SPARCclassic Engine OEM Technical Manual

PART THREE:

Appendix M

I/O Chipset Reference Material

Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

Part No. 801-3137-10
Revision A, April, 1993

NCR I/O Chip Specifications

M.1 NCR89C100 (MACIO)

The NCR Corporation Master I/O Controller device 89C100 specifications (including functional block specifications) are in this appendix. This device is also known by the Sun Microsystems code name MACIO (MAster Controller I/O). Use this specification to determine the operating parameters of the device.

The NCR documentation included here is accurate as of the date of release of the SPARCengine EC OEM Technical Manual. Please call NCR to ensure that you have the most current documentation. Please use caution in developing plans on this information until you confirm it is the latest information available.

M.2 NCR89C105 (SLAVIO)

The NCR Corporation Slave I/O Controller 89C105 specifications (including functional block specifications) are in this appendix. This device is also known by the Sun Microsystems code name of SLAVIO (SLAVe controller I/O). Use this specification to determine the operating parameters of the device.

The NCR documentation included here is accurate as of the date of release of the SPARCengine EC OEM Technical Manual. Please call NCR to ensure that you have the most current documentation. Please use caution in developing plans on this information until you confirm it is the latest information available.

Product Disclaimer

The NCR SBus I/O Chipset (NCR89C100 and NCR89C105) and NCR SBus Demonstration Board were **NOT** designed as fault-tolerant devices and are **NOT** designed or intended for use or resale in or as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or potentially life-threatening devices, systems or procedures (e.g. devices used in medical diagnostic applications, used in life-sustaining equipment, used in connection with surgical or other intrusive procedures, or otherwise used to support or sustain life or implement medical procedures), or weapons systems, in which the failure of Products could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). NCR specifically disclaims any express or implied warranty of fitness for High Risk Activities.

About this Version

This is Revision 1.0 of the *NCR SBus I/O Chipset Data Manual.*

The areas that were updated from the previous version are as follows:

NCR89C100 Master I/O section

- Additional timing diagrams were added.
- Packaging information was added.
- Operating conditions, AC and DC characteristics were updated.

NCR89C105 Slave I/O section

- Existing timing diagrams were modified.
- Additional timing diagrams were added.
- Packaging information was added.
- Operating conditions, AC and DC characteristics were updated

NCR53C9X SCSI Core

- Specification was updated to include enhanced feature set.

Application Notes

- Additional information on I/O Demonstration Board was included.
- The "Frequently Asked Questions" section was expanded.

NCR SBus I/O Chipset Data Manual
Contents

March 1993

DMA2 DMA Core

NCR92C990 Ethernet/IEEE 802.3 LAN Controller

NCR 53C9X Advanced SCSI Controller

NCR89C105 Chip Specification

NCR85C30 Serial Communications Controller

NCR82077 Floppy Disk Controller Core

Application Notes

NCR89C100 Chip Specification
Contents

List of Figures

List of Tables

NCR89C100
Chip Specification

Overview

The NCR89C100 is designed for low-cost, SBus-based systems. It incorporates standard workstation I/O devices with a DMA controller in a single 160-pin PQFP package, providing cost, area, and power savings over discrete implementations.

The 89C100 provides three special purpose SBus DMA channels that are commonly used on SPARC® platforms: Ethernet, SCSI, and Parallel Port. It consists of three major logic blocks: DMA2, ENET, SCSI as well as an additional TEST block. The DMA2 block provides internal buffering for each of its three channels in the form of a cache for the ENET interface and fifos for the SCSI and Parallel Port interfaces. It also provides control/status registers for each channel, plus several SCSI/PPORT-specific support registers, and a write buffer for slave accesses to the ENET. The DMA2 block design is based on the L64853[1] ASIC design with the addition of a programmable, Centronics-type parallel port. It incorporates a number of new features for increasing performance and allowing different modes of operation necessary for future desktop systems. The ENET block is based on the NCR92C990 Application Specific Function (ASF) which is a superset of (and fully backwards compatible with) the AM7990[2] previously found on SPARCstations. The SCSI block is based on the NCR53C9X[3] ASF which is a superset of (and fully backwards compatible with) the NCR53C90A also found on SPARCstations. The TEST block contains the JTAG TAP controller, JTAG boundary scan cells, ASF test muxes and some ancillary glue logic.

The 89C100 interfaces directly to the SBus with no additional glue logic. Together, with the 89C105 (slave I/O), it provides the core SPARCstation I/O subsystem.

This document reflects the integrated nature of the 89C100. This section, "NCR89C100 Master I/O", covers the chip as a whole and describes pinout information, test muxing, chip-level block diagrams and address map, and electrical and mechanical characteristics. The TEST block is described in detail, but DMA2, ENET, and SCSI blocks are only introduced and a list of differences from their discrete implementations is given. The full specifications for those discrete implementations follow in the next three sections: "DMA2 DMA Core," "NCR92C990 Ethernet Core," and "NCR53C9X SCSI Core". These specifications cover functional descriptions and theory of operations.

---

1. LSI Logic Corp.
2. Advanced Micro Devices, Inc.
3. The NCR53C9X is identical to the FAS101 licensed from Emulex Corporation.

Chip-Level Functional Block Diagram

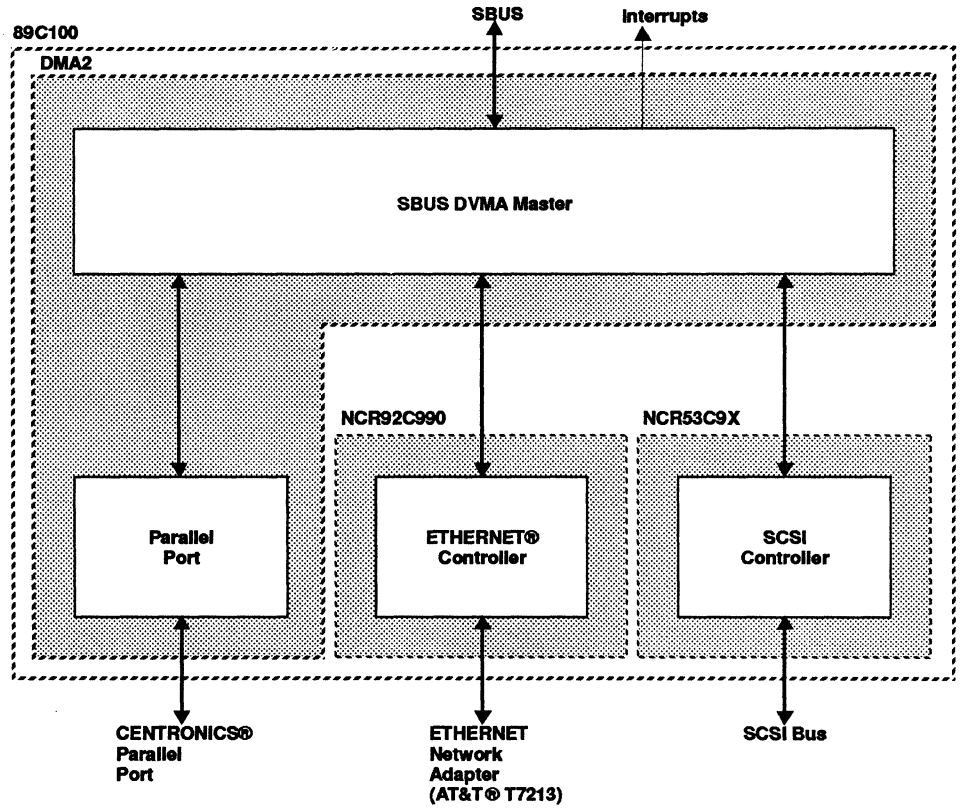


Figure 2-1 Chip-Level Functional Block Diagram

Features

- Single chip solution to standard SPARC DVMA devices - saves cost, power, board space, and weight. Together, with the 89C105 chip, forms a two-chip solution which provides the core SPARCstation I/O subsystem.
- Supports concurrent 10 MByte/sec SCSI transfers, 1.25 MByte/sec Ethernet transfers, and 4 MByte/sec Parallel Port transfers.
- Supports 4-word, 8-word, and 'no burst' SBus burst modes.
- 64-byte internal cache for Ethernet data buffering.
- 64-byte internal FIFOs for SCSI and Parallel Port data buffering.
- 16-bit write buffer for slave writes to Ethernet.
- Improved cache and FIFO draining algorithms for better SBus utilization.
- Internal address and byte count registers and "NEXT" address/byte count features for data block chaining on SCSI and Parallel Port interfaces.
- JTAG internal and boundary scan for improved chip and board level testability.

Intended Applications

The 89C100 is intended for low-end SBus-based systems in which cost, power, and area are the main design constraints. It is designed for use with either the Texas Instruments MicroSPARC or SuperSPARC processors, but will also work in any SBus-based system.

Related Products

The 89C100 is designed to share a single SBus slot with the NCR89C105.

Pinout Information

This section includes the pinout map and two tables that summarize the 89C100 pinout information in the following formats:

- Pinout by function
- Pinout by pin order on package
- JTAG boundary chain
- Pinout in TEST modes

Pinout Map

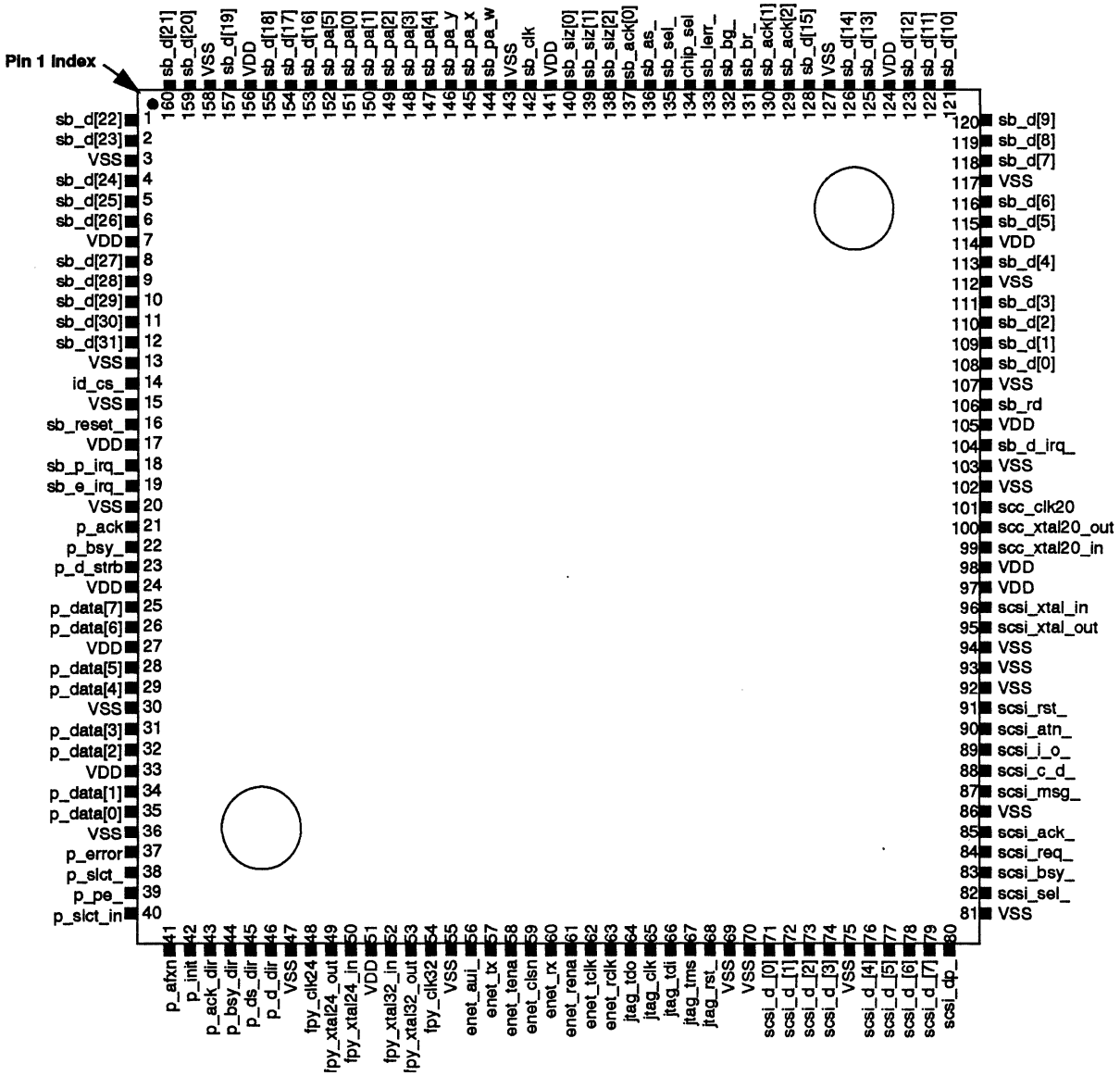


Figure 2-2 Pinout Map

Pinout Tables

Tables 2-4 and 2-5 present the 89C100 pinouts by function and by pin number sequence, respectively. The pin type entry in the Type column of Tables 2-4 and 2-5 is composed of fields which contain the mnemonic values shown in Tables 2-1 and 2-2:

Table 2-1 Valid Pin Mnemonics

Field	A	B	C	D	E	F	G	H	I
Value(s)	BS	IN	N	PD	2	L	U	25	T
	NCR	IO	—	—	4	—	D	100	H
	—	IOP			6		—	—	I
		ION			8				S16
		O			12				S18
		OT			16				S38
		IP			—				—

The values represented by the mnemonics in each of these fields is as follows:

Table 2-2 Mnemonic Descriptions

Field	Mnemonic	Description
A	BS	Boundary Scan
	NCR	NCR type
B	IN	Input
	IO	Bidirectional
	IOP	Bidirectional with pull-up/pull-down
	ION	Bidirectional open drain
	O	Output
	OT	Tristate output
	IP	Input with pull-up/pull-down
C	N	Open drain
D	PD	IO pad
E	*integer*	Pad drain in mA as indicated
F	L	Slew rate limited output
G	U	Pull-up
	D	Pull-down
H	*integer*	Pull-up/pull-down value in µA as indicated
I	T	TTL input receiver
	H	High drive TTL input receiver
	I	Inverting TTL input receiver
	S16	ds1216 Schmitt input receiver
	S18	ds1218 Schmitt input receiver
	S38	ds1238 Schmitt input receiver

For example, the pin type identification bsinpds18 means that the pin type is a boundary scan version of an input pad with a ds1218 Schmitt input receiver.

Notice that some fields in Table 2-1 are optional and that there are four exceptions to the above scheme. SCSIPAD is a 48 mA driver compatible with ANSI X3T9.2 requirements, SCSIPADF is the same with an input RC filter, OSC1401 is a crystal oscillator pad, and BSCLOCK is a clock input pad.

The Direction column in Tables 2-4 and 2-5 is used to identify the pin direction using the mnemonics shown in Table 2-3.

Table 2-3 Direction Mnemonic Descriptions

Mnemonic	Description
I	Input
O	Output
B	Bidirectional
T	Tristate
—	Not applicable

Note that the pad type listed in the following tables may not correspond exactly to the functional direction of the pin (input, output, bidirectional, or tristate) for either of the following reasons:

- The pin is used differently in a test mode. For instance, using an input as an output during test will require use of a bidirectional pad instead of an input.
- An equivalent output-only pad was not available. This applies specifically to the SBus outputs, which all use a custom 12 mA pad that was only available as a bidirectional pad.

Pinout by Function

Table 2-4 Pinout by Function

Name	Pin	Direction	Type	Description
SBus Interface: 59 pins				
sb_d[31]	12	B	BSIOPD12S16	SBus Data Bus (MSB)
sb_d[30]	11	B	BSIOPD12S16	SBus Data Bus
sb_d[29]	10	B	BSIOPD12S16	SBus Data Bus
sb_d[28]	9	B	BSIOPD12S16	SBus Data Bus
sb_d[27]	8	B	BSIOPD12S16	SBus Data Bus
sb_d[26]	6	B	BSIOPD12S16	SBus Data Bus
sb_d[25]	5	B	BSIOPD12S16	SBus Data Bus
sb_d[24]	4	B	BSIOPD12S16	SBus Data Bus
sb_d[23]	2	B	BSIOPD12S16	SBus Data Bus
sb_d[22]	1	B	BSIOPD12S16	SBus Data Bus
sb_d[21]	160	B	BSIOPD12S16	SBus Data Bus
sb_d[20]	159	B	BSIOPD12S16	SBus Data Bus
sb_d[19]	157	B	BSIOPD12S16	SBus Data Bus
sb_d[18]	155	B	BSIOPD12S16	SBus Data Bus
sb_d[17]	154	B	BSIOPD12S16	SBus Data Bus
sb_d[16]	153	B	BSIOPD12S16	SBus Data Bus
sb_d[15]	128	B	BSIOPD12S16	SBus Data Bus
sb_d[14]	126	B	BSIOPD12S16	SBus Data Bus
sb_d[13]	125	B	BSIOPD12S16	SBus Data Bus
sb_d[12]	123	B	BSIOPD12S16	SBus Data Bus
sb_d[11]	122	B	BSIOPD12S16	SBus Data Bus
sb_d[10]	121	B	BSIOPD12S16	SBus Data Bus
sb_d[9]	120	B	BSIOPD12S16	SBus Data Bus
sb_d[8]	119	B	BSIOPD12S16	SBus Data Bus
sb_d[7]	118	B	BSIOPD12S16	SBus Data Bus
sb_d[6]	116	B	BSIOPD12S16	SBus Data Bus
sb_d[5]	115	B	BSIOPD12S16	SBus Data Bus
sb_d[4]	113	B	BSIOPD12S16	SBus Data Bus
sb_d[3]	111	B	BSIOPD12S16	SBus Data Bus
sb_d[2]	110	B	BSIOPD12S16	SBus Data Bus
sb_d[1]	109	B	BSIOPD12S16	SBus Data Bus
sb_d[0]	108	B	BSIOPD12S16	SBus Data Bus (LSB)

Table 2-4 Pinout by Function (Continued)

Name	Pin	Direction	Type	Description
sb_br_	131	B	BSIOPD12T	SBus Bus Request
sb_bg_	132	I	BSINPDH	SBus Bus Grant
sb_ack[2]	129	B	BSIOPD12H	SBus Acknowledge
sb_ack[1]	130	B	BSIOPD12I	SBus Acknowledge
sb_ack[0]	137	B	BSIOPD12I	SBus Acknowledge
sb_reset_	16	I	BSINPDS16	SBus Reset
sb_lerr_	133	I	BSINPDT	SBus Late Error (INT15)
sb_clk	142	I	BSCLOCK	SBus Clock Input
sb_rd	106	B	BSIOPD12H	SBus Read/Write
sb_sel_	135	I	BSINPDH	SBus Select
sb_d_irq_	104	O	BSIOPD12U25T	SBus Interrupt for SCSI transfers (open-drain)
sb_e_irq_	19	O	BSIOPD12U25T	SBus Interrupt for ETHERNET transfers (open-drain)
sb_p_irq_	18	0	BSIOPD12U25T	SBus Interrupt for Parallel Port Transfers (open-drain)
sb_siz[2]	138	B	BSIOPD12H	SBus Transfer Size
sb_siz[1]	139	B	BSIOPD12H	SBus Transfer Size
sb_siz[0]	140	B	BSIOPD12H	SBus Transfer Size
sb_as_	136	I	BSINPDH	SBus Address Strobe (address is valid)
chip_sel[1]	134	I	BSINPDT	High order physical address bit
sb_pa[w]	144	I	BSINPDH	High order physical address bit
sb_pa[x]	145	I	BSIONPD4H	High order physical address bit
sb_pa[y]	146	I	BSINPDH	High order physical address bit
sb_pa[5]	152	I	BSINPDH	Low order physical address bit
sb_pa[4]	147	I	BSINPDH	Low order physical address bit
sb_pa[3]	148	I	BSINPDH	Low order physical address bit
sb_pa[2]	149	I	BSINPDH	Low order physical address bit
sb_pa[1]	150	I	BSINPDH	Low order physical address bit

Table 2-4 Pinout by Function (Continued)

Name	Pin	Direction	Type	Description
sb_pa[0]	151	I	BSINPDT	Low order physical address bit
Ethernet Interface: 8 pins				
enet_aui_[2]	56	O	BSOTPD4	Ethernet TP/AUI_ select
enet_tx	57	O	BSOTPD4	Ethernet Transmit data
enet_tena	58	O	BSOTPD4	Ethernet Transmit enable
enet_clsn	59	I	BSINPDT	Ethernet Collision detect
enet_rx	60	I	BSINPDT	Ethernet Receive data
enet_rena	61	I	BSINPDT	Ethernet Receiver enable (carrier sense)
enet_tclk	62	I	BSINPDT	Ethernet Transmit clock
enet_rclk	63	I	BSINPDT	Ethernet Receive clock
SCSI Interface[3]: 20 pins				
scsi_d_[7]	79	B	SCSIPAD	SCSI Data
scsi_d_[6]	78	B	SCSIPAD	SCSI Data
scsi_d_[5]	77	B	SCSIPAD	SCSI Data
scsi_d_[4]	76	B	SCSIPAD	SCSI Data
scsi_d_[3]	74	B	SCSIPAD	SCSI Data
scsi_d_[2]	73	B	SCSIPAD	SCSI Data
scsi_d_[1]	72	B	SCSIPAD	SCSI Data
scsi_d_[0]	71	B	SCSIPAD	SCSI Data
scsi_dp_	80	B	SCSIPAD	SCSI Data Parity
scsi_sel_	82	B	SCSIPAD	SCSI Select
scsi_bsy_	83	B	SCSIPAD	SCSI Busy
scsi_req_	84	B	SCSIPADF	SCSI Request
scsi_ack_	85	B	SCSIPADF	SCSI Acknowledge
scsi_msg_	87	B	SCSIPAD	SCSI Message
scsi_c_d_	88	B	SCSIPAD	SCSI Command/Data
scsi_i_o_	89	B	SCSIPAD	SCSI Input/Output
scsi_atn_	90	B	SCSIPADF	SCSI Attention
scsi_rst_	91	B	SCSIPAD	SCSI Reset
scsi_xtal_in	96	I	OSC1401	SCSI Clock Crystal In (can drive with external CMOS clock)

