the clock appeared to have a significant effect upon the transfer of data. Details of the two clocks at 1MHz (clean), and 10MHz (spikes) can be seen in Figures H2Sclock[a,b].

Figure H2S[a] show the beginning of the loading of HIPPI data to memory. The HIPPI signal shows the data pattern being clocked into the HPPPI I/F while the Data signal shows the lower half of data being written to memory starting at address 0x00000. The HIPPI data are bytes whose value increase by one for each byte (ie. 0x01, 0x02, 0x03 ...). This figure also shows how the first byte of HIPPI data was dropped by the HIPPI I/F. Figure H2S[b] shows the beginning of a HIPPI->SCSI test in which the first byte was not dropped. (This can be seen were the upper bytes of memory data at address 0x00000 are 0x0403 which left 0x0201 in the lower bytes.) Figure H2S[c] shows the beginning of the data output through the SCSI I/F. The SCSI data can be seen incrementing properly (other than the first byte) in this figure. The address does not change during this portion of the test since the SCSI data is being read from the SCSI I/F FIFO.

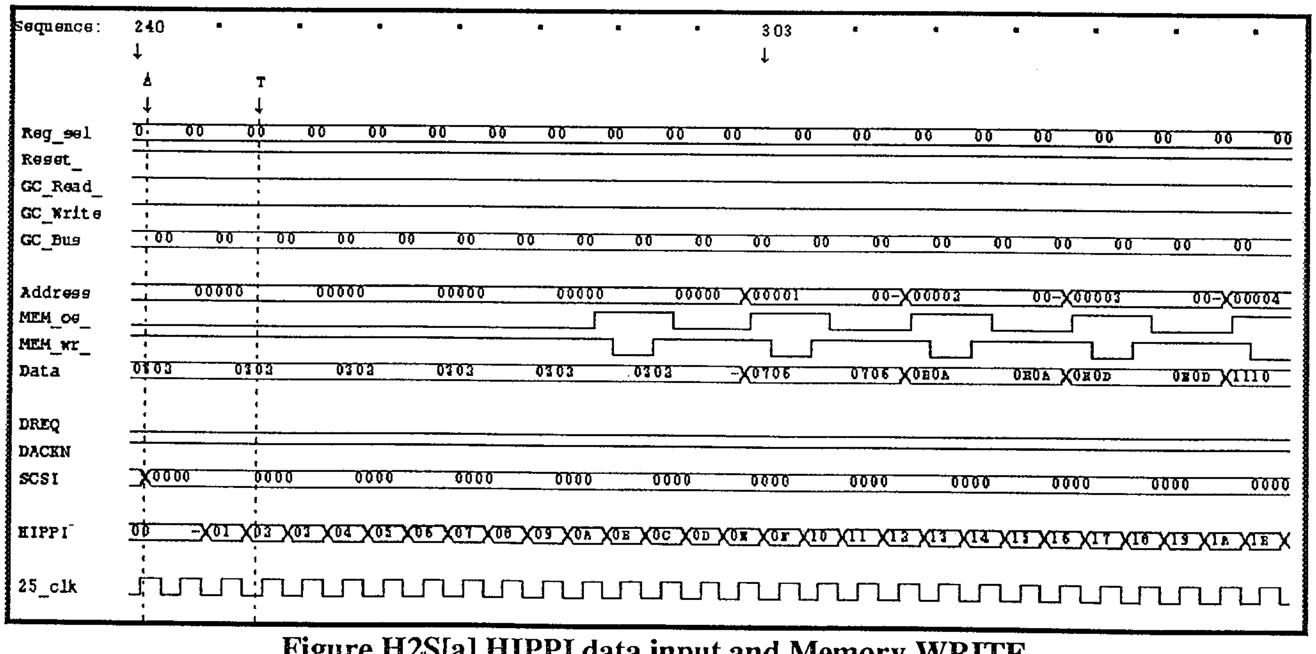


Figure H2S[a] HIPPI data input and Memory WRITE.