Table 2-4 Pinout by Function (Continued)

Name	Pin	Direction	Type	Description
scsi_xtal_out	95	O	BSOSC1401	SCSI Clock Crystal Out (must not connect to any external load)
Parallel Port Interface: 22 Pins				
p_data[7]	25	T	BSIOPD4T	Parallel Port Data Bus
p_data[6]	26	T	BSIOPD4T	Parallel Port Data Bus
p_data[5]	28	T	BSIOPD4T	Parallel Port Data Bus
p_data[4]	29	T	BSIOPD4T	Parallel Port Data Bus
p_data[3]	31	T	BSIOPD4T	Parallel Port Data Bus
p_data[2]	32	T	BSIOPD4T	Parallel Port Data Bus
p_data[1]	34	T	BSIOPD4T	Parallel Port Data Bus
p_data[0]	35	T	BSIOPD4T	Parallel Port Data Bus
p_d_strb	23	B	BSIOPPD4D25T	Parallel Port Data Strobe (25 uA pull-down)
p_bsy_	22	B	BSIOPPD4U25T	Parallel Port Busy (25 uA pull-up)
p_ack	21	B	BSIOPPD4D25T	Parallel Port Acknowledge (25 uA pull-down)
p_pe_	39	B	BSIONPD4T	Parallel Port Paper Error
p_slct_	38	B	BSIONPD4T	Parallel Port Select
p_error	37	I	BSINPDT	Parallel Port Error
p_init	42	O	BSOTPD4	Parallel Port Initialize
p_slct_in	40	O	BSOTPD4	Parallel Port Select In
p_afxn	41	O	BSOTPD4	Parallel Port Auto Feed
p_ds_dir[4]	45	O	BSOTPD4	Parallel Port Data Strobe Direction
p_bsy_dir[4]	44	O	BSOTPD4	Parallel Port Busy Direction
p_ack_dir[4]	43	O	BSOTPD4	Parallel Port Acknowledge Direction
p_d_dir[4]	46	O	BSOTPD4	Parallel Port Data Direction

Table 2-4 Pinout by Function (Continued)

Name	Pin	Direction	Type	Description
id_cs_	14	O	BSIOPPD4U25T	Secondary Device Select (boot prom) output; pull low to specify absence of external prom
Test: 5 pins				
jtag_tdo	64	O	NCROTPD4	JTAG Test Data Output
jtag_tdi	66	I	NCRIPPDU100	JTAG Test Data Input (100 uA pull-up)
jtag_clk	65	I	NCRINPD	JTAG Clock
jtag_tms	67	I	NCRIPPDU100	JTAG Test Mode Select (100 uA pull-up)
jtag_rst_	68	I	NCRIPPDU100	JTAG Reset (100 uA pull-up)
General Purpose Oscillators[5]: 9 pins				
scc_xtal20_in	99	I	OSC1401	SCC Clock Crystal In (19.66 MHz) (can drive with external CMOS clock)
scc_xtal20_out	100	O	OSC1401	SCC Clock Crystal Out (19.66 MHz) (must not connect to any external load)
scc_clk20	101	O	OPD16SYM	SCC Clock Out (19.66 MHz)
fpy_xtal24_in	50	I	OSC1401	Floppy Clock Crystal In (24 MHz) (can drive with external CMOS clock)
fpy_xtal24_out	49	O	OSC1401	Floppy Clock Crystal Out (24 MHz) (must not connect to any external load)
fpy_clk24	48	O	OPD16SYM	Floppy Clock Out (24 MHz)
fpy_xtal32_in	52	I	OSC1401	Floppy Clock Crystal In (32 MHz) (can drive with external CMOS clock)
fpy_xtal32_out	53	O	OSC1401	Floppy Clock Crystal Out (32 MHz) (must not connect to any external load)
fpy_clk32	54	O	OPD16SYM	Floppy Clock Out (32 MHz)
Power, Ground: 37 pins				

Table 2-4 Pinout by Function (Continued)

Name	Pin	Direction	Type	Description
VDD	7	—		Power Connection
VDD	17	—		Power Connection
VDD	24	—		Power Connection
VDD	27	—		Power Connection
VDD	33	—		Power Connection
VDD	51	—		Power Connection
VDD	97	—		Power Connection
VDD	98	—		Power Connection
VDD	105	—		Power Connection
VDD	114	—		Power Connection
VDD	124	—		Power Connection
VDD	141	—		Power Connection
VDD	156	—		Power Connection
VSS	3	—		Ground Connection
VSS	13	—		Ground Connection
VSS	15	—		Ground Connection
VSS	20	—		Ground Connection
VSS	30	—		Ground Connection
VSS	36	—		Ground Connection
VSS	47	—		Ground Connection
VSS	55	—		Ground Connection
VSS	69	—		Ground Connection
VSS	70	—		Ground Connection
VSS	75	—		Ground Connection
VSS	81	—		Ground Connection
VSS	86	—		Ground Connection
VSS	92	—		Ground Connection
VSS	93	—		Ground Connection
VSS	94	—		Ground Connection
VSS	102	—		Ground Connection
VSS	103	—		Ground Connection
VSS	107	—		Ground Connection
VSS	112	—		Ground Connection
VSS	117	—		Ground Connection

Table 2-4 Pinout by Function (Continued)

Name	Pin	Direction	Type	Description
VSS	127	—		Ground Connection
VSS	143	—		Ground Connection
VSS	158	—		Ground Connection

1. The chip_sel pin is an additional qualifier (active high) to the sb_sel_ line. In some system configurations where the 89C100 and the 89C105 share a single SBus select line, PA[27] can be used to select between the two.
2. Drives MIS input of the AT&T T7213 chip to select between twisted pair and AUI-type Ethernet interfaces, with ENET_AUI_ = 0 selecting AUI.
3. All of the SCSI pads (except the crystal oscillator pads) are custom NCR 48 mA bidirectional open-drain pads with hysteresis on inputs.
4. The Parallel Port control and data line direction bits, (for example, p_*_dir), are gang programmed by the DIR bit of the Transfer Control Register. DIR=0 sets transfer direction away from the 89C100 (p_d_dir=p_ds_dir=1; p_bsy_dir=p_ack_dir=0); DIR=1 sets transfer direction towards the 89C100 (p_d_dir=p_ds_dir=0; p_bsy_dir=p_ack_dir=1).
5. In some system configurations, the 89C100 provides these three clocks to the 89C105 (which is pin limited). These are really general-purpose 20-50 MHz crystal oscillator pads that can operate in both fundamental and overtone mode. Refer to page 60"OSC1401 Crystal Oscillator" for more information.

Pinout by Pin Number Sequence

Table 2-5 Pinout by Pin Number Sequence

Pin	Name	Direction	Type	Description
1	sb_d[22]	B	BSIOPD12S16	SBus Data Bus
2	sb_d[23]	B	BSIOPD12S16	SBus Data Bus
3	VSS	—		Ground Connection
4	sb_d[24]	B	BSIOPD12S16	SBus Data Bus
5	sb_d[25]	B	BSIOPD12S16	SBus Data Bus
6	sb_d[26]	B	BSIOPD12S16	SBus Data Bus
7	VDD	—		Power Connection
8	sb_d[27]	B	BSIOPD12S16	SBus Data Bus
9	sb_d[28]	B	BSIOPD12S16	SBus Data Bus
10	sb_d[29]	B	BSIOPD12S16	SBus Data Bus
11	sb_d[30]	B	BSIOPD12S16	SBus Data Bus
12	sb_d[31]	B	BSIOPD12S16	SBus Data Bus (MSB)
13	VSS	—		Ground Connection
14	id_cs_	O	BSIOPPD4U25T	Secondary Device Select (boot prom) output; pull low to specify absence of external prom
15	VSS	—		Ground Connection
16	sb_reset_	I	BSINPDS16	SBus Reset
17	VDD	—		Power Connection
18	sb_p_irq_	O	BSIOPD12U25T	SBus Interrupt for Parallel Port Transfers (open-drain)
19	sb_e_irq_	O	BSIOPD12U25T	SBus Interrupt for Ethernet transfers (open-drain)
20	VSS	—		Ground Connection
21	p_ack	B	BSIOPPD4D25T	Parallel Port Acknowledge (25 µA pull-down)
22	p_bsy_	B	BSIOPPD4U25T	Parallel Port Busy (25 µA pull-up)
23	p_d_strb	B	BSIOPPD4D25T	Parallel Port Data Strobe (25 µA pull-down)
24	VDD	—		Power Connection
25	p_data[7]	T	BSIOPD4T	Parallel Port Data Bus
26	p_data[6]	T	BSIOPD4T	Parallel Port Data Bus
27	VDD	—		Power Connection

Table 2-5 Pinout by Pin Number Sequence (Continued)

Pin	Name	Direction	Type	Description
28	p_data[5]	T	BSIOPD4T	Parallel Port Data Bus
29	p_data[4]	T	BSIOPD4T	Parallel Port Data Bus
30	VSS	—		Ground Connection
31	p_data[3]	T	BSIOPD4T	Parallel Port Data Bus
32	p_data[2]	T	BSIOPD4T	Parallel Port Data Bus
33	VDD	—		Power Connection
34	p_data[1]	T	BSIOPD4T	Parallel Port Data Bus
35	p_data[0]	T	BSIOPD4T	Parallel Port Data Bus
36	VSS	—		Ground Connection
37	p_error	I	BSINPDT	Parallel Port Error
38	p_slct_	B	BSIONPD4T	Parallel Port Select
39	p_pe_	B	BSIONPD4T	Parallel Port Paper Error
40	p_slct_in	O	BSOTPD4	Parallel Port Select In
41	p_afxn	O	BSOTPD4	Parallel Port Auto Feed
42	p_init	O	BSOTPD4	Parallel Port Initialize
43	p_ack_dir[1]	O	BSOTPD4	Parallel Port Acknowledge Direction
44	p_bsy_dir[1]	O	BSOTPD4	Parallel Port Busy Direction
45	p_ds_dir[1]	O	BSOTPD4	Parallel Port Data Strobe Direction
46	p_d_dir[1]	O	BSOTPD4	Parallel Port Data Direction
47	VSS	—		Ground Connection
48	fpy_clk24[2]	O	OPD16SYM	Floppy Clock Out (24 MHz)
49	fpy_xtal24_out[2]	O	OSC1401	Floppy Clock Crystal Out (24 MHz) (must not connect to any external load)
50	fpy_xtal24_in[2]	I	OSC1401	Floppy Clock Crystal In (24 MHz) (can drive with external CMOS clock)
51	VDD	—		Power Connection
52	fpy_xtal32_in[2]	I	OSC1401	Floppy Clock Crystal In (32 MHz) (can drive with external CMOS clock)
53	fpy_xtal32_out[2]	O	OSC1401	Floppy Clock Crystal Out (32 MHz) (must not connect to any external load)
54	fpy_clk32[2]	O	OPD16SYM	Floppy Clock Out (32 MHz)

Table 2-5 Pinout by Pin Number Sequence (Continued)

Pin	Name	Direction	Type	Description
55	VSS	—		Ground Connection
56	enet_aui_[3]	O	BSOTPD4	Ethernet TP/AUI_ select output
57	enet_tx	O	BSOTPD4	Ethernet Transmit data output
58	enet_tena	O	BSOTPD4	Ethernet Transmit enable output
59	enet_clsn	I	BSINPDT	Ethernet Collision detect input
60	enet_rx	I	BSINPDT	Ethernet Receive data input
61	enet_rena	I	BSINPDT	Ethernet Receiver enable (carrier sense) input
62	enet_tclk	I	BSINPDT	Ethernet Transmit clock input
63	enet_rclk	I	BSINPDT	Ethernet Receive clock input
64	jtag_tdo	O	NCROTPD4	JTAG Test Data Output
65	jtag_clk	I	NCRINPD	JTAG Clock
66	jtag_tdi	I	NCRIPPDU100	JTAG Test Data Input (100 µA pull-up)
67	jtag_tms	I	NCRIPPDU100	JTAG Test Mode Select (100 µA pull-up)
68	jtag_rst_	I	NCRIPPDU100	JTAG Reset (100 µA pull-up)
69	VSS	—		Ground Connection
70	VSS	—		Ground Connection
71	scsi_d_[0][4]	B	SCSIPAD	SCSI Data
72	scsi_d_[1][4]	B	SCSIPAD	SCSI Data
73	scsi_d_[2][4]	B	SCSIPAD	SCSI Data
74	scsi_d_[3][4]	B	SCSIPAD	SCSI Data
75	VSS	—		Ground Connection
76	scsi_d_[4][4]	B	SCSIPAD	SCSI Data
77	scsi_d_[5][4]	B	SCSIPAD	SCSI Data
78	scsi_d_[6][4]	B	SCSIPAD	SCSI Data
79	scsi_d_[7][4]	B	SCSIPAD	SCSI Data
80	scsi_dp_[4]	B	SCSIPAD	SCSI Data Parity
81	VSS	—		Ground Connection
82	scsi_sel_[4]	B	SCSIPAD	SCSI Select
83	scsi_bsy_[4]	B	SCSIPAD	SCSI Busy

Table 2-5 Pinout by Pin Number Sequence (Continued)

Pin	Name	Direction	Type	Description
84	scsi_req_[4]	B	SCSIPADF	SCSI Request
85	scsi_ack_[4]	B	SCSIPADF	SCSI Acknowledge
86	VSS	—		Ground Connection
87	scsi_msg_[4]	B	SCSIPAD	SCSI Message
88	scsi_c_d_[4]	B	SCSIPAD	SCSI Command/Data
89	scsi_i_o_[4]	B	SCSIPAD	SCSI Input/Output
90	scsi_atn_[4]	B	SCSIPADF	SCSI Attention
91	scsi_rst_[4]	B	SCSIPAD	SCSI Reset
92	VSS	—		Ground Connection
93	VSS	—		Ground Connection
94	VSS	—		Ground Connection
95	scsi_xtal_out	O	OSC1401	SCSI Clock Crystal Out (must not connect to any external load)
96	scsi_xtal_in	I	BSOSC1401	SCSI Clock Crystal In (can drive with external CMOS clock)
97	VDD	—		Power Connection
98	VDD	—		Power Connection
99	scc_xtal20_in[2]	I	OSC1401	SCC Clock Crystal In (19.66 MHz) (can drive with external CMOS clock)
100	scc_xtal20_out[2]	O	OSC1401	SCC Clock Crystal Out (19.66 MHz) (must not connect to any external load)
101	scc_clk20[2]	O	OPD16SYM	SCC Clock Out (19.66 MHz)
102	VSS	—		Ground Connection
103	VSS	—		Ground Connection
104	sb_d_irq_	O	BSIOPD12U25T	SBus Interrupt for SCSI transfers (open-drain)
105	VDD	—		Power Connection
106	sb_rd	T	BSIOPD12H	SBus Read/Write
107	VSS	—		Ground Connection
108	sb_d[0]	B	BSIOPD12S16	SBus Data Bus (LSB)
109	sb_d[1]	B	BSIOPD12S16	SBus Data Bus
110	sb_d[2]	B	BSIOPD12S16	SBus Data Bus
111	sb_d[3]	B	BSIOPD12S16	SBus Data Bus

Table 2-5 Pinout by Pin Number Sequence (Continued)

Pin	Name	Direction	Type	Description
112	VSS	—		Ground Connection
113	sb_d[4]	B	BSIOPD12S16	SBus Data Bus
114	VDD	—		Power Connection
115	sb_d[5]	B	BSIOPD12S16	SBus Data Bus
116	sb_d[6]	B	BSIOPD12S16	SBus Data Bus
117	VSS	—		Ground Connection
118	sb_d[7]	B	BSIOPD12S16	SBus Data Bus
119	sb_d[8]	B	BSIOPD12S16	SBus Data Bus
120	sb_d[9]	B	BSIOPD12S16	SBus Data Bus
121	sb_d[10]	B	BSIOPD12S16	SBus Data Bus
122	sb_d[11]	B	BSIOPD12S16	SBus Data Bus
123	sb_d[12]	B	BSIOPD12S16	SBus Data Bus
124	VDD	—		Power Connection
125	sb_d[13]	B	BSIOPD12S16	SBus Data Bus
126	sb_d[14]	B	BSIOPD12S16	SBus Data Bus
127	VSS	—		Ground Connection
128	sb_d[15]	B	BSIOPD12S16	SBus Data Bus
129	sb_ack[2]	B	BSIOPD12H	SBus Acknowledge
130	sb_ack[1]	B	BSIOPD12I	SBus Acknowledge
131	sb_br_	B	BSIOPD12T	SBus Bus Request
132	sb_bg_	I	BSINPDH	SBus Bus Grant
133	sb_lerr_	I	BSINPDT	SBus Late Error (INT15)
134	chip_sel[5]	I	BSINPDT	High order physical address bits (for slave decodes)
135	sb_sel_	I	BSINPDH	SBus Select
136	sb_as_	I	BSINPDH	SBus Address Strobe (address is valid)
137	sb_ack[0]	B	BSIOPD12I	SBus Acknowledge
138	sb_siz[2]	B	BSIOPD12H	SBus Transfer Size
139	sb_siz[1]	B	BSIOPD12H	SBus Transfer Size
140	sb_siz[0]	B	BSIOPD12H	SBus Transfer Size
141	VDD	—		Power Connection
142	sb_clk	I	BSINPDH	SBus Clock Input
143	VSS	—		Ground Connection

Table 2-5 Pinout by Pin Number Sequence (Continued)

Pin	Name	Direction	Type	Description
144	sb_pa_w	I	BSINPDH	High order physical address bit
145	sb_pa_x	I	BSIONPD4H	High order physical address bit
146	sb_pa_y	I	BSINPDH	High order physical address bit
147	sb_pa[4]	I	BSINPDH	Low order physical address bit
148	sb_pa[3]	I	BSINPDH	Low order physical address bit
149	sb_pa[2]	I	BSINPDH	Low order physical address bit
150	sb_pa[1]	I	BSINPDH	Low order physical address bit
151	sb_pa[0]	I	BSINPDT	Low order physical address bit
152	sb_pa[5]	I	BSINPDT	Low order physical address bit
153	sb_d[16]	B	BSIOPD12S16	SBus Data Bus
154	sb_d[17]	B	BSIOPD12S16	SBus Data Bus
155	sb_d[18]	B	BSIOPD12S16	SBus Data Bus
156	VDD	—		Power Connection
157	sb_d[19]	B	BSIOPD12S16	SBus Data Bus
158	VSS	—		Ground Connection
159	sb_d[20]	B	BSIOPD12S16	SBus Data Bus
160	sb_d[21]	B	BSIOPD12S16	SBus Data Bus

1. The Parallel Port control and data line direction bits, (for example, - p_*_dir), are gang programmed by the DIR bit of the Transfer Control Register. DIR=0 sets transfer direction away from the 89C100 (p_d_dir=p_ds_dir=1; p_bsy_dir=p_ack_dir=0); DIR=1 sets transfer direction towards the 89C100 (p_d_dir=p_ds_dir=0; p_bsy_dir=p_ack_dir=1).
2. In some system configurations, the 89C100 provides these three clocks to the 89C105 (which is pin limited). These are really general-purpose 20-50 MHz crystal oscillator pads that can operate in both fundamental and overtone mode. Refer to page 60, "OSC1401 Crystal Oscillator" for more information.
3. Drives MIS input of the AT&T T7213 chip to select between twisted pair and AUI-type Ethernet interfaces, with ENET_AUI_ = 0 selecting AUI.
4. All of the SCSI pads are custom NCR 48 mA bidirectional open-drain pads with hysteresis on inputs.
5. The chip_sel pin is an additional qualifier (active high) to the sb_sel_ line. In some system configurations, where the 89C100 and the 89C105 share a single SBus select line, PA[27] is used to select between the two, with PA[27]=1 selecting the 89C100.

JTAG Boundary Information

Table 2-6 describes the boundary scan chain. The numbers listed in the Input, Output, and Enable columns represent the bit order of the scan chain. Bit 0 is the first to be scanned in. All of the enable signals are active low.

Table 2-6 Boundary Chain Description

Pin	Name	Type	Input	Output	Enable
1	sb_d[22]	BIDIR	87	88	97
2	sb_d[23]	BIDIR	89	90	97
3	io_vss	POWER	—	—	—
4	sb_d[24]	BIDIR	91	92	97
5	sb_d[25]	BIDIR	93	94	97
6	sb_d[26]	BIDIR	95	96	97
7	io_vdd	POWER	—	—	—
8	sb_d[27]	BIDIR	98	99	97
9	sb_d[28]	BIDIR	100	101	97
10	sb_d[29]	BIDIR	102	103	97
11	sb_d[30]	BIDIR	104	105	97
12	sb_d[31]	BIDIR	106	107	97
13	io_vss	POWER	—	—	—
14	id_cs_	BIDIR	109	110	108
15	core_vss	POWER	—	—	—
16	sb_reset_	INPUT	111	—	—
17	core_vdd	POWER	—	—	—
18	sb_p_irq_	BIDIR	113	114	112
19	sb_e_irq_	BIDIR	116	117	115
20	core_vss	POWER	—	—	—
21	p_ack	BIDIR	120	121	118
22	p_bsy_	BIDIR	123	124	122
23	p_d_strb	BIDIR	126	127	125
24	core_vdd	POWER	—	—	—
25	p_data[7]	BIDIR	128	129	136
26	p_data[6]	BIDIR	130	131	136
27	io_vdd	POWER	—	—	—
28	p_data[5]	BIDIR	132	133	136
29	p_data[4]	BIDIR	134	135	136
30	core_vss	POWER	—	—	—

Table 2-6 Boundary Chain Description (Continued)

Pin	Name	Type	Input	Output	Enable
31	p_data[3]	BIDIR	137	138	136
32	p_data[2]	BIDIR	139	140	136
33	core_vdd	POWER	—	—	—
34	p_data[1]	BIDIR	142	143	136
35	p_data[0]	BIDIR	144	145	136
36	io_vss	POWER	—	—	—
37	p_error	INPUT	146	—	—
38	p_slct_	BIDIR	147	148	119
39	p_pe_	BIDIR	149	150	119
40	p_slct_in	TRISTATE	—	152	151
41	p_afxn	TRISTATE	—	153	155
42	p_init	TRISTATE	—	154	155
43	p_ack_dir	TRISTATE	—	156	155
44	p_bsy_dir	TRISTATE	—	158	157
45	p_ds_dir	TRISTATE	—	159	141
46	p_d_dir	TRISTATE	—	160	155
47	io_vss	POWER	—	—	—
48	fpy_clk24	OUTPUT	—	—	—
49	fpy_xtal24_out	OUTPUT	—	—	—
50	fpy_xtal24_in	INPUT	—	—	—
51	io_vdd	POWER	—	—	—
52	fpy_xtal32_in	INPUT	—	—	—
53	fpy_xtal32_out	OUTPUT	—	—	—
54	fpy_clk32	OUTPUT	—	—	—
55	io_vss	POWER	—	—	—
56	enet_aui_	TRISTATE	—	162	161
57	enet_tx	TRISTATE	—	163	141
58	enet_tena	TRISTATE	—	164	141
59	enet_clsn	INPUT	165	—	—
60	enet_rx	INPUT	166	—	—
61	enet_rena	INPUT	167	—	—
62	enet_tclk	INPUT	168	—	—
63	enet_rclk	INPUT	169	—	—
64	jtag_tdo	TDO	—	—	—

Table 2-6 Boundary Chain Description (Continued)

Pin	Name	Type	Input	Output	Enable
65	jtag_clk	TCK	—	—	—
66	jtag_tdi	TDI	—	—	—
67	jtag_tms	TMS	—	—	—
68	jtag_rst_	TRSTB	—	—	—
69	io_vss	POWER	—	—	—
70	scsipad_gnd	POWER	—	—	—
71	scsi_d[0]	BIDIR	170	171	188
72	scsi_d[1]	BIDIR	172	173	188
73	scsi_d[2]	BIDIR	174	175	188
74	scsi_d[3]	BIDIR	176	177	188
75	scsipad_gnd	POWER	—	—	—
76	scsi_d[4]	BIDIR	178	179	188
77	scsi_d[5]	BIDIR	180	181	188
78	scsi_d[6]	BIDIR	182	183	188
79	scsi_d[7]	BIDIR	184	185	188
80	scsi_dp_	BIDIR	186	187	188
81	scsipad_gnd	POWER	—	—	—
82	scsi_sel_	BIDIR	194	195	208
83	scsi_bsy_	BIDIR	196	197	208
84	scsi_req_	BIDIR	189	190	193
85	scsi_ack_	BIDIR	191	192	193
86	scsipad_gnd	POWER	—	—	—
87	scsi_msg_	BIDIR	198	199	208
88	scsi_c_d_	BIDIR	200	201	208
89	scsi_i_o_	BIDIR	202	203	208
90	scsi_atn_	BIDIR	204	205	208
91	scsi_rst_	BIDIR	206	207	208
92	scsipad_gnd	POWER	—	—	—
93	core_vss	POWER	—	—	—
94	io_vss	POWER	—	—	—
95	scsi_xtal_out	OUTPUT	—	—	—
96	scsi_xtal_in	INPUT	0	—	—
97	io_vdd	POWER	—	—	—
98	core_vdd	POWER	—	—	—

Table 2-6 Boundary Chain Description (Continued)

Pin	Name	Type	Input	Output	Enable
99	scc_xtal20_in	INPUT	—	—	—
100	scc_xtal20_out	OUTPUT	—	—	—
101	scc_clk20	OUTPUT	—	—	—
102	io_vss	POWER	—	—	—
103	core_vss	POWER	—	—	—
104	sb_d_irq_	BIDIR	2	3	1
105	core_vdd	POWER	—	—	—
106	sb_rd	BIDIR	5	6	4
107	io_vss	POWER	—	—	—
108	sb_d[0]	BIDIR	7	8	15
109	sb_d_[1]	BIDIR	9	10	15
110	sb_d[2]	BIDIR	11	12	15
111	sb_d[3]	BIDIR	13	14	15
112	core_vss	POWER	—	—	—
113	sb_d[4]	BIDIR	16	17	15
114	io_vdd	POWER	—	—	—
115	sb_d[5]	BIDIR	18	19	15
116	sb_d[6]	BIDIR	20	21	15
117	io_vss	POWER	—	—	—
118	sb_d[7]	BIDIR	22	23	30
119	sb_d[8]	BIDIR	24	25	30
120	sb_d[9]	BIDIR	26	27	30
121	sb_d[10]	BIDIR	28	29	30
122	sb_d[11]	BIDIR	31	32	30
123	sb_d[12]	BIDIR	33	34	30
124	io_vdd	POWER	—	—	—
125	sb_d[13]	BIDIR	35	36	30
126	sb_d[14]	BIDIR	37	38	70
127	io_vss	POWER	—	—	—
128	sb_d[15]	BIDIR	39	40	70
129	sb_ack[2]	BIDIR	41	42	43
130	sb_ack[1]	BIDIR	44	45	43
131	sb_br_	BIDIR	46	47	141
132	sb_bg_	INPUT	48	—	—

Table 2-6 Boundary Chain Description (Continued)

Pin	Name	Type	Input	Output	Enable
133	sb_lerr_	INPUT	49	—	—
134	chip_sel	INPUT	50	—	—
135	sb_sel_	INPUT	51	—	—
136	sb_as_	INPUT	52	—	—
137	sb_ack[0]	BIDIR	54	55	53
138	sb_size[2]	BIDIR	56	57	58
139	sb_size[1]	BIDIR	59	60	58
140	sb_size[0]	BIDIR	61	62	58
141	io_vdd	POWER	—	—	—
142	sb_clk	INPUT	63	—	—
143	io_vss	POWER	—	—	—
144	sb_pa_w	INPUT	64	—	—
145	sb_pa_x	BIDIR	65	66	119
146	sb_p_y	INPUT	67	—	—
147	sb_pa[4]	INPUT	68	—	—
148	sb_pa[3]	INPUT	69	—	—
149	sb_pa[2]	INPUT	71	—	—
150	sb_pa[1]	INPUT	72	—	—
151	sb_pa[0]	INPUT	73	—	—
152	sb_pa[5]	INPUT	74	—	—
153	sb_d[16]	BIDIR	75	76	70
154	sb_d[17]	BIDIR	77	78	70
155	sb_d[18]	BIDIR	79	80	70
156	io_vdd	POWER	—	—	—
157	sb_d[19]	BIDIR	81	82	70
158	io_vss	POWER	—	—	—
159	sb_d[20]	BIDIR	83	84	70
160	sb_d[21]	BIDIR	85	86	70

Functional Operation

This section includes the following:

- Detailed chip block diagram
- Chip-level address map
- Functional chip description

Detailed Chip Block Diagram

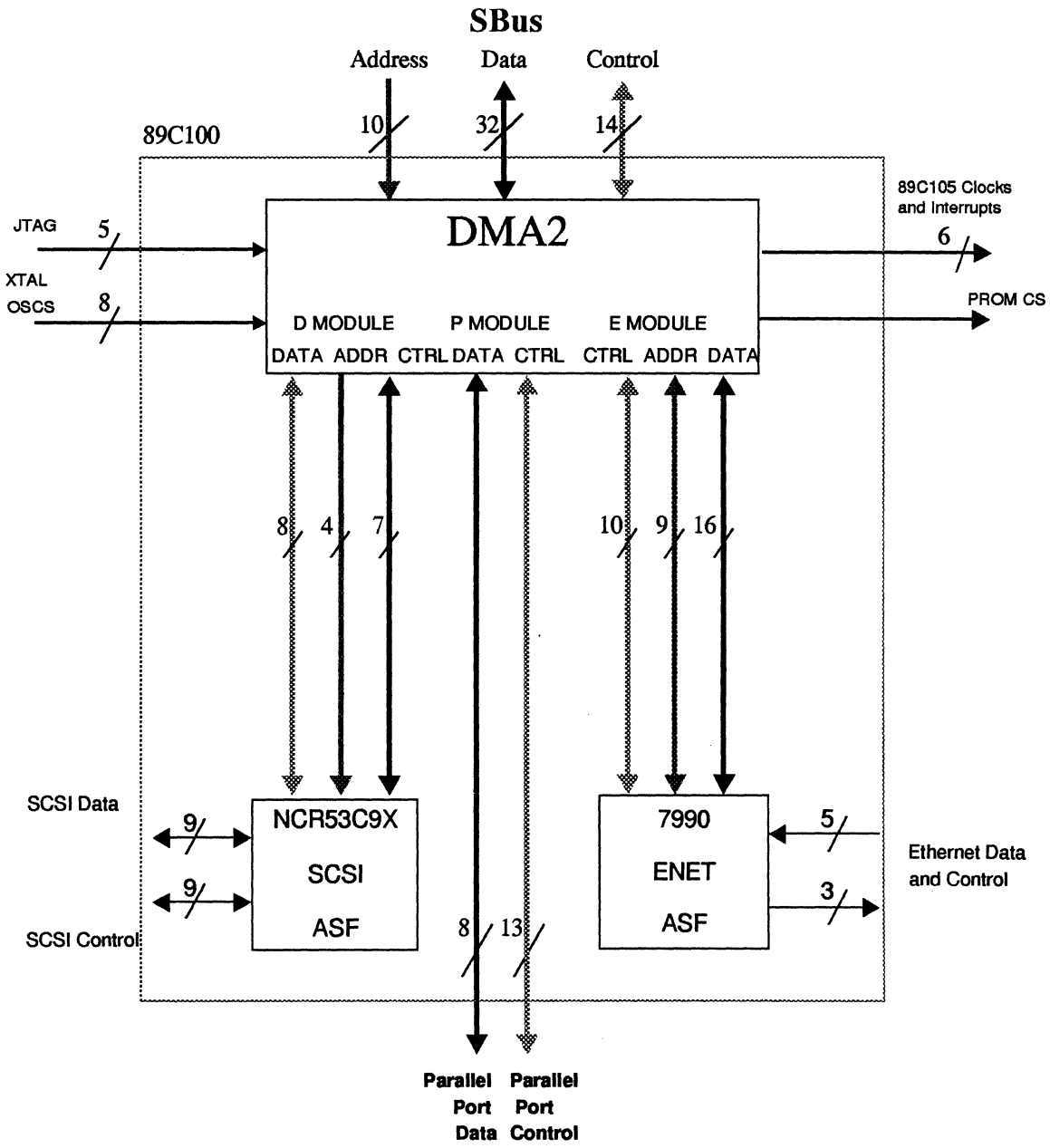


Figure 2-3 Chip Block Diagram

Chip-Level Address Map

Table 2-7 Chip-Level Address Map

sb_pa(27:0)[1] [2]	Register Accesses	Type	Size
0x800 0000	**DMA2 Internal ID Register[3]**	R	32
0x840 0000 -> 0x840 000f	**DMA2 ESP Registers**		
0x840 0000	Control/Status Register (D_CSR)	R/W	32
0x840 0004	Address Register (D_ADDR)[4]	R/W	32
0x840 0008	Byte Count Register (D_BCNT)[4]	R/W	24
0x840 000c	Test Control/Status Reg (D_TST_CSR)	R/W	32
0x840 0010 -> 0x840 001f	**DMA2 Ethernet Registers**		
0x840 0010	Control/Status Register (E_CSR)	R/W	32
0x840 0014	Test Control/Status Register (E_TST_CSR)	R/W	32
0x840 0018	Cache Valid Bits (E_VLD)	R/W	32
0x840 001c	Base Address Reg (E_BASE_ADDR)	R/W	8
0xc80 0000 -> 0xc80 001f	**DMA2 Parallel Port Registers**		
0xc80 0000	Control/Status Register (P_CSR)	R/W	32
0xc80 0004	Address Register (P_ADDR)[5]	R/W	32
0xc80 0008	Byte Count Register (P_BCNT)[5]	R/W	32
0xc80 000c	Test Control/Status Register (P_TST_CSR)	R/W	32
0xc80 0010	Hardware Configuration Reg (P_HCR)	R/W	16
0xc80 0012	Operation Configuration Reg (P_OCR)	R/W	16
0xc80 0014	Parallel Data Register (P_DR)	R/W	8
0xc80 0015	Transfer Control Register (P_TCR)	R/W	8
0xc80 0016	Output Register (P_OR)	R/W	8
0xc80 0017	Input Register (P_IR)	R/W	8
0xc80 0018	Interrupt Control Register (P_ICR)	R/W	16

Table 2-7 Chip-Level Address Map (Continued)

sb_pa(27:0)[1] [2]	Register Accesses	Type	Size
0x880 0000 - > 0x880 003f	**SCSI Controller Registers**		
0x880 0000	Transfer Count Low (7:0)	R/W	8
0x880 0004	Transfer Count Middle (15:8)	R/W	8
0x880 0008	FIFO Data	R/W	8
0x880 000c	Command	R/W	8
0x880 0010	Status	R	8
0x880 0010	Select-Reselect Bus ID	W	8
0x880 0014	Interrupt	R	8
0x880 0014	Select-Reselect Time-Out	W	8
0x880 0018	Sequence Step	R	8
0x880 0018	Synchronous Transfer Period	W	8
0x880 001c	FIFO Flags	R	8
0x880 001c	Synchronous Offset	W	8
0x880 0020	Configuration #1	R/W	8
0x880 0024	Clock Conversion Factor	W	8
0x880 0028	Test (Chip Test Use Only)	W	8
0x880 002c	Configuration #2	R/W	8
0x880 0030	Configuration #3	R/W	8
0x880 0038	Transfer Count High (23:16)	R/W	8
0x8c0 0000 -> 0x8c0 0003	**Ethernet Controller Registers**		
0x8c0 0000	Register Data Port (RDP)	R/W	16
0x8c0 0002	Register Address Port (RAP)	R/W	16

1. (chip_sel, sb_pa_w,sb_pa_x,sb_pa_y) = sb_pa(27,26,23,22) would be the mapping for a typical system. This is the mapping shown.
2. It is recommended that software access the DMA2 registers with sb_pa[5] = 0 for future expansion.
3. Byte and 1/2 word accesses to this register are also allowed.
4. The "NEXT" Address/Byte Count registers are accessed at these addresses using the D_EN_NEXT bit in the D_CSR. Refer to "DMA2 DMA Core" for details.
5. The "NEXT" Address/Byte Count registers are accessed at these addresses using the P_EN_NEXT bit in the P_CSR. Refer to "DMA2 DMA Core" for details.

Functional Description

Overview

The 89C100 integrates the SBus DMA2 Controller, the NCR92C990 802.3 LAN Controller, and the NCR53C9X Fast SCSI Processor in a 160-pin plastic quad flat package (PQFP). The programmer's interface is identical to that of a discrete implementation using the above ICs. The chip-level address map is defined in the next section. Refer to "DMA2 DMA Core," "NCR92C990 Ethernet Core," and "NCR53C9X SCSI Core" for more information on the above devices.

89C100 and 89C105 Interdependencies

When the 89C100 and the 89C105 are used together, the 89C105 receives three clocks from the 89C100 (fpy_clk24, fpy_clk32, and scc_clk_20). The 89C100 simply provides oscillator pads on its pins because of a pin limitation on the 89C105. The 89C100 does not use these clock signals internally. The 89C100 also sends its three interrupt signals to the 89C105 for processing, they are; sb_d_irq_, sb_e_irq_, and sb_p_irq_ for; SCSI, Ethernet, and parallel port interrupts, respectively. Refer to "NCR89C105 Slave I/O" for a description of how the 89C105 handles interrupts. It is necessary to provide both enet_tclk and scsi_clk even if the Ethernet controller and SCSI controller are not used.

Technology

The 89C100 is a standard cell design, based on the NCR VS700H technology (.95 µ drawn,.7 effective). It consists of 60,000 equivalent gates.

Start-Up Information

The 89C100 receives a reset from the SBus signal sb_reset_. This signal must be asserted for at least 512 SBus clock cycles after the system power is stable, as specified by SBus specification B.0. After this, the 89C100 is ready for programming.

Functional Blocks

Overview

This section includes block diagrams, descriptions, and block-level address maps for the following:

- DMA2 Block
- SCSI Block
- Ethernet Block
- Test Block

DMA2 Block

The DMA2 block is a functionally and logically equivalent implementation of the L64854 SBus DMA controller with two minor differences.

Differences

- E_CSR bit 20, E_ALE/AS_ is not implemented. This allows use of other ENET controllers which is only an option in a discrete implementation.
- The pullup for the id_cs_ pin is provided internally. To use an external PROM simply connect id_cs_ with the PROM chip select pin. To signify absence of an external PROM connect id_cs_ to logic low.

DMA2 Block Diagram

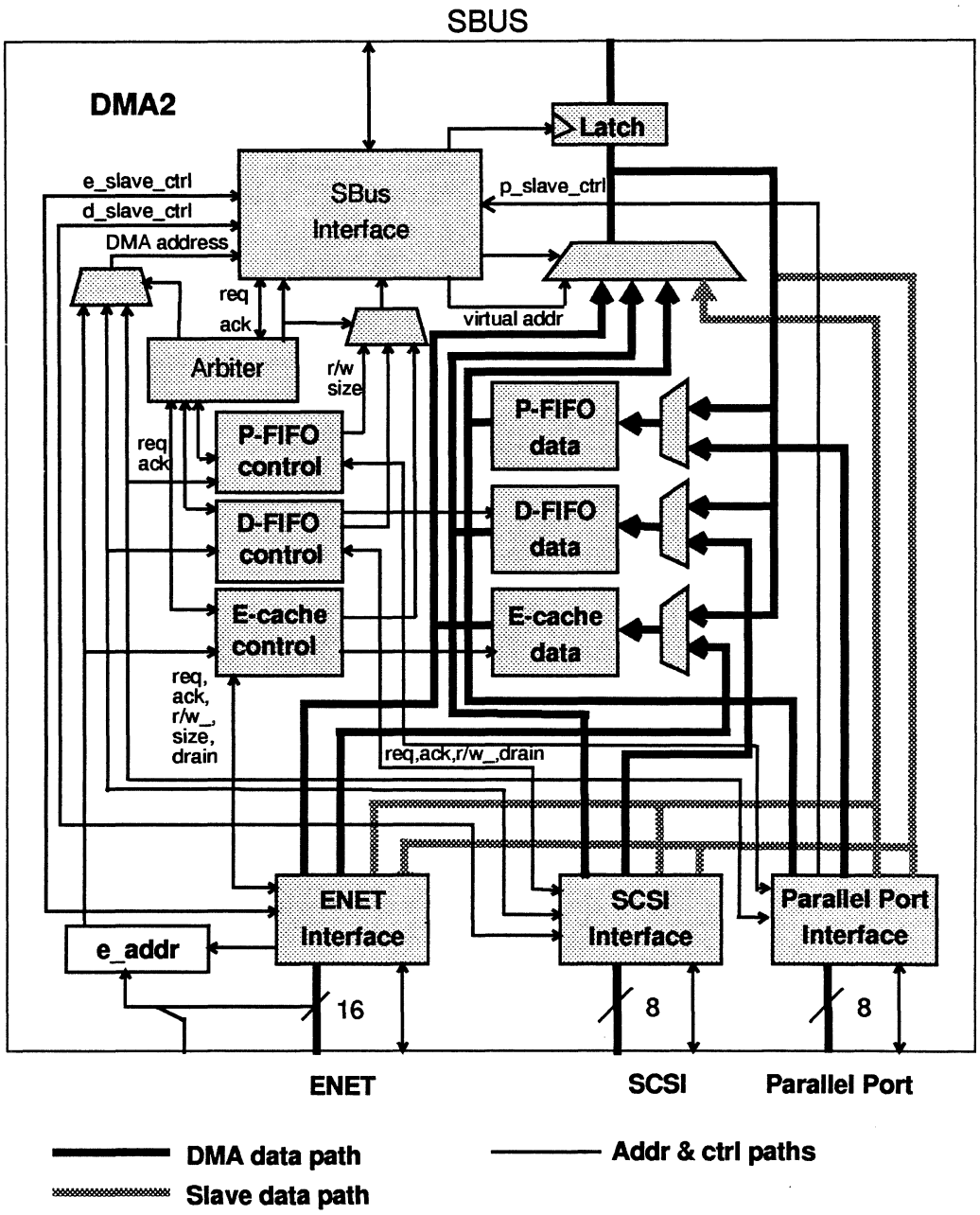


Figure 2-4 DMA2 Block Diagram

DMA2 Level Address Map

Table 2-8 DMA2 Level Address Map

sb_pa_(w,x,y)[1]	sb_pa(27:0)[2]	Register Accessed	Type	Size
000	**0x800 0000**	**Internal ID Register[3]**	R	32
	0x840 0000 -> 0x840 000f	**DMA2 ESP Registers**		
001	0x840 0000	Control/Status Register (D_CSR)	R/W	32
001	0x840 0004	Address Register (D_ADDR)[4]	R/W	32
001	0x840 0008	Byte Count Register (D_BCNT)[4]	R/W	24
001	0x840 000c	Test Control/Status Reg (D_TST_CSR)	R/W	32
	0x840 0010 -> 0x840 001f	**DMA2 Ethernet Registers**		
001	0x840 0010	Control/Status Register (E_CSR)	R/W	32
001	0x840 0014	Test Control/Status Reg (E_TST_CSR)	R/W	32
001	0x840 0018	Cache Valid Bits (E_VLD)	R/W	32
001	0x840 001c	Base Address Reg (E_BASE_ADDR)	R/W	8
	0xc80 0000 -> 0xc80 001f	**DMA2 Parallel Port Registers**		
110	0xc80 0000	Control/Status Register (P_CSR)	R/W	32
110	0xc80 0004	Address Register (P_ADDR)[5]	R/W	32
110	0xc80 0008	Byte Count Register (P_BCNT)[5]	R/W	32
110	0xc80 000c	Test Control/Status Reg (P_TST_CSR)	R/W	32
110	0xc80 0010	Hardware Configuration Reg (P_HCR)	R/W	16
110	0xc80 0012	Operation Configuration Reg (P_OCR)	R/W	16
110	0xc80 0014	Parallel Data Register (P_DR)	R/W	8
110	0xc80 0015	Transfer Control Register (P_TCR)	R/W	8
110	0xc80 0016	Output Register (P_OR)	R/W	8
110	0xc80 0017	Input Register (P_IR)	R/W	8
110	0xc80 0018	Interrupt Control Register (P_ICR)	R/W	16

1. (chip_sel,sb_pa_(w,x,y)) = sb_pa(27,26,23,22) in a typical system.
2. This column specifies a 27-bit address for systems using the same chip_sel and sb_pa_(w,x,y) mapping as described in 1.
3. Byte and 1/2 word accesses to this register are also allowed. If the id_cs_ pin is tied to ground, then addresses 0x800,0000-0x83F,FFFF all map to the internal chip ID register. If the id_cs_ pin it tied high or allowed to float (it has an internal pullup), then this address range maps to the external ID PROM.
4. The "NEXT" Address/Byte Count registers are accessed at these addresses using the D_EN_NEXT bit in the D_CSR. Refer to "DMA2 DMA Core" for details.
5. The "NEXT" Address/Byte Count registers are accessed at these addresses using the P_EN_NEXT bit in the P_CSR. Refer to "DMA2 DMA Core" for details.