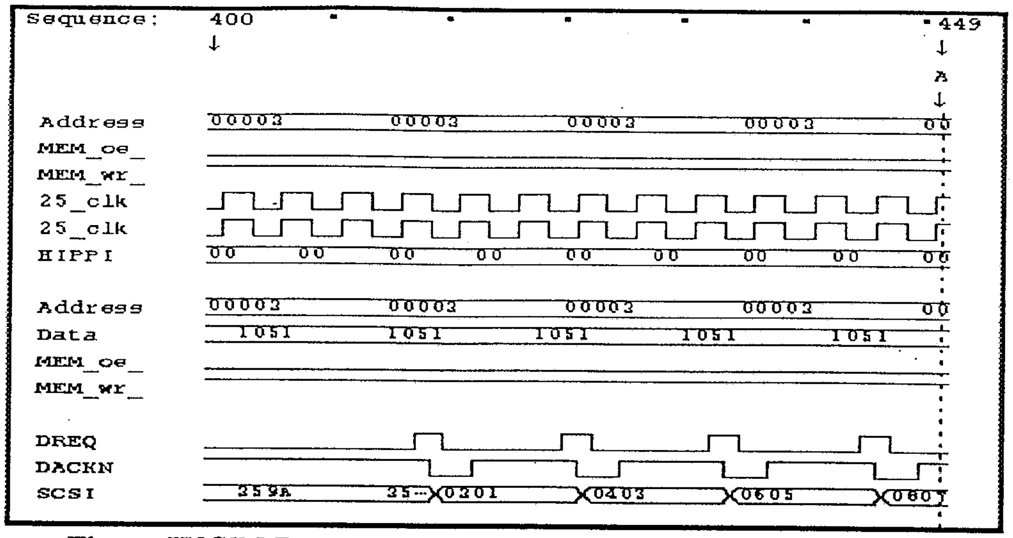


Figure H2S[b] Proper start of SCSI output. No loss of first byte.

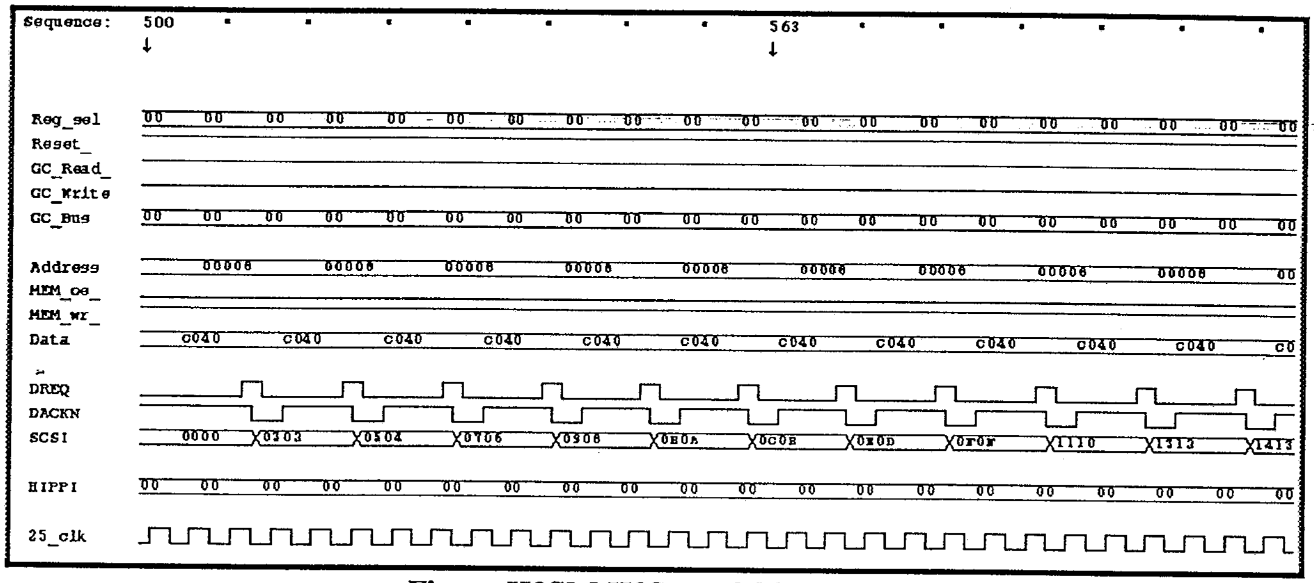
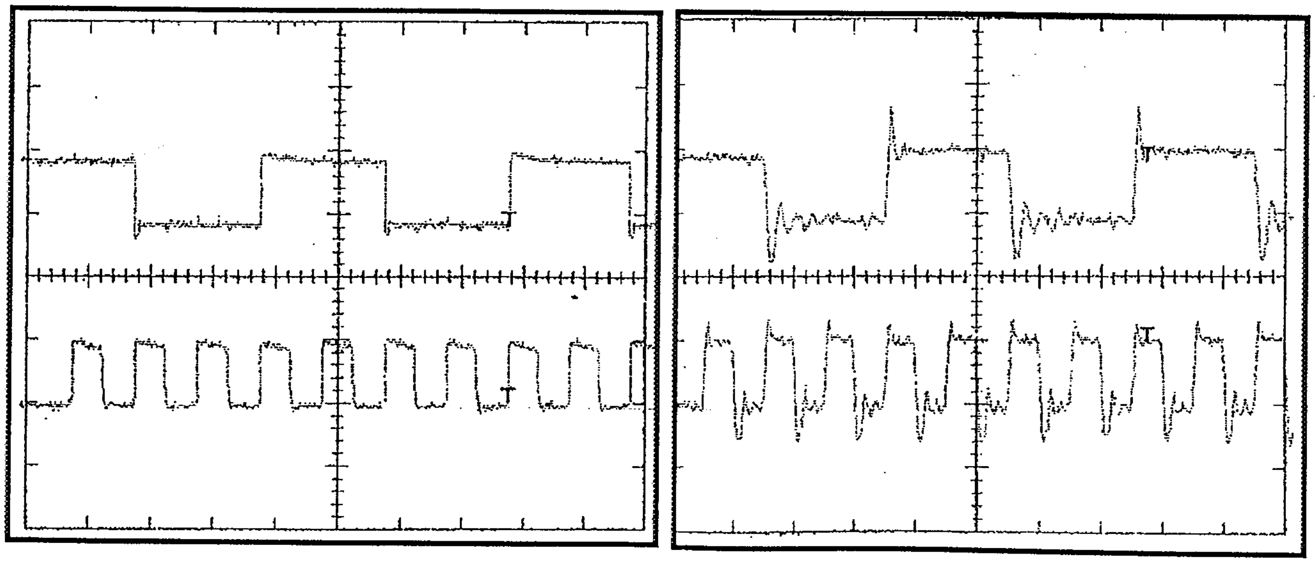