SCSI Block

The SCSI block is based on the NCR53C9X ASF which is a superset of (and fully backward compatible with) the NCR53C90A previously found on SPARCstations.

Differences

The 89C100 implementation of the SCSI channel differs from former discrete implementations as follows:

- The following pins exist in the discrete implementation but not in the 89C100 chip:
  - TGS, IGS, DIFFM—Not needed for single-ended SCSI
  - RESETO—Not normally used in a system
  - All current SPARCstation designs operate as single-ended SCSI only, with the RESETO pin floating so this was chosen for the 89C100.
- The NCR53C9X ASF has two additional registers:
  - Configuration Register 3 (used to enable Fast SCSI)
  - Transfer Count High (allows for up to 16 Mbyte block transfers)
- The NCR53C9X ASF can be clocked at 40 Mhz (necessary for Fast SCSI). Software using hardware clocked at this speed will need to adjust the Clock Conversion Factor Register accordingly.

Refer to "NCR53C9X SCSI Core" for details.

SCSI Block Diagram

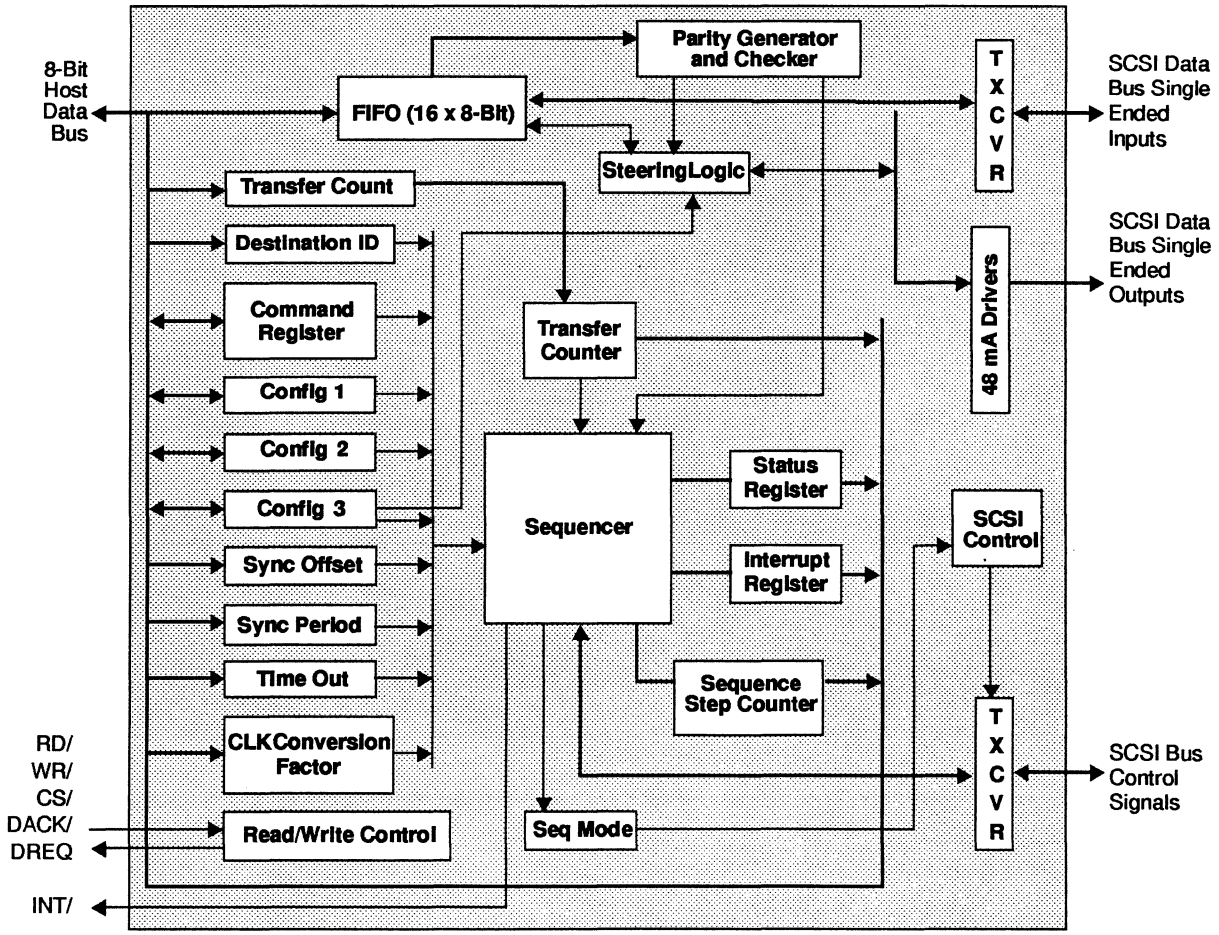


Figure 2-5 SCSI Block Diagram

SCSI-Level Address Map

Table 2-9 NCR53C9X Registers

sb_pa_(w,x,y)[1]	sb_pa(27:0)	Register Accessed	Type	Size
	0x880 0000 - > 0x880 003f	**SCSI Controller Registers**		
010	0x880 0000	Transfer Count Low (7:0)	R/W	8
010	0x880 0004	Transfer Count Middle (15:8)	R/W	8
010	0x880 0008	FIFO Data	R/W	8
010	0x880 000c	Command	R/W	8
010	0x880 0010	Status	R	8
010	0x880 0010	Select-Reselect Bus ID	W	8
010	0x880 0014	Interrupt	R	8
010	0x880 0014	Select-Reselect Time-Out	W	8
010	0x880 0018	Sequence Step	R	8
010	0x880 0018	Synchronous Transfer Period	W	8
010	0x880 001c	FIFO Flags	R	8
010	0x880 001c	Synchronous Offset	W	8
010	0x880 0020	Configuration #1	R/W	8
010	0x880 0024	Clock Conversion Factor	W	8
010	0x880 0028	Test (Chip Test Use Only)	W	8
010	0x880 002c	Configuration #2	R/W	8
010	0x880 0030	Configuration #3	R/W	8
010	0x880 0038	Transfer Count High (23:16)	R/W	8

1. (chip_sel,sb_pa_(w,x,y)) = sb_pa(27,26,23,22) in a typical system.

Ethernet Block

The ENET block is based on the NCR92C990 ASF which is a superset of (and fully backwards compatible with) the AM7990 previously found on SPARCstations.

Differences

The only differences between the NCR92C990 and the AM7990 are:

- Programmable Inter Packet Gap (IPG). The NCR92C990 allows one to program the Transmit after Transmit (Tx-Tx) or Transmit after Receive (Rx-Tx) IPG time within the range of 9.6 μ sec (the Ethernet spec minimum IPG) to 22.4 μsec.

  This feature can be accessed via the upper bits of CSR3, as shown below:

CSR3 Bit	Description
15	Enable programmable IPG (default is 0, not programmable)
14-12	Rx-Tx IPG value: (default=110 or 20.8 μsec)
11-9	Tx-Tx IPG (default=000 or 9.6 μsec)
8-0	As normally defined in AM7990

  NOTE:
  - The formula for calculating the IPG value is [9.6 + 1.6*(3 bit IPG #)] μsec
  - The default values chosen to closely mimic the operation of the AM7990.

  The programmable IPG time assumes its default value should ANY of the following occur:

- Ethernet hard reset (either as a result of an SBus reset or the E_CSR E_RESET bit of the DMA2).
- The CSR0 STOP bit is set.
- The CSR0 INIT bit is set.
- The CSR3 Enable programmable IPG is reset to 0.

Software drivers should set CSR3 right after the last INIT, while waiting for the IDON interrupt. It is recommended that the Enable, Rx-Tx IPG and Tx-Tx IPG fields be ORed into all CSR3 writes.

- The NCR92C990 core used in the 89C100 differs from both the AM7990 and the stand-alone NCR92C990 core with respect to the memory error (MERR) time-out value. The description for bit 11 (ME) in the Control/Status Register 0 tables shows that READYb_IN must be received within 25.6us after asserting DAL_OUT(15:0). This value has been extended to 102.4us (4X) to avoid memory errors in high latency systems. This feature helps to avoid unneeded reinitializations of the NCR92C990 during periods of high system activity.

  Refer to "NCR92C990 Ethernet Core" for details.

Ethernet Block Diagram

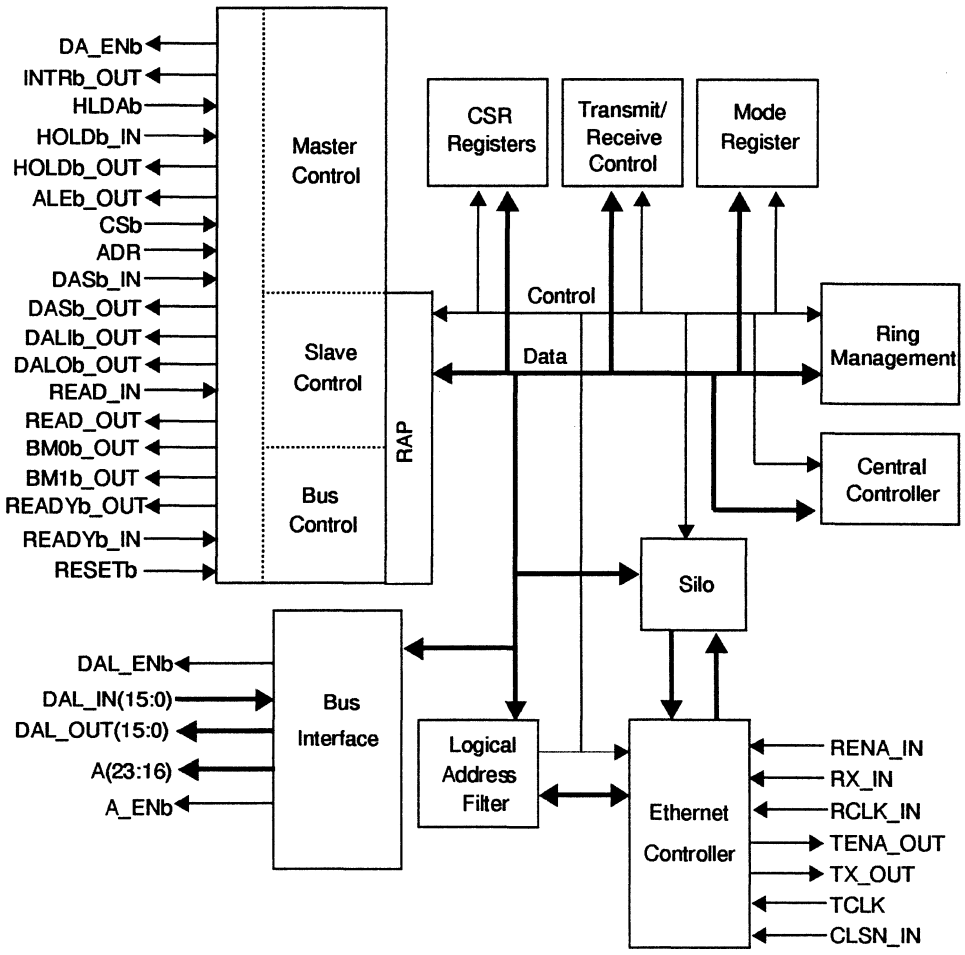


Figure 2-6 Ethernet Block Diagram

Ethernet-Level Address Map

Table 2-10 NCR92C990 Registers

sb_pa_(w,x,y)[1]	sb_pa(27:0)	Register Accessed	Type	Size
	0x8c0 0000 -> 0x8c0 0003	Ethernet Controller Registers		
011	0x8c0 0000	Register Data Port (RDP)	R/W	16
011	0x8c0 0002	Register Address Port (RAP)	R/W	16

1. (chip_sel,sb_pa_(w,x,y)) = sb_pa(27,26,23,22) for the mapping shown.

Test Block

The 89C100 contains an IEEE JTAG 1149.1 compliant test controller and boundary scan architecture. All mandatory instructions are supported, and this document contains the chip specific boundary scan information. The 89C100 also contains internal test logic and reserved instructions. The basic description of this logic appears below but is not supported.

This section describes the goals and implementation of the testability features implemented in the 89C100. These features have been incorporated to provide a structured test approach to both device fabrication testing and board-level testing and debug.

JTAG Scan Access

The goals for the 89C100 testability are to provide for high stuck at fault coverage at both the IC and board level. This is provided by the incorporation of an IEEE 1149.1 (JTAG) compatible TAP controller and boundary scan, which in conjunction with modular broadside access modes provides access to each of the major functional blocks on the I/O chips through either full scan (in the case of the DMA2 block) or boundary scan (in the case of the NCR ASFs). These ASFs are tested during device fabrication by a full broadside pin mode that provides direct access to all ports of each ASF from the device pins. This allows standardized test patterns to be applied directly to each ASF without the need for additional high fault coverage patterns for these blocks. At the board-level, the JTAG compatible boundary scan provides for complete access to PCB interconnect, including die to package bonding.

Block Access Modes

Diagnostic multiplexing between the pad ring and the internal ASFs is configurable into four different modes: Normal Mode, in which the device operates as required in the system; TBLK1 Mode, for scanned logic, in which all the ports to the DMA2 logic are accessible via scannable elements. In addition, the internal scan chain of the block is connected in series with the boundary scan chain, and the partition scan chain (if one is required) to form a complete scan path for access to all state and primary inputs of the block. TBLK2 and TBLK3, for NCR designed logic, in which each block is presented to the pins of the device as if it were a stand-alone device.

Tristate Pin Function

All output pins of the device are tristate-able, controlled by elements in the boundary scan chain, to support manufacturing system test. At power-up and in normal operation of the system this function is disabled by the TRSTB JTAG pin being held in the active low state.

Block Diagnostic Modes

TBLK1 (Internal Scan) Diagnostic Mode

Figure 2-1 illustrates the operation of the TBLK1 diagnostic mode. In this mode, the test logic is configured to connect every primary input to the Q-output of a scannable flip-flop and every primary output to the D-input of a scannable flip-flop. In addition, every flip-flop inside the block is configured into a single scan chain, known as the internal, or "iscan" chain.

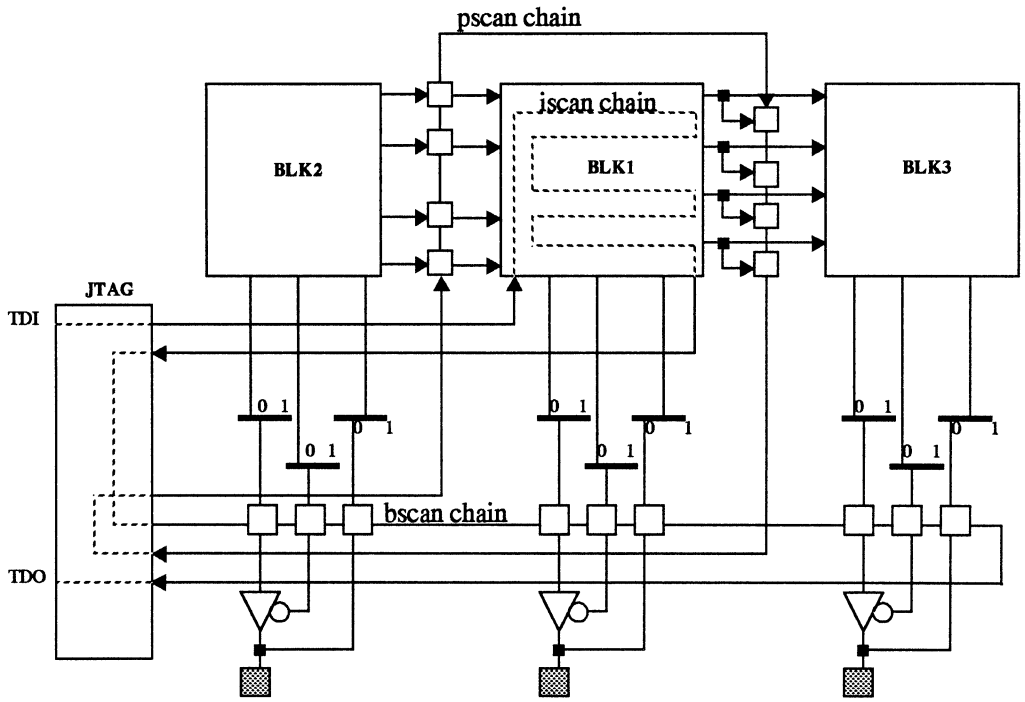


Figure 2-7 TBLK1 (Internal Scan) Diagnostic Mode

TBLK2/TBLK3 Diagnostic Mode

Figure 2-2 illustrates the operation of the TBLK2(TBLK3) diagnostic mode. In these modes the test logic is configured to connect internal inputs and outputs to BLK2 (BLK3) to pins normally assigned to BLK1 or BLK3(BLK2). Since these blocks are non-scannable, the only function of the JTAG controller in this mode is to configure the multiplexor logic into this mode. Hence the scan datapath is placed in BYPASS mode.

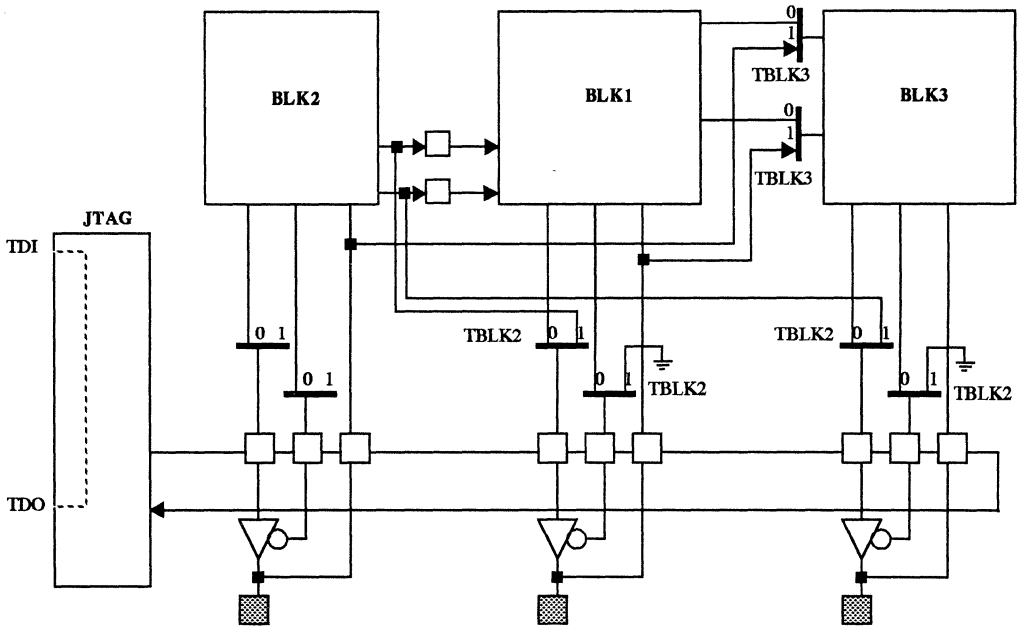


Figure 2-8 TBLK2/TBLK3 Diagnostic Mode

Outputs of the block that normally connect to BLK1 are multiplexed into the chip outputs of the other blocks, configured by the TBLK2(TBLK3) mode signal. Inputs to BLK2 (BLK3) are multiplexed with inputs from the other blocks. Figure 2-2 shows how the outputs of BLK2 and the inputs of BLK3 are configured.

Other JTAG test modes (TBLK2_BS and TBLK3_BS) are provided that operate identically except that the scan data path is configured to pass through the boundary chain. This allows application of the broadside test vectors to the blocks using the boundary chain to drive primary inputs and sample primary outputs in a pseudo-static manner, i.e. it does not directly support complex edge relationships between inputs. Instead these vectors must be "exploded" into multiple boundary scan vectors.

JTAG Controller

The JTAG controller contains the following elements:

- NCR Tap controller
- Scan Datapath including instruction register, bypass register and ID register
- Clock control register and state machine

The following figure shows a simplified block diagram of the JTAG controller. It has been partitioned into two main functional areas: Scan Datapath and Scan Control Logic.

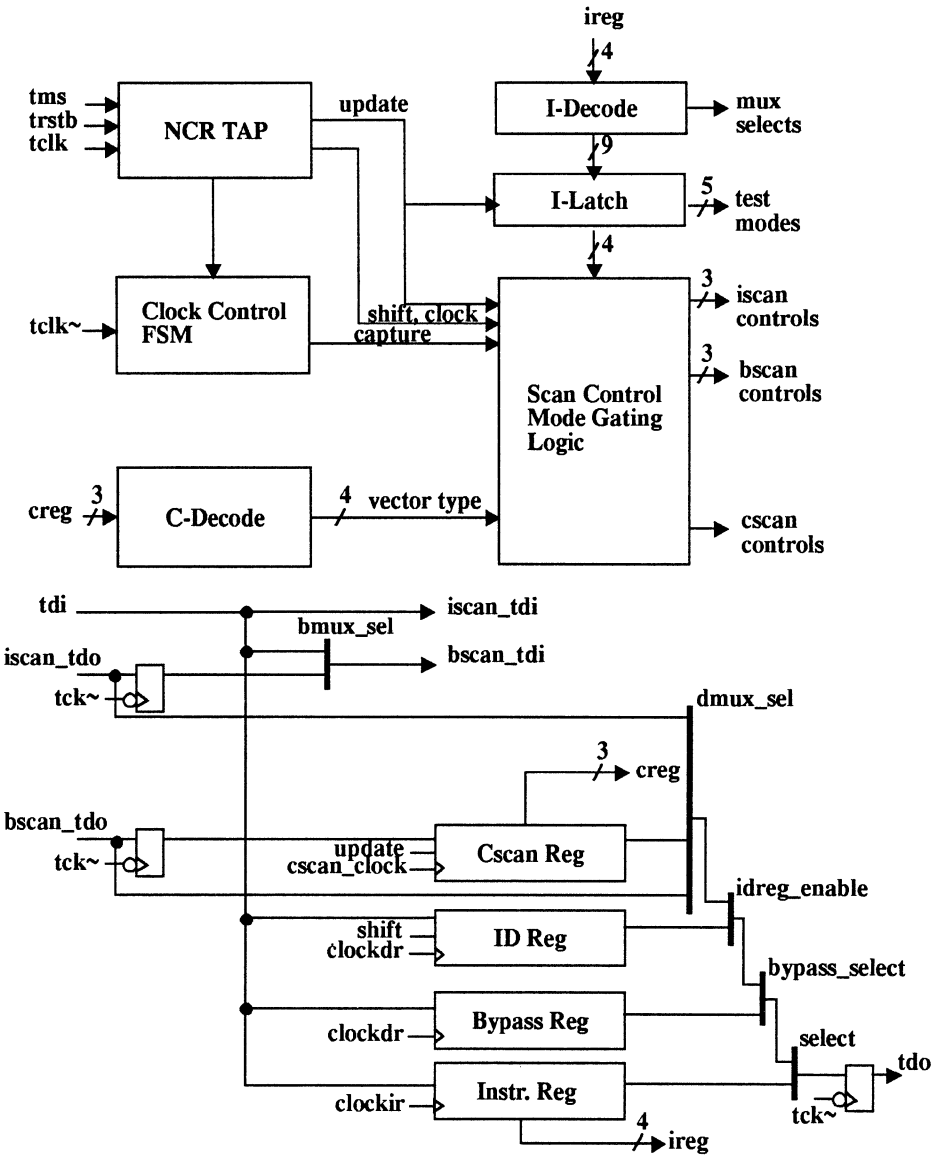


Figure 2-9 JTAG Controller Block Diagram

The NCR tap controller is a standard cell implementation of a reference 1149.1 tap state machine[1]. It is connected directly to the test access port on the 89C100 (TCK, TMS, TRSTB) and generates the basic scan controls (clock_dr, clock_ir, reset_l, select, shift_dr, shift_ir, update_dr, update_dr) which are used to control the scan architecture.

The NCR TAP implementation has been modified slightly to also make available the TAP state for use by supplemental state machines. The NCR state machine implements the reference state diagram described by the 1149.1 specification[2]. The state coding is shown Table 2-12.

1. IEEE Std. 1149.1-1990 chapter 5.
2. IEEE Std. 1149.1-1990 page 5-1.

Table 2-11 State Assignments for NCR TAP Controller

Controller State	State[3:0]
Exit2-DR	0
Exit1-DR	1
Shift-DR	2
Pause-DR	3
Select-IR-Scan	4
Update-DR	5
Capture-DR	6
Select-DR-Scan	7
Exit2-IR	8
Exit1-IR	9
Shift-IR	A
Pause-IR	B
Run-Test/Idle	C
Update-IR	D
Capture-IR	E
Test-Logic-Reset	F

The instruction register for the 89C100 is a 4- bit register comprised of simple scannable elements. When the TAP state machine issues a reset signal this register is initialized to the IDCODE (1110) instruction. The parallel inputs of the instruction register are not used to load design-specific information and are tied-off to logic 0.

The 4-bit output of the instruction register is followed by an instruction decode stage which decodes up to 16 unique instructions. Not all of these are used by the 89C100 but are given mnemonics for completeness.

Table 2-2-12 lists these mnemonics and the instruction value that corresponds to

them:

Table 2-12 Decoded JTAG Instructions

Value	Mnemonic	Description
0000[1]	EXTEST	Boundary scan board interconnect test.
0001	SAMPLE	Boundary scan sample/preload.
0010	TBLK1	BLK1 ATPG scan test mode (Internal+-Boundary+Clock chains).
0011	TBLK2	BLK2 broadside test mode (Bypass).
0100	TBLK3	BLK3 broadside test mode (Bypass).
0101	RESERVED	—
0110	PSCAN	Reserved for partition scan (if implemented, otherwise Bypass).
0111	INTEST	Boundary scan capture of internal I/O.
1000	TBLK2_BS	BLK2 boundary scan test mode.
1001	TBLK3_BS	BLK3 boundary scan test mode.
1011	TPSCAN	Reserved for BLK1 tester partition scan mode (if implemented, otherwise Bypass). Other BLK1 pins controlled by broadside tester.
1100	BPSCAN	Reserved for BLK1 boundary partition scan mode (if implemented, otherwise Bypass). Other BLK1 pins controlled by boundary scan.
1101	ZMODE0	General purpose test mode.
1110	IDCODE	Device ID register.
1111[1]	BYPASS	Bypass mode.

1. Required instruction.

The scan controls decoded by these instructions configure the scan datapath, the test multiplexors and control the clocks and pseudo clocks for the test mode in progress.

Instruction Decode

The I-Decode logic converts the 4-bit instruction register contents into decoded signals that control mux selection in the scan datapath and test mode configuration in the ASFs. Nine of these signals are latched by the I-latch to provide glitch free values on these signals which are updated during the IR-Update state.

Clock Control FSM

The clock control finite state machine monitors the state output from the NCR TAP controller and determines when it is necessary to insert the capture clock and/or pseudo clocks required to support ATPG stimulus application to BLK1. The clock gating is designed such that the boundary clock is guaranteed to be asserted at all elements of the boundary scan chain before it is applied to either the clock or pseudo clock (set/reset) inputs to the BLK1 internals. This requirement is present due to the fact that ATPG vectors have an assumed order in which stimulus is applied to the circuit and state or primary outputs are captured. The clock control state machine in the 89C100 has been verified to support the requirements of TestScan ATPG from Cadence Design Systems, Inc. although it may function equally well with other ATPG systems.

The assumed sequence of operations required for ATPG pattern application is:

1. Stimulate pins - boundary and pscan chains shift/update sequence.
2. Stimulate shift register/latches - internal scan chain shift in.
3. Measure pins - boundary and pscan chain capture sequence.
4. Pulse clocks/pseudo clocks - internal chain clock/set/reset.
5. Measure shift register/latches - internal scan chain shift out.

The Clock Control FSM has been designed to support this event ordering in a single continuous shift-update-capture-shift sequence. In TBLK1 mode the scan chains within the 89C100 are concatenated into a single chain containing internal, boundary and clock control scan chains. Hence after an initial shift-update sequence, the requirements of (1) and (2) have been met. The Capture-DR state is then used to measure the state of the primary outputs of BLK1 by issuing a clock to the boundary with the shift control not asserted. A delayed version of the clock (or update pulse in the case of the pseudo clocks) is then used to apply clock, set or reset to the internal scan chain to implement the internal chain capture. This occurs only when indicated by the value of the "capture" output from the clock control state machine.

Clock Control Register

The other three bit positions in the clock control scan chain are transferred to the clock control register during a DR-update sequence, where they are decoded by the C-Decode logic to specify which vector class the following capture sequence belongs to out of the following categories:

1. Shift only, no capture.
2. Capture scan chain, (i.e. shift high during capture clock).
3. Normal clocked vector.
4. Set vector, no clock.
5. Reset vector, no clock.

Since the last two categories are only required when the logic under test contains asynchronous sets or resets, they are not required for the 89C100.

Mode Gating Logic

The decoded vector type information is combined with the clock control FSM information and the primary scan controls from the NCR TAP controller and instruction register decodes to generate control signals for each of the four scan chains with the 89C100: iscan, bscan, pscan (if present) or cscan. These signals are buffered and distributed throughout the device to the various chain elements. Since the 89C100 does not require a partition scan chain for its final implementation, these controls have been deleted.

Scan Datapath

The scan datapath within the JTAG Controller contains the chain configuration logic, implemented as a series of multiplexors; inter-chain flops to guarantee hold margins; the Cscan register; JTAG compliant ID, BYPASS and IR shift registers and the TDO output multiplexors and flop. This datapath, like the external scan chains and test logic is controlled by the scan control logic described above. The only variation from a more conventional IEEE 1149.1 implementation is the ability to configure the scan chain into various different modes based on the instruction type. The use of the hold flops, clocked by TCK~ is simply an implementation detail to reduce the effects of clock skew between the separate scan chains.

The 89C100's scan datapath does not include a partition scan chain, as in its final implementation this functionality has been incorporated into the internal, or "iscan" chain to facilitate physical implementation of the device. The elements of the embedded partition scan chain are therefore controlled by the same datapath controls as the existing iscan chain elements.

Table 2-14 lists the lengths of the various scan chains that comprise the 89C100's scan datapath.

Table 2-13 89C100 JTAG Chain Lengths

Chain Name	Number of Elements
BYPASS	1
I.D.	32
Instruction Register	4
Internal	893
Boundary	209
ATPG	1105

Performance

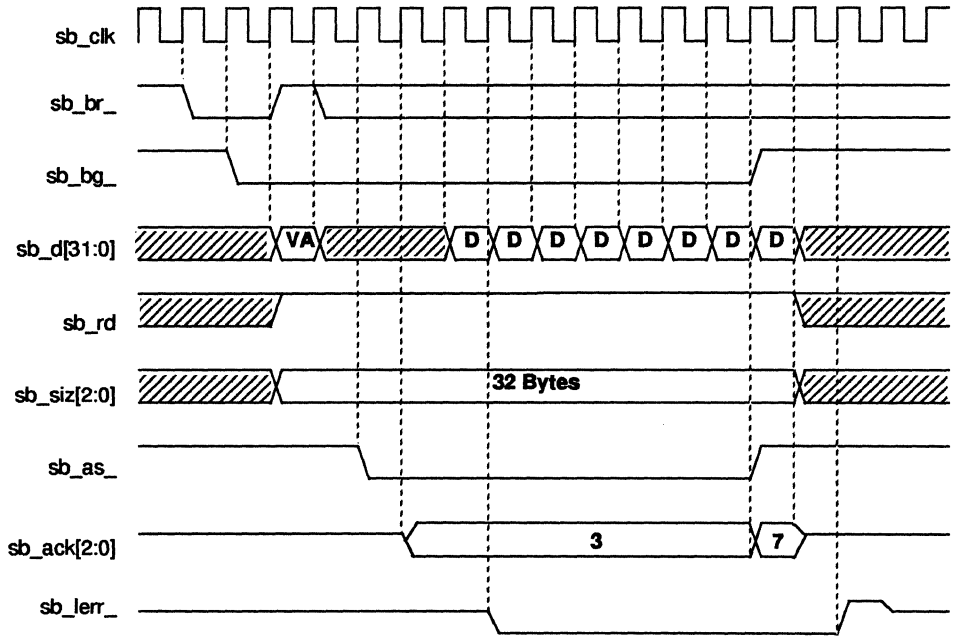
The design as implemented in NCR's VS700H 0.95um (drawn) standard cell library has been verified to operate at a 5MHz scan rate.

The JTAG controller occupies approximately 700 gates, and the scan overhead for the simple multiplexed flop scan element that it supports is estimated at about 10% from a gate count perspective, 5% in total area overhead.

Functional Timing Diagrams

These timing diagrams illustrate a DMA access by the 89C100 on the SBus as well as all the common slave accesses to the 89C100 address ports. Optimal translation time (1 cycle) is assumed for DMA.

SBus DMA Burst Read (1 Word/Clock)



(Data flow from memory to the DMA2)
For other possible SBus cycles, see the SBus Specification; this is just one example.

Figure 2-10 SBus DMA Burst Read (1 Word/Clock)