Figure H2S[c] H2S test SCSI output.



Figures H2Sclock[a,b] Clean 1MHz clock and noisy 10MHz clock.

The SCSItoHIPPI test was successful at 50MHz. This test involves writing data into the SCSI I/F and then emulating timing of the HIPPI Source/Dest chips to read data out of the HIPPI I/F. Figure S2H[a] shows the last part of data 0x04030201 being written into memory through the SCSI I/F. Figure S2H[b] shows the IFIELD value 0x49 (0xB6) on the HIPPI data bus and then the read out of HIPPI data which appears as bytes 0x01, 0x02, 0x03,. 0x04. This test perfromed suitably at a speed of 50MHz without problems.

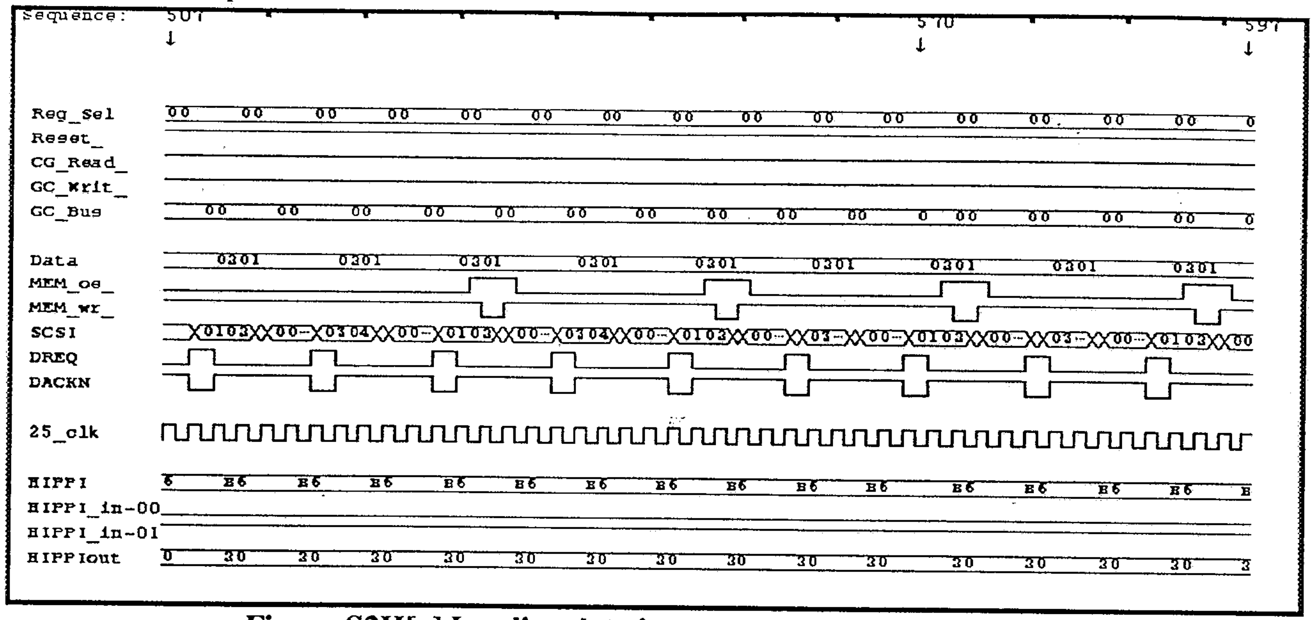


Figure S2H[a] Loading data into memory through SCSI I/F.

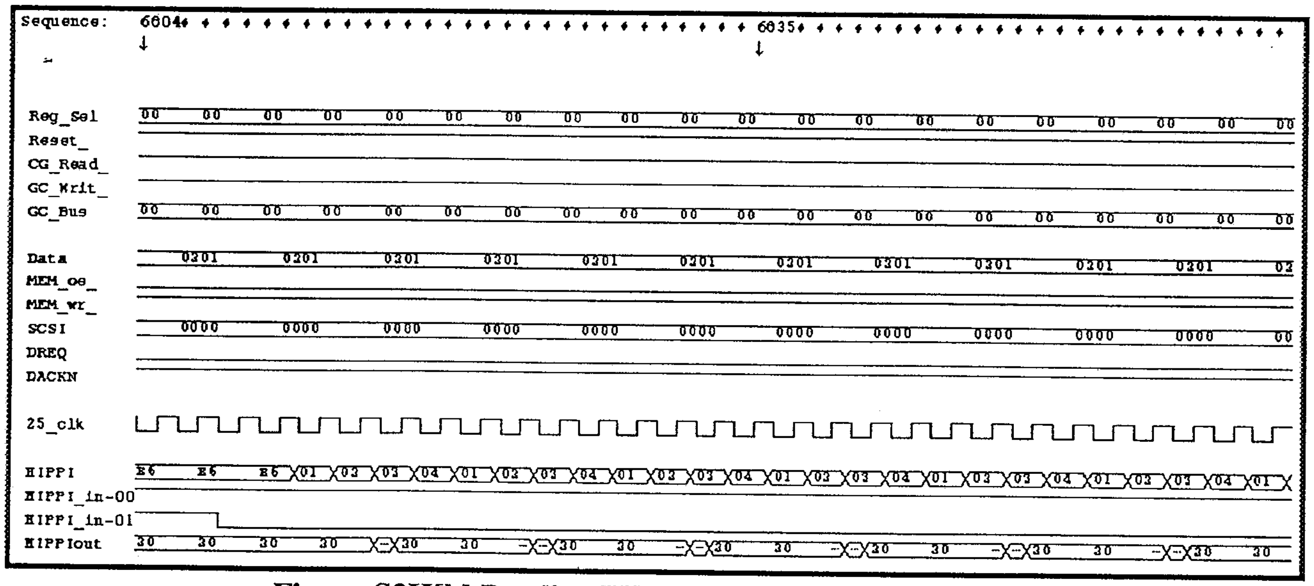


Figure S2H[b] Reading HIPPI data out during S2H test.

Results

The DAS test results have shown that all of the interfaces in the SODR MBC ASIC to be operational at speeds up to 1us (1MHz). Clock noise on the DAS probes, as well as timing, memory, and setup limitations of the DAS, prevented successful tests at 50MHz for most tests of the major data paths. Many of the problems associated with the noise due to the clocks derived from the DAS as well as that introduced by the length and number of wire-wrap connections can be reduced or eliminated with the use of a ground plane circuit board and clock oscillator circuit. The memory limitations on the DAS cards currently installed (8Kx36b) means that the results of long tests cannot be captured completely. Instead, triggers must be set so that the desired portions of test can be captured. For large tests, this may mean several runs to verify all portions of such tests. The fastest speed at which wide (32b) data can be acquired is 50MHz. Since this was the SODR ASIC's design speed, our sampling resolution was only two samples per clock cycle. Ideally more samples should be taken every clock cycle in order to monitor the response timing of the SODR interfaces.