DMA2 ENET and SCSI Register Accesses

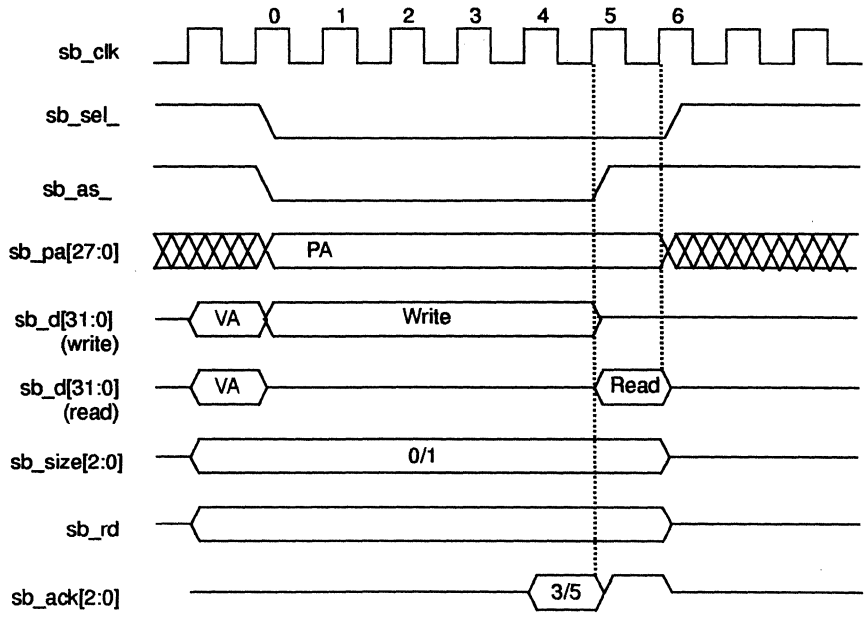


Figure 2-11 DMA2 ENET and SCSI Register Accesses

DMA2 Parallel Port Register Accesses

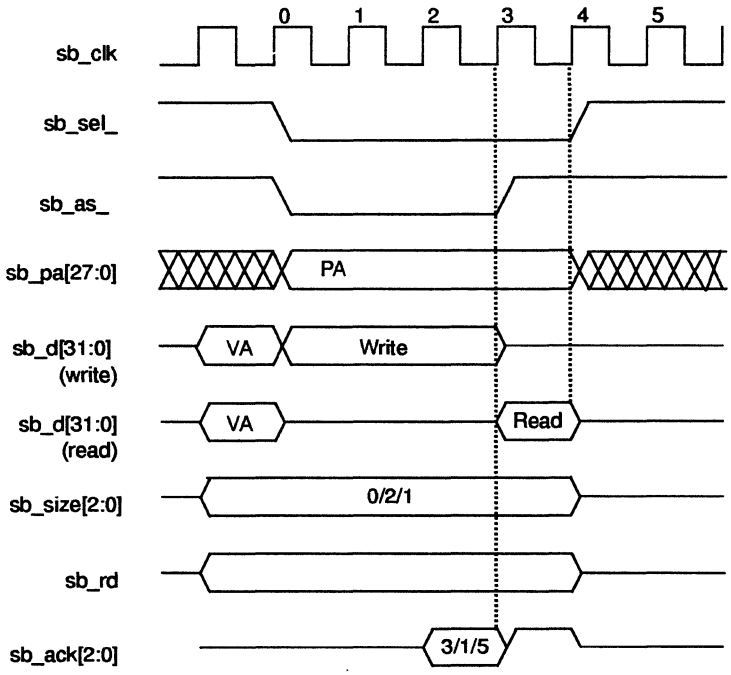


Figure 2-12 DMA2 Parallel Port Register Accesses

ENET Controller Register Accesses

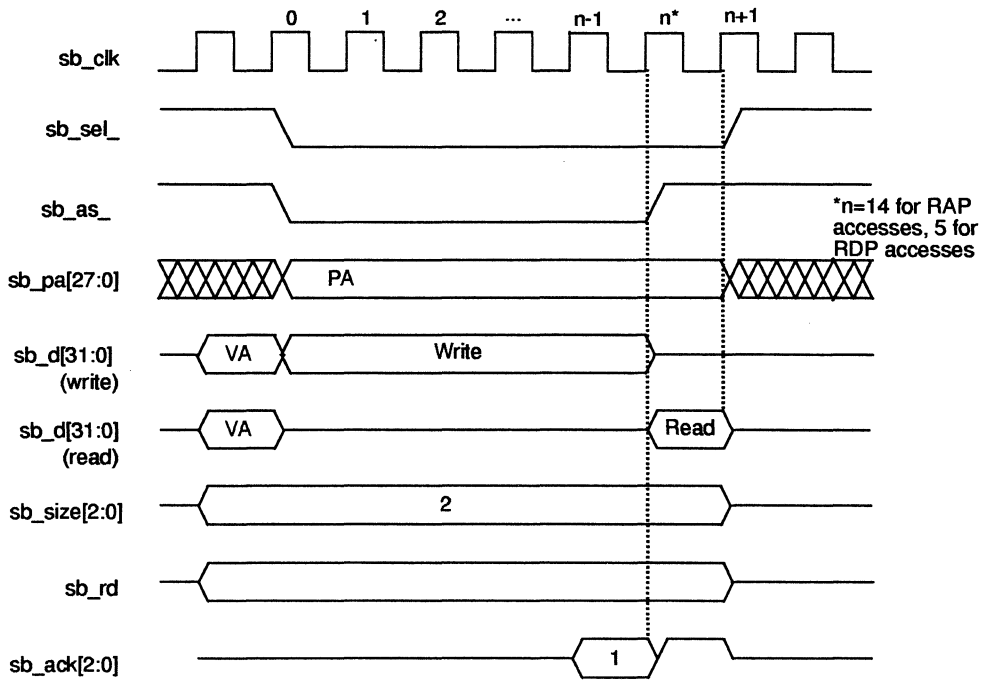


Figure 2-13 ENET Controller Register Accesses

SCSI Controller Register Access

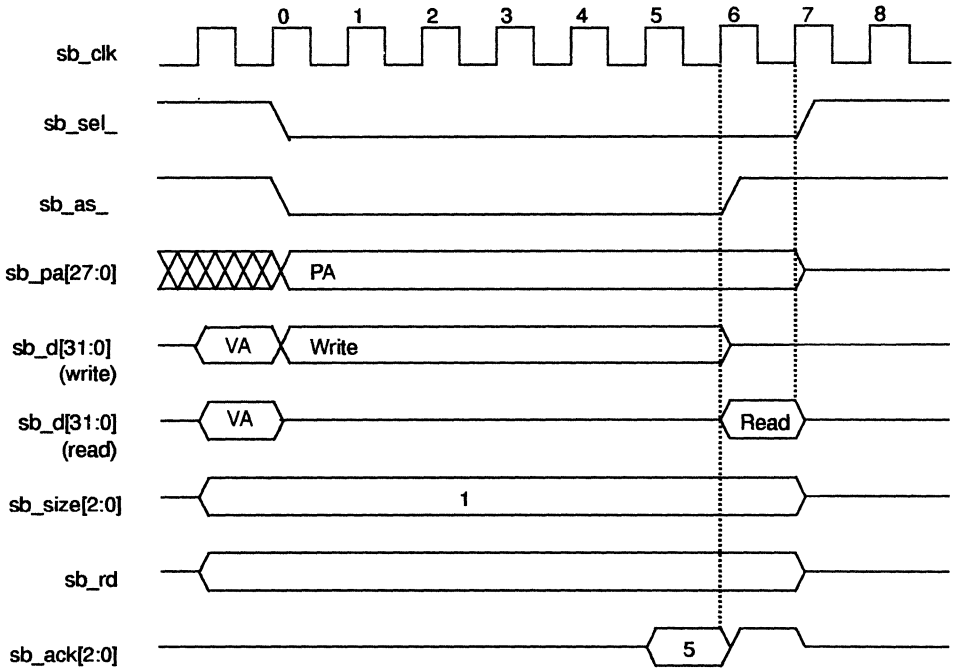


Figure 2-14 SCSI Controller Register Accesses

External ID PROM Read Access

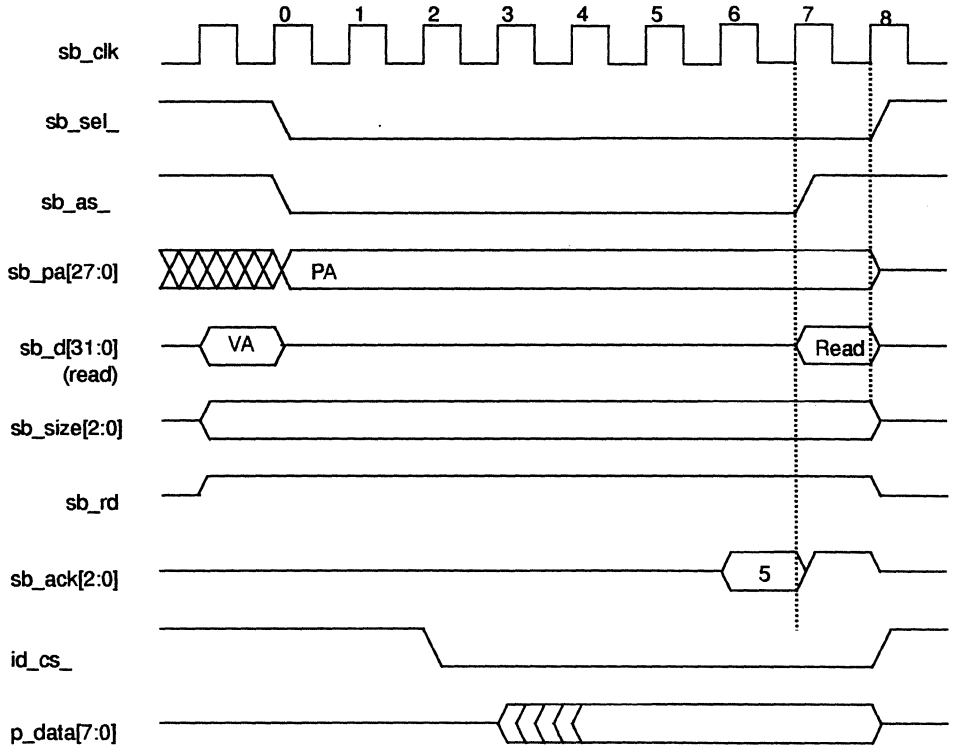


Figure 2-15 External ID PROM Read Access

Electrical Considerations

This section defines the following electrical specifications for the 89C100:

- Absolute and recommended operating conditions
- DC characteristics
- AC characteristics
- Power consumption

The 89C100 is implemented in NCR's VS700H process, and normal handling precautions required by all MOS devices must be observed. Regardless of the fact that all inputs and outputs are protected against ESD damage by internal protection structures, the device can be degraded or destroyed by exposure to high electrostatic fields.

Absolute Maximum Ratings

The following section details the absolute maximum ratings of the 89C100 chip. Operation of the device at values in excess of those listed here will result in degradation or destruction of the device and should be avoided. This table does not imply that functional operation at conditions above those listed in the "Recommended Operating Conditions" is possible. This is a stress rating and operation of a device at or above this rating may cause failure or affect reliability.

Table 2-14 Absolute Maximum Operating Conditions

Name	Symbol	Min	Max	Units
Supply Voltage	V_{dd}	-0.5	+7	Volts
Input, Output Voltage	V_{im}	-0.5	V_{dd} + 0.5	Volts
Current Drain Vdd and Vss pins	I_i		100	mA
Lead Temperature (less than 10 second soldering)	T_l		250	°C
Operating Temperature	T_j	0	70	°C
Storage Temperature	T_s	-55	150	°C

Recommended Operating Conditions

The following section details the recommended DC operating conditions for the 89C100 chip:

Table 2-15 Recommended Operating Conditions

Name	Symbol	Min	Nom	Max	Units
Supply Voltage	Vdd	4.75	5.0	5.25	Volts
Operating Free-Air Temperature	Ta	0	25	70	°C

DC Characteristics

This table specifies the DC characteristics of the 89C100 chip over the range of the recommended operating conditions.

Table 2-16 DC Characteristics

Name	Symbol	Min	Nom	Max	Units
TTL Input Receiver					
High Level Input Voltage	V_{ih}	2.0			Volts
Logic Low Input Voltage	V_{il}			0.8	Volts
DS1216 Schmitt Input Receiver					
High Level Input Voltage	V_{ih}	1.9	1.6		Volts
Logic Low Input Voltage	Vil		1.2	0.9	Volts
DS1218 Schmitt Input Receiver					
High Level Input Voltage	V_{ih}	2.1	1.8		Volts
Logic Low Input Voltage	V_{il}		1.2	0.9	Volts
DS1238 Schmitt Input Receiver					
High Level Input Voltage	V_{ih}	4.1	3.8		Volts
Logic Low Input Voltage	V_{il}		1.2	0.9	Volts
Minimum high-level source current, V_{oh} = 2.4 V					
2 mA buffer	I_{oh}	2.0			mA
4mA buffer		4.0			
8mA buffer		8.0			
16 mA buffer		16.0			
24 mA buffer		24.0			
48mA buffer		n/a			
Minimum low-level sink current, V_{ol} = 0.4 V					
2 mA buffer	I_{ol}	2.0			mA
4mA buffer		4.0			
8mA buffer		8.0			
16 mA buffer		16.0			
24 mA buffer		24.0			
48mA buffer		48.0			
SCSIPAD (V_{ol} = 0.5 V)		48.0			
SCSIPADF (V_{ol} = 0.5 V)		48.0			

Table 2-16 DC Characteristics

Name	Symbol	Min	Nom	Max	Units
Input Leakage Current	I_{in}			±10	µA
Tristate Output Leakage Current	I_{oz}			±10	µA
High Level Output Voltage	V_{oh}	4.4	4.5		Volts
Low Level Output Voltage	V_{ol}		0.0	0.1	Volts
Input Capacitance	Ci		6		pF
Output Capacitance	Co		6		pF
Bidirectional Pin Capacitance	Cb		6		pF
SCSI Pin Capacitance				10	pF

AC Characteristics

The following table lists the 89C100 AC characteristics specification:

Table 2-17 89C100 AC Characteristics

Number	Description	Conditions	Min	Max	Units
SBus Timing					
1	Clock Period		40.0	60.0	ns
2	Clock High		17.0		ns
3	Clock Low		17.0		ns
4	Hold wrt CLK Rising		0.0		ns
5	Setup to CLK Rising		15.0		ns
6	Hold wrt CLK Rising	See Note 1	1.0		ns
7	Hold wrt CLK Rising		0.0		ns
8	CLK Rising to Output Valid	160 pF load	2.5	22.5	ns
9	CLK Rising to Output Invalid	160 pF load	2.5	20.0	ns
Parallel Port Timing					
10	CLK to p_d_strb	75 pF		35	ns
11	p_d_strb nom. width	DSW=0,1,2,3	3		sb.clk periods
12	p_data valid to p_d_strb assert	75 pF	5		ns
13	p_data valid (nominal)	DSS=0,DSN=3	6		sb_clk periods
14	p_ack, p_bsy setup to clk		5		ns

Table 2-17 89C100 AC Characteristics (Continued)

Number	Description	Conditions	Min	Max	Units
15	p-ack,p_bsy input pulse width		3		sb_clk periods
16	p_d_strb setup to clk		5		ns
17	p_d_strb input pulse width		3		sb_clk periods
18	p_data setup to p_d_strb		36		ns
19	p_data input hold from p_d_strb		4		sb_clk periods
20	p_d_strb to p_bsy valid	75 pF	2	3+26 ns	sb_clk periods
21	CLK to p_ack, p_bsy	75 pF		40	ns
22	p_ack, p_bsy nominal pulse width	75 pF	3		sb_clk periods
23	CLK to output	75 pF		35	ns
SCSI Timing					
24	Clock period (t_{CP})		25	83.3	ns
25	Synchronization latency		t_{CL}	$t_{CL+}t_{CP}$	ns
With FASTCLK bit reset					
26	Clock frequency, asynchronous	See Note 2	12*	25	MHz
27	Clock frequency, synchronous	See Note 2	20*	25	MHz
28	Clock high		14.58	$0.65*t_{CP}$	ns
29	Clock low (t_{CL})		14.58	$0.65*t_{CP}$	ns
With FASTCLK bit set					
26	Clock frequency, asynchronous	See Note 2	20*	40	MHz
27	Clock frequency, synchronous	See Note 2	38*	40	MHz
28	Clock high		$0.40*t_{CP}$	$0.60*t_{CP}$	ns
29	Clock low (t_{CL})		$0.40*t_{CP}$	$0.60*t_{CP}$	ns
Asynchronous SCSI					
30	Data setup to scsi_ack_/ scsi_req_ low		60		ns
31	Data hold from scsi_req_ high/scsi_ack_ low	See Note 3	5		ns
32	scsi_ack_ low to scsi_req_ high			50	ns

Table 2-17 89C100 AC Characteristics (Continued)

Number	Description	Conditions	Min	Max	Units
33	scsi_ack_ high to scsi_req_ low (data already setup)	See Note 4		45	ns
34	scsi_req_ high to scsi_ack_ high			50	ns
35	scsi_req_ low to scsi_ack_ low (data already setup)	See Note 4		50	ns
36	Data setup to scsi_req_/ scsi_ack_ low		0		ns
37	Data hold from scsi_req_/ scsi_ack_ low			18	ns
Synchronous SCSI					
Normal SCSI					
38	Data setup to scsi_req_/ scsi_ack_ low		55		ns
39	Data hold from scsi_req_/ scsi_ack_ low		100		ns
40	scsi_req_/scsi_ack_ assertion period		90		ns
41	scsi_req_/scsi_ack_ negation period		90		ns
Fast SCSI					
38	Data setup to scsi_req_/ scsi_ack_ low		25		ns
39	Data hold from scsi_req_/ scsi_ack_ low		35		ns
40	scsi_req_/scsi_ack_ assertion period		30		ns
41	scsi_req_/scsi_ack_ negation period		30		ns
Synchronous SCSI Input Cycle					
42	Data setup to scsi_req_/ scsi_ack_ low		5		ns
43	Data hold from scsi_req_/ scsi_ack_ low		15		ns
44	scsi_req_ assertion period		27		ns
45	scsi_req_ negation period		20		ns
46	scsi_ack_ assertion period		20		ns
47	scsi_ack_ negation period		20		ns
Ethernet Timing					
48	enet_tclk period		99	101	ns

Table 2-17 89C100 AC Characteristics (Continued)

Number	Description	Conditions	Min	Max	Units
49	enet_tclk high pulse duration		45		ns
50	enet_tclk low pulse duration		45		ns
51	enet_tclk rise time			8	ns
52	enet_tclk fall time			8	ns
53	enet_tena propagation delay after rising edge of enet_tclk			25	ns
54	enet_tena hold time after enet_tclk rising		7		ns
55	enet_tx propagation delay after enet_tclk rising			32	ns
56	enet_tx hold time after enet_tclk rising		7		ns
57	enet_rclk period		85	118	ns
58	enet_rclk high pulse duration		38		ns
59	enet_rclk low pulse duration		38		ns
60	enet_rclk rise time			8	ns
61	enet_rclk fall time			8	ns
62	enet_rx data rise time			8	ns
63	enet_rx data fall time			8	ns
64	enet_rx data hold time from enet_rclkrising		5		ns
65	enet_rx data setup time to enet_rclkrising		40		ns
66	enet_rena low duration		120		ns
67	enet_clsn high duration		110		ns
68	enet_rena hold time after the rising edge of enet_rclk		1		ns
69	enet_rena defer before enet_tena		356		ns
70	enet_rena extended after enet_rclk last falling			275	ns

NOTE 1: This is the only violation of SBus Specification B.0. No known implementation to date provides less than 1.0 ns hold time on these signals.

NOTE 2: These minimum numbers required to comply with ANSI SCSI specification. For Synchronous SCSI data transfers and FASTCLK enabled, the clock inputs must also meet the following requirements: $2*t_{CP}+t_{CL}>97.92$ ns and $2t_{CP}+t_{CH}>97.92$ ns

NOTE 3: FIFO is not empty.

NOTE 4: FIFO is not full.