Despite the limitations of the DAS (and our knowledge of it), and the problems with the teststand, success was made in functionally testing these chips. Results from these tests indicate that all of the SODR ASIC's data paths will operate at the 50MHz design speed with the use of dedicated clock circuitry and a custom printed circuit board populated with the actual HIPPI Source and Destination chips. Noise problems encountered during tests involving the GC interface are believed to be inconsequential. A table of results is included below.

Test	**Best Speed**	**Problems**	**STATUS**
GCtoMem	50MHz	NONE	Functional
MtoGC	50MHz	Intermittent noise	Functional
GC Read/Write	50MHz	Intermittent noise	Functional
Xfer	50MHz	NONE	Functional
GCtoSCSI	50MHz	NONE	Functional
SCSItoGC	50MHz	Intermittent noise	Functional
HIPPItoSCSI	1 MHz	Lost first byte, clock noise	Functional
SCSItoHIPPI	50MHz	NONE	Functional

IX. Recommendations

1) The design process with the front end work provided by NASA and the layout and fabrications performed by US2 was shown to be viable. Although, two tools could be added into the design flow to provided greater confidence in the final devices. A fault coverage tool for generating test vectors and a critical timing tool for identifying critical data paths.

2) When working closely with a West coast company it is important to establish good communication early on. Phone, fax, express mail, e-mail, and ftp will all be required over the course of a design. Electronic communications are very important for information exchange, and were a problem during this project.

3) It is important to freeze the software version and operating systems version of the system unless there is a compelling reason to do otherwise. Systems upgrades are never easy with EDA tools and seem to always result in at least a week of down time.

4) Technical management must be setup before work starts on the project. It should be clear who has responsibility for what portions of the design work and who has authority over the designers for technical decisions. Technical reviews should also be regularly scheduled.

X. Conclusions

All work associated with the SODR Memory Buffer Control ASIC Research Grant has been completed. Specifically:

1) A 144 pin CMOS ASIC was designed and simulated.
2) Ten prototypes were fabricated by US2 and passed acceptance testing.
3) Independent test were run at NASA using a Textonics DAS.
4) The ASIC design process was effectively exercised using NASA tools for front end design and US2 for layout and fabrication.
5) All work was was document internally and through publication.
6) This final report was completed and submitted to NASA.

XI. Acknowledgments

This work was a collaborative effort with personnel from NASA, CNU, US2, SAIC other organizations. I would like to give special thanks to Tom Shull, Jerry Tucker, Steve Jurczyk, Glenn Hines, Steve Campbell and Steve Comer for their help and support throughout this project.

Appendicies

A) Schematics

1) SODR MBC Asic (Hierarchical)
2) HIPPI Verilog Models

B) Test Files

1) SODR Functional Test Stimulus File
2) SODR Acceptance Test Stimulus File (US2)
3) SODR DAS Test File

C) Related Publications

1) *A Spacecraft Mass Storage Optical Disk System*, 12th IEEE Symposium on Mass Storage Systems, G. Hines, S. Jurczyk, R. Hodson
2) *An ASIC Memory Buffer Controller for a High Speed Disk System*, 5th NASA Symposium on VLSI Design, R. Hodson, S. Campbell

D) DAS Operation Overview

E) Related Data Sheets

1) IDT 7MP4045 256Kx32 CMOS Static RAM Module
2) AMCC S2020/S2021 HIPPI Source/Destination
3) EMULEX FAS366 SCSI II Processor

Readers are hereby informed that appendices have been omitted due to their aggregate bulk.

Interested persons should avail themselves of a copy of the appendices, if needed, by contacting:

Office of Sponsored Programs
Christopher Newport University
50 Shoe Lane
Newport News, Virginia 23606-2998

Telephone: (804) 594-7266 Facsimile (804) 594-7713

When requesting a copy please refer to NASA GRANT NUMBER NAG-1-1439, PI Dr. Robert F. Hodson, account number 33170