AC Timing Diagrams

SBus Input Signals

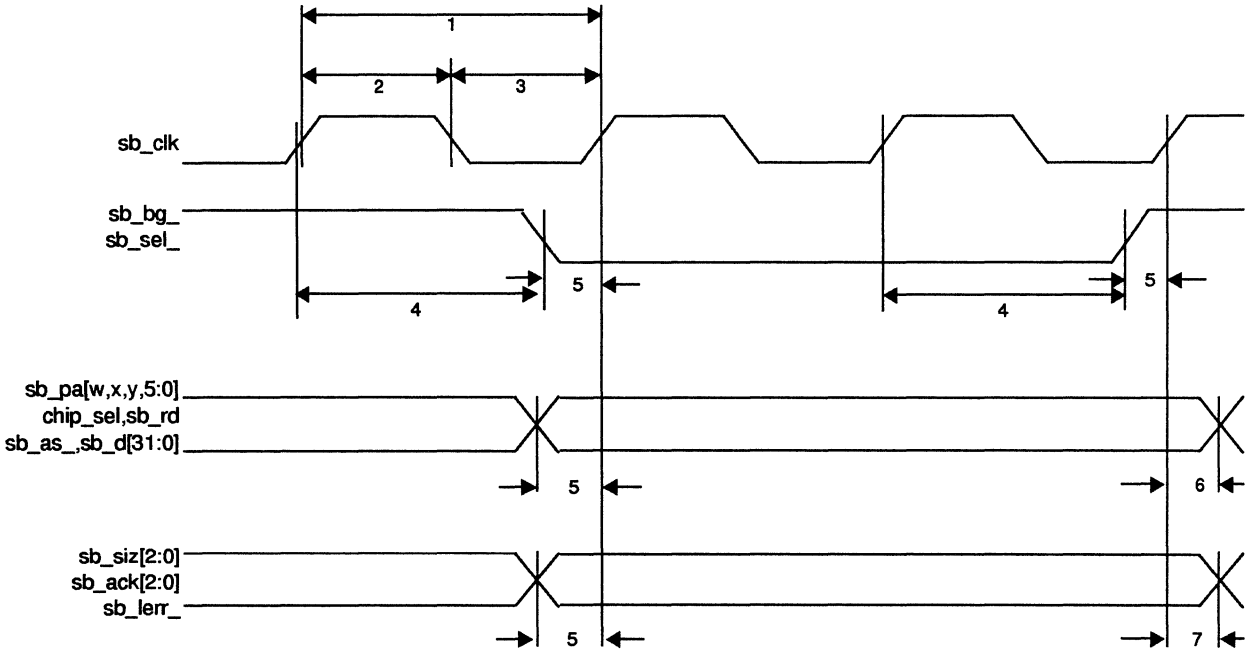


Figure 2-16 SBus Input Signals

SBus Output Signals

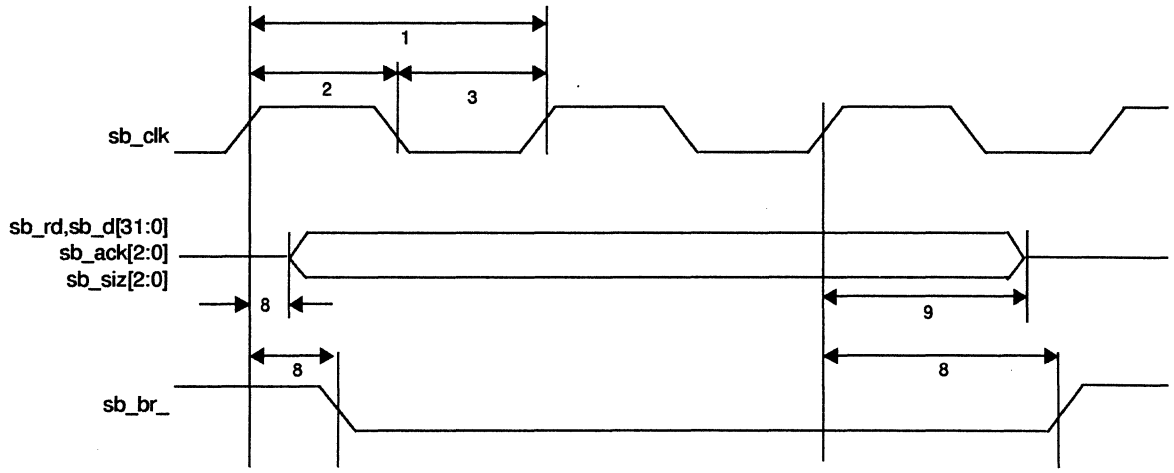


Figure 2-17 SBus Output Signal

Parallel Port Output Timing

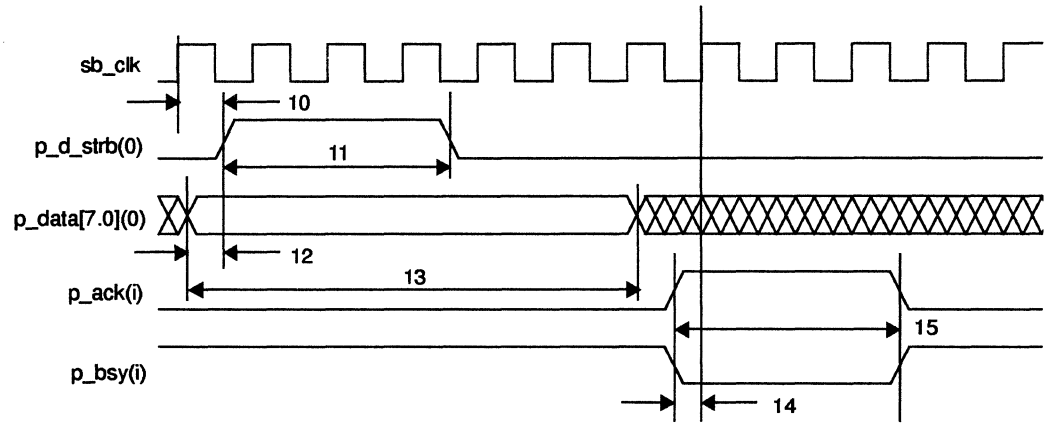


Figure 2-18 Parallel Port Output Timing

Parallel Port Input Timing

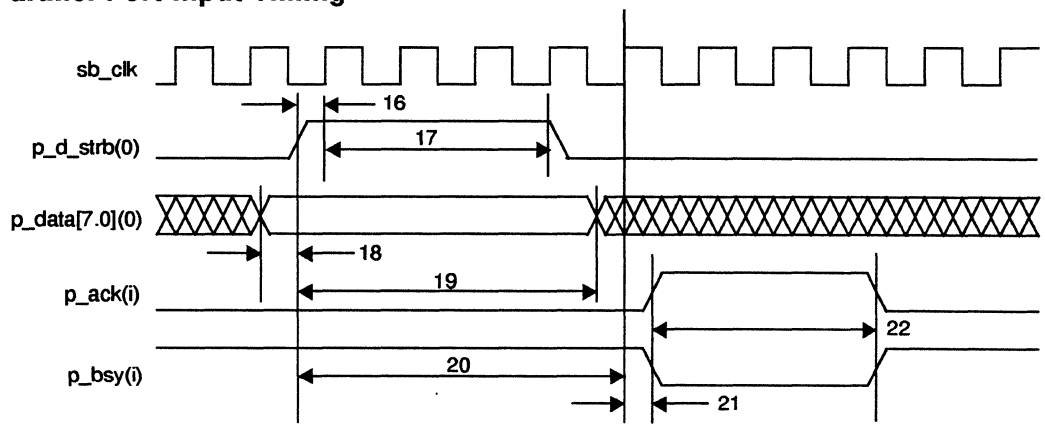


Figure 2-19 Parallel Port Input Timing

Parallel Port, Other Timing

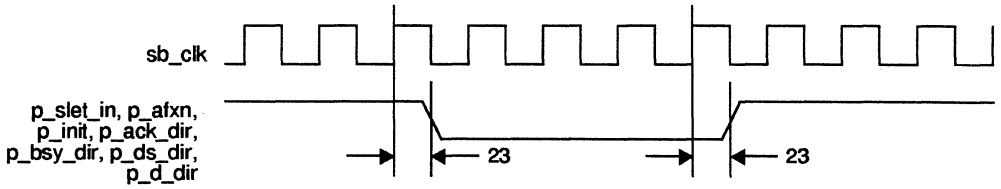


Figure 2-20 Parallel Port, Other Timing

SCSI Clock

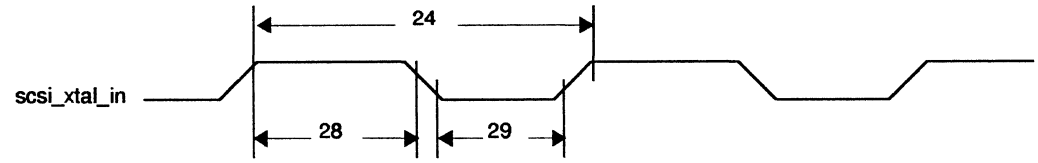


Figure 2-21 SCSI Clock

SCSI Asynchronous Output

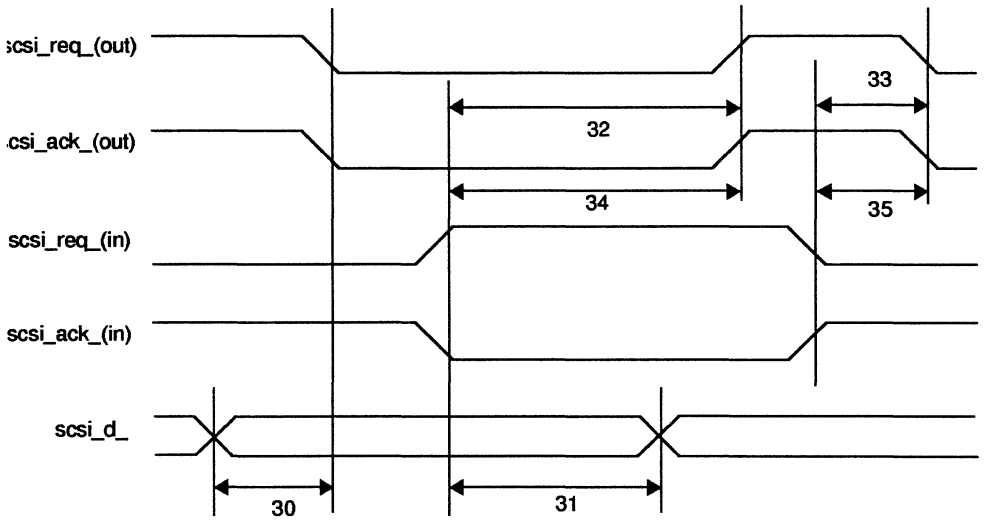


Figure 2-22 SCSI Asynchronous Output

SCSI Asynchronous Input

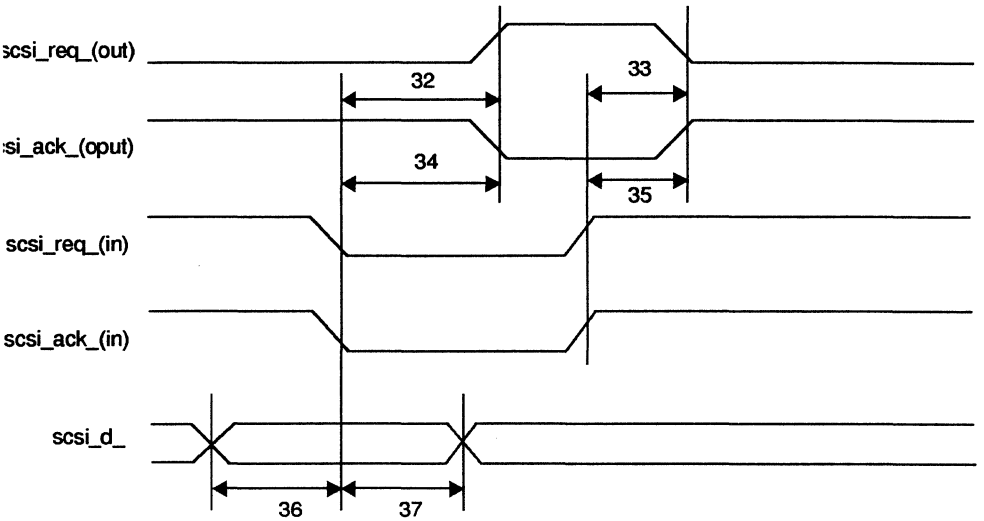


Figure 2-23 SCSI Asynchronous Input

SCSI Synchronous Output

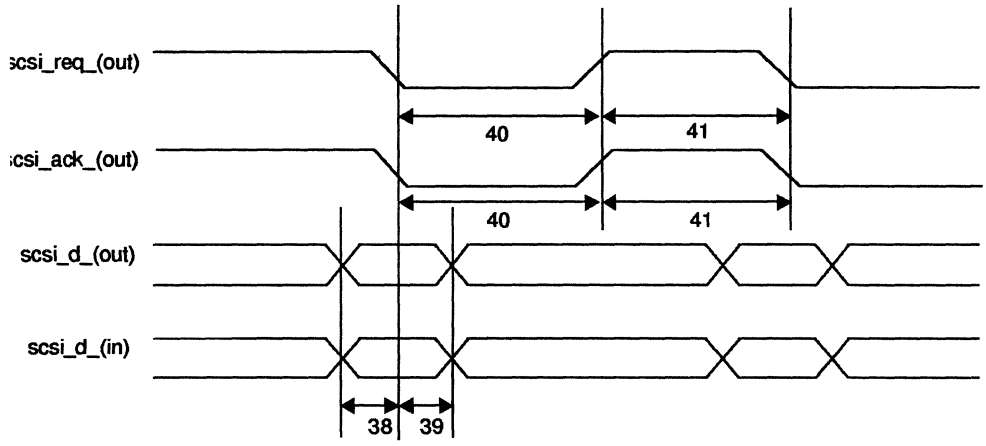


Figure 2-24 SCSI Synchronous Output

SCSI Synchronous Input

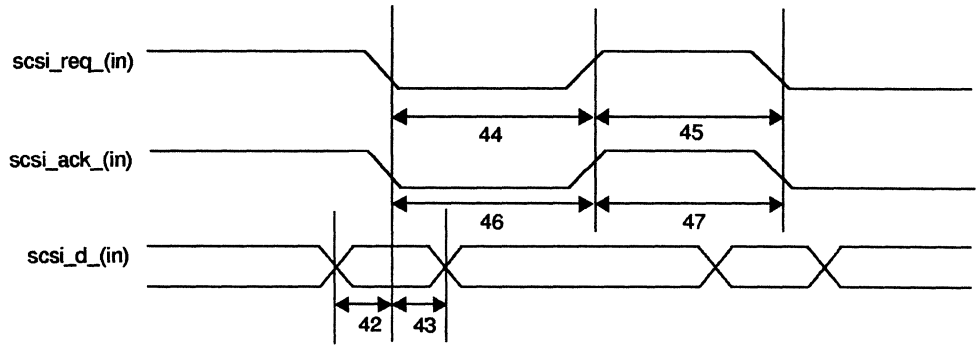


Figure 2-25 SCSI Synchronous Input

Ethernet Transmit/Receive Timing

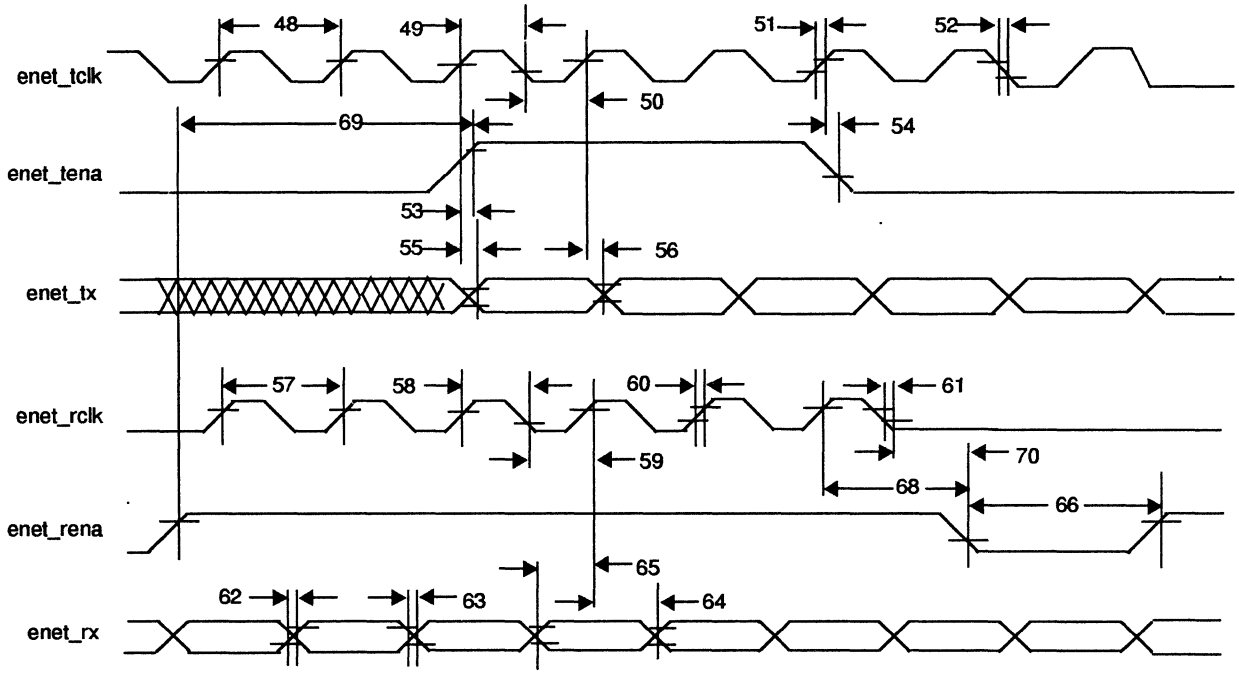


Figure 2-26 Ethernet Transmit/receive timing

Ethernet Collision Timing

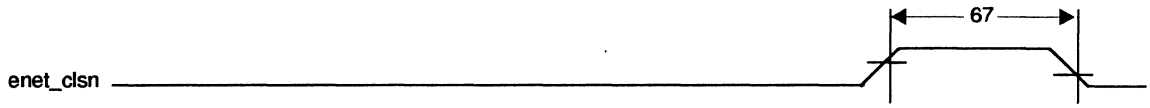


Figure 2-27 Ethernet Collision Timing

Power Consumption

The 89C100 power consumption depends on SBus clock frequency, SBus and other output loading, and the workload. Power measurements are given for maximum loading (every ASF/functional block running in a manner to maximize power consumption) of the device. These measurements were taken over the entire operating ranges of voltage and temperature. The typical number reflects the average power consumed by the device under these conditions and the maximum number reflects the high limit of power that the part may consume in operation. Note that in normal system operation, the power consumption should fall at or below the typical number given here.

Table 2-18 Power Consumption

SBus Freq	Typical power	Maximum power	Units
25 MHz	750	1400	mW

The 89C100 is packaged in a 160-pin PQFP package, and uses a custom copper leadframe to enhance thermal performance. The package thermal parameters are:

Table 2-19 Package Thermal Parameters (Still Air)

Θja	Units
24	°C/W

The 89C100 AC characteristics are given at 70°C junction temperature. By using the package characteristics and the power consumption numbers, one can get a rough idea of the allowable operating environments. Operation at junction temperatures in excess of 70°C is not recommended for performance reasons (the critical timing paths will not meet 25 MHz SBus specifications above this temperature). Operation at junction temperatures above 125°C is not recommended at any time, as it will cause reliability problems.

Packaging Information

The 89C105 chip is a standard cell design, based on the NCR VS700H technology (0.95 micron drawn, 0.7 effective).

Packaging Identification

The 89C105 is packaged in a 160 pin Plastic Quad Flat Pack (PQFP), with the following marking (the last line will be filled in with wafer lot and date code information):

Figure 2-28 Packaging Identification

Mechanical Packaging Specification

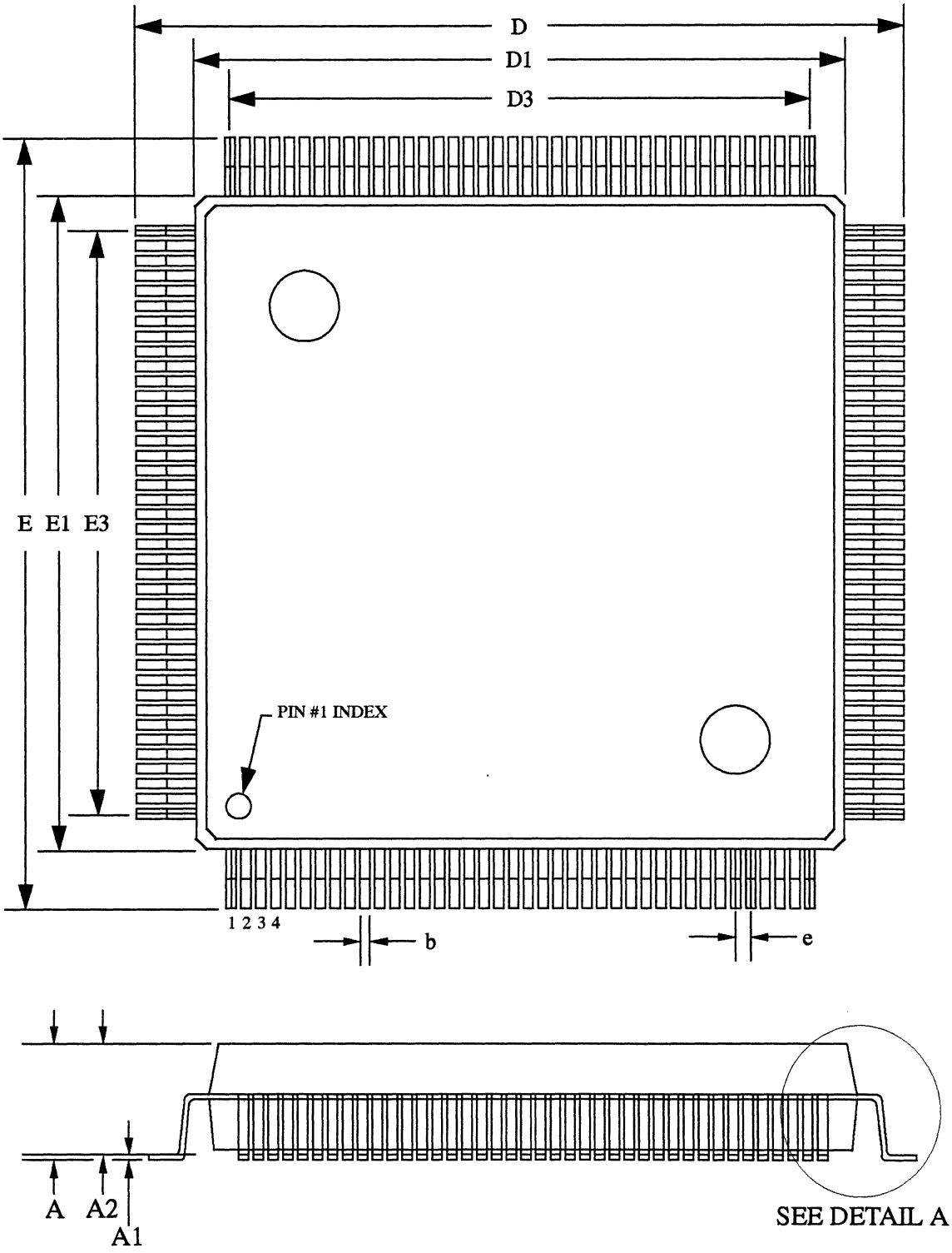


Figure 2-29 Mechanical Packaging Specification

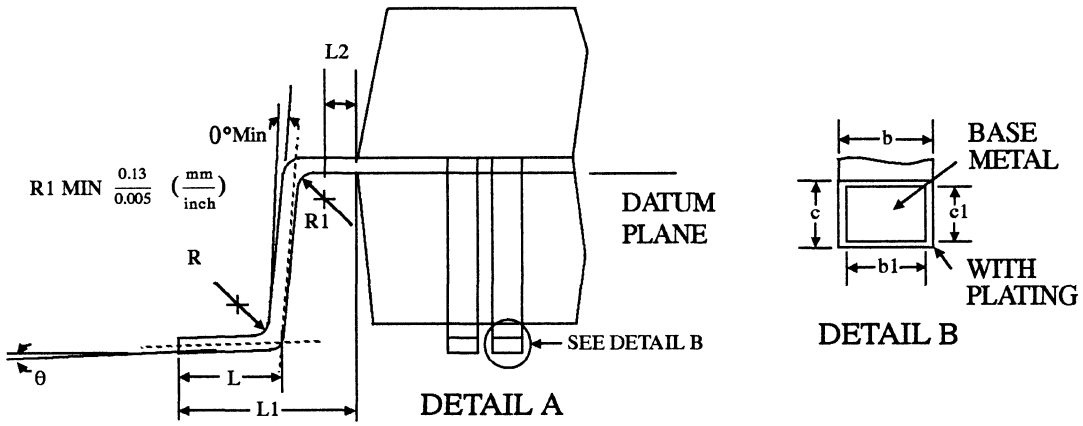


Figure 2-30 Mechanical Packaging Specification (Detail A and B)

Table 2-20 Package Measurements (mm)

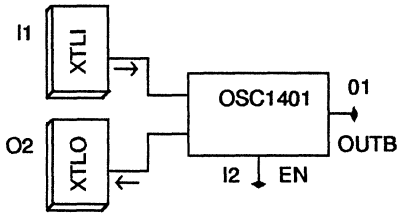
SYM	MIN.	NOM.	MAX.
A	—	—	4.07
A1	0.25	—	—
A2	3.17	3.42	3.67
D	31.65	31.90	32.15
D1	27.90	28.00	28.10
D3		25.35 Ref.	
E	31.65	31.90	32.15
E1	27.90	28.00	28.10
E3		25.35 Ref.	
L	0.65	0.80	0.95
L1		1.95 Ref.	
L2	0.40	—	—
e		0.65 BSC.	
b	0.22	—	0.38
b1	0.22	0.30	0.33
c	0.13	—	0.23
c1	0.13	—	0.17
R	0.13	—	0.40
θ	0°	—	7°

OSC1401 Crystal Oscillator

Features

- 20–50 MHz operation
- Buffered on-chip output
- Power-down mode
- gm = 72 mA/V typical

Inputs: XTLI, EN
Outputs: OUTB, XTLO
Kit Part: OSCHIP 1-A



Description

OSC1401 is a Pierce-type high frequency crystal oscillator cell designed to operate from 25 MHz to 50 MHz. It is also possible to operate the OSC1401 as a crystal oscillator down to 3 MHz. Lower frequency operation may be desired if the frequency range of the IC product extends below 25 MHz as well as between 25 and 50 MHz.

Designs using a fundamental mode crystal require two external tuning capacitors and a resistor to complete the oscillator circuit. The resistor, R1, shown in Figure 2-10will be required to provide adequate phase shift at the lower fundamental frequencies. Typical values for R1 will be 100 ohms at 25 MHz to several hundred ohms at 10 MHz.

Fundamental mode crystals are not as easy to obtain above 25 or 30 MHz so oscillators in this frequency range are often designed to use a third overtone crystal. An additional inductor and coupling capacitor are required for overtone operation. Overtone operation from 25 to 50 MHz may also require a resistor in series with C_3 to optimize performance (see the overtone circuit shown in Figure 2-11.

A power-down mode allows the oscillator to be turned off when it is not needed, to conserve power. This is especially useful in battery powered applications. The EN (ENable) pin must be high for normal operation and low for power-down mode. In power-down mode the self-bias device, MP1, is turned off and the XTLI input is pulled to ground by an open drain n-channel FET. This causes XTLO and OUTB to go to a logic 1. Note that the buffer between XTLO and OUTB is noninverting.

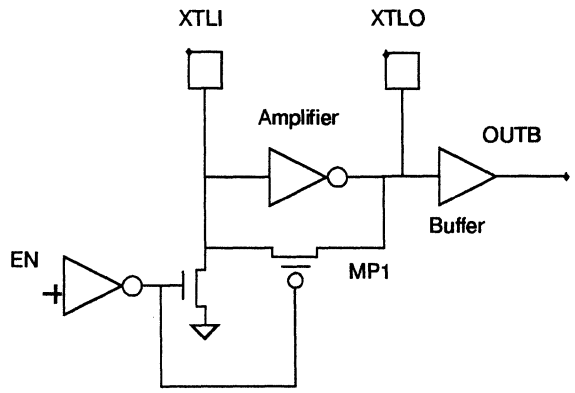


Figure 2-31 Functional Diagram

When EN is changed from low to high the oscillator may require several milliseconds to start-up and produce a stable output. The start-up time is dependent upon crystal parameters, especially the motional inductance of the crystal, L, as shown in Figure 2-12. A higher inductance value causes a slower start-up time. The analysis of startup time is beyond the scope of this data sheet but is covered in the references.

Preferred locations for OSC1401 are near the center of any side of the packaged part to minimize bond wire and lead frame parasitics. This is especially important in DIP packages. It is also desirable to place power (V_{DD}) and ground (V_{SS}) pins near the oscillator cell. A short, wide circuit board ground trace must be used from the V_{SS} pin(s) to the external tuning capacitors, C_1 and C_2. It is also important to provide a 0.01 µF or 0.1 µF bypass capacitor from V_{DD} to V_{SS} (very close to the pins) for proper oscillator operation.

Driving External Circuits

To drive external circuits with an oscillator generated clock, the on-chip oscillator output (OUTB) should drive a pad buffer such as OUTINV, which in turn drives an output pad such as an OPD8. A special output pad cell, the OPD16SYM, may be driven directly by the oscillator OUTB port without a buffer. This method provides the best possible waveform symmetry for driving external CMOS level circuits since the OPD16SYM has symmetrical low-high and high-low output drive. The oscillator output, XTLO, should not normally be used to drive external circuits except the oscillator components. To avoid undesired parasitic feedback paths, separate power and ground pins must be used to isolate the oscillator cell when it drives any pad cell, with or without an intermediate buffer.

Driving the OSC1401

In some cases, the user may want the option of driving the OSC1401 input with an external clock source rather than operating the cell as an oscillator. This may be the case if a clock source is available on the PC board of some products and not on others. The XTLI input can be driven with an external source provided XTLO is left unconnected. The maximum toggle frequency is typically 50 MHz with no load on XTLO. The signal driving XTLI must be from a CMOS driver such as standard 54/74HCxxx IC's or some other rail-to-rail driver. A TTL driver does not provide an adequate HI output level to drive the OSC1401 at high frequencies. Any application of the oscillator cell in which the input is driven should be first reviewed with an NCR applications engineer.

Theory and External Components

The internal bias device, MP1, causes the inverter to self bias to approximately $V_{DD}/2$. This is an operating point where the inverting amplifier has high gain, which is necessary for the circuit to oscillate. The on-resistance of MP1 is approximately 5 MΩ such that it will not affect the AC performance of the circuit.

Figures 2-11 and 2-12 show the external components required for fundamental and overtone operation. In Figure external components C_1, C_2 and the crystal, form a pi network which resonates at the specified crystal frequency. The ratio C_2/C_1 should be somewhat greater than unity since it is a term in the loop gain equation. Increasing the ratio too much will cause the voltage swing on XTLI to exceed the supply rails which is undesirable. Typically, a range of $1.1 < (C_2/C_1) < 1.5$ should be used.

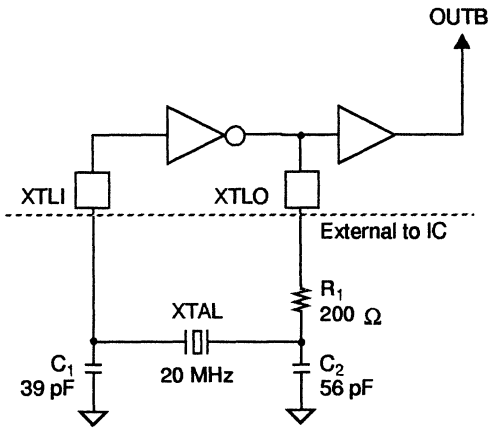


Figure 2-32 Typical Fundamental Mode Circuit

The series combination of C_2 and C_1 should also be approximately equal to the load capacitance specified for the quartz crystal. The strays associated with each node, and the oscillator input and output capacitance should be included in calculations which involve C_2 and C_1. Typical component values are shown in Figures 2-12 and 2-13.

The output resistance of the oscillator core, along with C_2, forms an RC low pass circuit. This pole contributes additional phase shift to insure that the phase shift around the loop is greater than 360 degrees, which is required for oscillation. In some cases the addition of a resistor in series with the output will improve performance and reduce the power supply current.

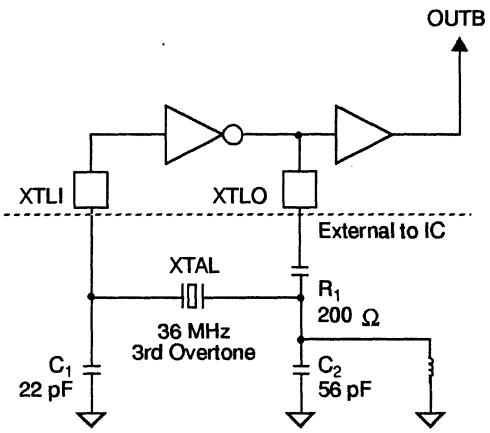


Figure 2-33 Typical Third Overtone Circuit

To achieve overtone oscillation, the crystal's fundamental mode must be suppressed by making the loop gain lower at the fundamental frequency compared to the third overtone frequency. An overtone crystal will resonate at its fundamental frequency unless the loop gain is forced to be higher at the overtone frequency. Loop gain must be less than one at the fundamental and greater than one at the desired overtone to insure startup on the overtone frequency.

Referring to Figure 2-11, the additional components necessary for overtone operation are a coupling capacitor and an inductor. The coupling capacitor is a DC block so the inductor does not short the inverter output to ground. In some cases a resistor in the range of 10-100 ohms in series with C_3 will improve the duty cycle and minimize supply current. C_3 is chosen such that its impedance is low compared to other components at the operating frequency. Values between 0.0015 µF and 0.01 µF are a good choice.

The inductor, L_1, is selected such that its impedance lowers the loop gain at the fundamental frequency relative to the third harmonic. The resonant frequency of the circuit made up of C_2 and L_1 is set midway between the fundamental and third overtone frequency. This causes the equivalent impedance to look inductive at the fundamental and capacitive at the third overtone frequency. The equivalent capacitance of C_2 and L_1 in parallel at the third overtone is used to calculate C_2 (effective) at the output node.

The value of C_1 may be reduced to as low as 10 pF since the input referred Miller capacitance actually increases the effective input capacitance beyond the specified values. Some experimentation with component values should be anticipated prior to specifying final production values. Refer to Table 2-23 for typical external component values.

Table 2-21 Typical Overtone Component Values

Third Overtone Frequency	C_1	C_2	L_1
25 MHz	33 pF	120 pF	0.68 µ
35 MHz	22 pF	100 pF	0.47 µH
50 MHz	22 pF	100 pF	0.22 µH
Note: C1 = 0.01 µF			

Crystals

The OSC1401 is a Pierce type oscillator circuit in which the crystal is operated in its parallel resonant mode. (Refer to the crystal equivalent circuit - Figure 2-12.) At parallel resonance, the LRC leg appears slightly inductive and resonates with C_0 and the circuit load capacitances, C_1 and C_2. Typical equivalent circuit component values are shown in Table 2-20. The values should only be used as a guideline in selecting a crystal for your application. The R_{MAX} column, however, should be adhered to when specifying a crystal. It will ensure a quality crystal which will resonate in a circuit using the OSC1401 cell.

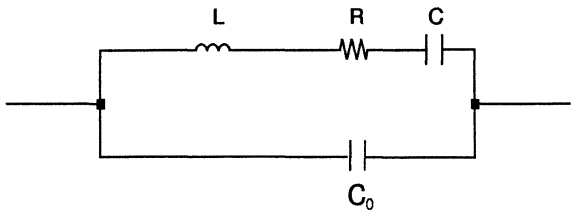


Figure 2-34 Crystal Equivalent Circuit

Simulation and Test

For simulation purposes, the XTLI to XTLO propagation delay is zero. In production test, the XTLO pin will be driven with the complement of the XTLI test signal. This is transparent to the designer.

The most thorough method of testing an oscillator circuit is to determine phase margin and gain by breaking the loop and making measurements with a network analyzer. A very basic test is observing the output waveform at XTLO on an oscilloscope when Vdd is first turned on, and in steady state. A FET probe should be used to keep capacitive loading to less than 2 pF. A standard 10X probe has an input capacitance of approximately 13 pF which will change the characteristics of the oscillator loop. A buffered output, which is available on the kit part, should be monitored if a standard probe is to be used.

Table 2-22 Typical Measured Crystal Parameters (Represents Several Manufacturers' AT Cut Crystals)

Crystal Frequency	L	C	C0	R	R_{MAX}
10 MHz	10.64 mH	23.83 fF	5.998 pF	14 Ω	30 Ω
20 MHz	3.042 mH	20.81 fF	5.310 pF	6.7 Ω	25Ω
24 MHz	25.005 mH	1.762 fF	4.095 pF	19 Ω	50Ω
36 MHz	13.321 mH	1.467 fF	6.88 pF	32 Ω	50Ω
48 MHz	6/997 mH	1.571 fF	6.407 pF	23 Ω	50Ω

Specifications

Table 2-23 Electrical Specifications

Parameter	Min	Typical @ 25 °C	Max	Unit	Operating Temp. Range °C	Comments/ Conditions
Power supply range	4.5		5.5	V	-55 to 125	
Amplifier transconductance (gm)	49 47 38 38	72	106 120 120 125	mmhos (mA/V)	25 0 to 70 -40 to 85 -40 to 125 -55 to 125	 V_{DD} = 5.0 ± 10% Note 4

Note 1 - Some experimentation with external component values should be anticipated prior to specifying production values to achieve the duty cycle specifications shown.

Note 2 - DC bias voltage of core inverter input and output (XTLI and XTLO) with no crystal in the circuit.

Note 3 - Start-up time is dependent on the external circuit and the specific crystal being used. Start-up time is defined as the time required for the envelope of the XTLO output to reach 90% of final amplitude. This insures that OUTB is a valid clock.

Note 4 - At operating temperatures above 85 °C the load capacitance (CL) must be reduced below 32 pF for adequate gm margin at higher frequencies.

Table 2-23 Electrical Specifications (Continued)

Parameter	Min	Typical @ 25 °C	Max	Unit	Operating Temp. Range °C	Comments/ Conditions
Input capacitance (CXTLI)		11.9			25	
	8.3		16.0		0 to 70	
	8.3		16.0	pF	-40 to 85	V_{DD} = 5.0 ± 10%
	8.3		16.0		-40 to 125	
	8.3		16.0		-55 to 125	
Output capacitance (CXTLO)		6.1			25	
	4.5		8.1		0 to 70	
	4.5		8.1	pF	-40 to 85	V_{DD} = 5.0 ± 10%
	4.5		8.1		-40 to 125	
	4.5		8.1		-55 to 125	
Duty cycle		50		%	25	
	45		55		0 to 70	At OUTB output
	45		55		-40 to 85	Note 1
	45		56		-40 to 125	
	45		57		-55 to 125	
Self bias voltage		2.50		V	25	V_{DD} = 5.0 V Note 2
Startup time		2.0		mS	25	Note 3
Power-down (EN=0) supply current				µA		

Note 1 - Some experimentation with external component values should be anticipated prior to specifying production values to achieve the duty cycle specifications shown.

Note 2 - DC bias voltage of core inverter input and output (XTLI and XTLO) with no crystal in the circuit.

Note 3 - Start-up time is dependent on the external circuit and the specific crystal being used. Start-up time is defined as the time required for the envelope of the XTLO output to reach 90% of final amplitude. This insures that OUTB is a valid clock.

Note 4 - At operating temperatures above 85 °C the load capacitance (CL) must be reduced below 32 pF for adequate gm margin at higher frequencies.

Table 2-24 Power Supply Current - Overtone Mode

Supply Current	Operating Temp. °C
I_{DD} = 11.0 mA + (0.1 mA/MHz) x f (typical)	25
I_{DD} = 19.0 mA + (0.1 mA/MHz) x f (maximum)	0 to 70
I_{DD} = 20.0 mA + (0.1 mA/MHz) x f (maximum)	-40 to 85
I_{DD} = 20.0 mA + (0.1 mA/MHz) x f (maximum)	-40 to 125
I_{DD} = 20.5 mA + (0.1 mA/MHz) x f (maximum)	-55 to 125
Note: f = frequency in MHz (between 25 - 50 MHz)	

Table 2-25 Power Supply Current - Fundamental Mode

Supply Current	Operating Temp. °C
I_{DD} = 6.0 mA + (0.2 mA/MHz) x f (typical)	25
I_{DD} = 11.0 mA + (0.2 mA/MHz) x f (maximum)	0 to 70
I_{DD} = 12.0 mA + (0.2 mA/MHz) x f (maximum)	-40 to 85
I_{DD} = 12.0 mA + (0.2 mA/MHz) x f (maximum)	-40 to 125
I_{DD} = 12.5 mA + (0.2 mA/MHz) x f (maximum)	-55 to 125
Note: f = frequency in MHz (between 25 - 35 MHz) with a fundamental mode crystal.	

DMA2 DMA Core
Contents

DMA2
DMA Core

The DMA2 core is functionally equivalent to the LSI L64854 DMA controller and provides three channels for DMA over the SBus. It has three external interfaces designed to provide DMA access to one NCR92C990 Ethernet Controller (compatible with AMD7990), one NCR53C9X SCSI Controller[1] (SCSI), and one programmable parallel port (Centronics®-type). The DMA2 contains internal buffering for these DMA channels in the form of a cache for the ENET interface and FIFOs for the SCSI and parallel port interfaces. It also contains control/status registers for each of the three interfaces, several SCSI and parallel port support registers, and a write buffer for slave accesses to the ENET. The design of the DMA2 is based on those of the SBUS DMA+ controller chip (LSI L645853A) and the parallel port from the SBus Hardcopy I/O Device (HIOD chip), and incorporates a number of new features for increasing performance and allowing different modes of operation necessary for new systems. By default, DMA2 is software compatible with DMA+. Software enables DMA2's additional features by changing the default values of appropriate mode bits in the control/status registers.

Features

- Supports 10 MByte/sec SCSI transfers, 1.25 MByte/sec Ethernet transfers, and 4 MByte/sec parallel port transfers.
- Supports 4-word, 8-word, and 'no burst' SBus burst modes.
- 64-byte internal cache for Ethernet data buffering.
- 64-byte internal FIFOs for SCSI and parallel port data buffering.
- 16-bit write buffer for slave writes to ENET.
- Improved cache and FIFO draining algorithms for better SBus utilization.
- Internal address and byte count registers and "NEXT" address/byte count features for data block chaining on SCSI and parallel port interfaces.
- JTAG test interface.

1. The NCR53C9X is identical to the FAS101 licensed from Emulex Corporation.

Pinout Information

SBus Interface
57 Signals

Name	Direction	Description
sb_d(31:0)	I/O	SBus Data Bus
sb_br_	O	SBus Bus Request
sb_bg_	I	SBus Bus Grant
sb_ack(2:0)	I/O	SBus Acknowledge
sb_reset_	I	SBus Reset
sb_lerr_	I	SBus Late Error (INT15)
sb_clk	I	SBus Clock input
sb_rd	I/O	SBus Read/Write_
sb_sel_	I	SBus Select
sb_d_irq_	O	SBus Interrupt for SCSI transfers
sb_e_irq_	O	SBus Interrupt for ENET transfers
sb_p_irq_	O	SBus Interrupt for Parallel Port transfers
sb_siz(2:0)	I/O	SBus transfer Size
sb_as_	I	SBus Address Strobe (addr is valid)
sb_a[x][1]	I	High order physical address bits (for slave decodes)
sb_a[y][1]	I	High order physical address bits (for slave decodes)
sb_a[z][1]	I	High order physical address bits (for slave decodes)
sb_a(4:0)	I	Low order physical address bits

1. These signals, sb_a (x, y, z), correspond to 89C100 signals sb_pa (x, y, w), respectively.

Lance Interface
35 Signals

Name	Direction	Description
e_as_	I	ENET Address Strobe
e_hold_	I	ENET Hold
e_hlda_	O	ENET Hold Acknowledge
e_read	I/O	ENET Read
e_das_	I/O	ENET Data Strobe
e_rdy_	I/O	ENET Ready
e_cs_	O	ENET Chip Select
e_irq_	I	ENET Interrupt Request
e_reset_	O	ENET Reset
e_byte	I	ENET Byte Marker
e_a(23:16)	I	ENET High Order Address
e_ad(15:0)	I/O	ENET Address/Data Bus
e_sb_al	O	Buffered Version of sb_a(1) for ENET

SCSI Interface
16 Signals

Name	Direction	Description
d_d(7:0)	I/O	SCSI Data Bus
d_req	I	SCSI DMA Request
d_ack_	O	SCSI DMA Acknowledge
d_rd_	O	SCSI Read Strobe (reg read or dma data to memory)
d_wr_	O	SCSI Write Strobe (reg write or dma from memory)
d_cs_	O	SCSI Chip Select for slave register access.
d_irq_	I	SCSI Interrupt Request
d_reset	O	SCSI Reset
id_cs_	O	Secondary Device Select (boot prom) output. Pull high to specify existence of external prom

Parallel Port Interface
21 Signals

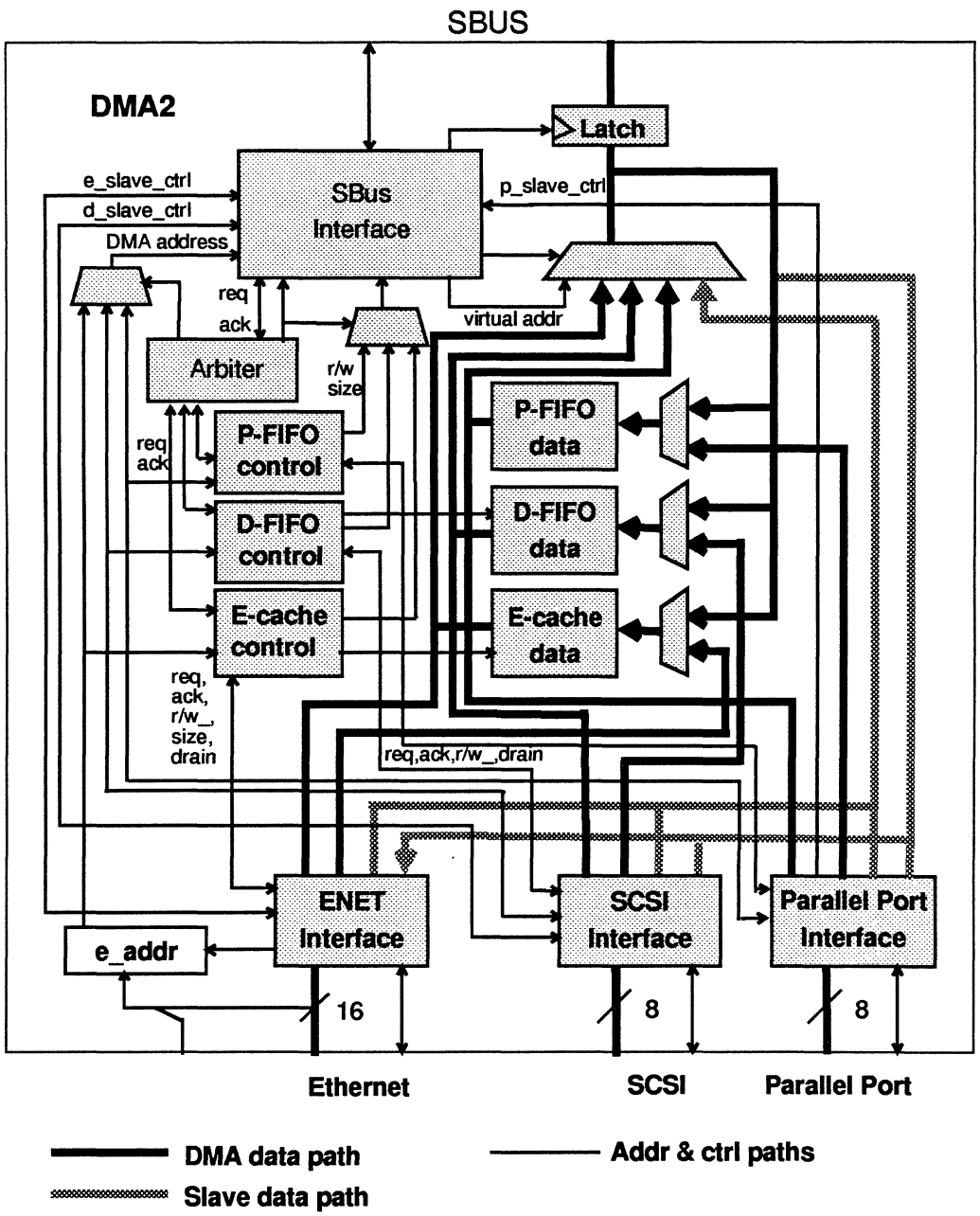
Name	Direction	Description
p_data(7:0)	I/O	Parallel Port Data Bus
p_d_strb	I/O	Parallel Port Data Strobe
p_bsy_	I/O	Parallel Port Busy
p_ack	I/O	Parallel Port Acknowledge
p_pe_	I/O	Parallel Port Paper Error
p_slct_	I/O	Parallel Port Select
p_error	I	Parallel Port Error
p_init	O	Parallel Port Initialize
p_slct_in	O	Parallel Port Select In
p_afxn	O	Parallel Port Auto Feed
p_ds_dir_[1]	O	Parallel Port Data Strobe Direction
p_bsy_dir_[1]	O	Parallel Port Busy Direction
p_ack_dir_[1]	O	Parallel Port Acknowledge Direction
p_d_dir_[1]	O	Parallel Port Data Direction

1. The Parallel port direction bits, p_ds_dir_, p_bsy_dir_, p_ack_dir_, and p_d_dir_, have the following polarity: asserted (p_*_dir_=0) sets transfer direction toward DMA2, unasserted (p_*_dir_=1) sets transfer direction away from DMA2.

Miscellaneous
6 Signals

Name	Direction	Description
jtagdi	I	JTAG data input
jtagclk	I	JTAG clock
jtagtms	I	JTAG test mode select
jtagdo	O	JTAG data output
slow_	I	Fast or slow DMA acknowledge cycles (for use with SCSI controller fast mode should be used - internal pullup means this requires no external connection for fast mode).
tp_aui_	O	Ethernet select output. Drives NIS input of T7213 chip to select between twisted pair and AUI type Ethernet interfaces (See definition of E_CSR(22)).

Chip Level Block Diagram



Address Maps

DMA2 Internal Registers

Table 3-1 DMA Internal Registers

sb_a(x, y, z)[1]	sb_a(4:0)[2]	Register Accessed	Type	Size
000	00000	Internal ID Register[3]	R	32
010	00000	SCSI Control/Status Register (D_CSR)	R/W	32
010	00100	SCSI Address Register (D_ADDR)[4]	R/W	32
010	01000	SCSI Byte Count Register (D_BCNT)[4]	R/W	24
010	01100	SCSI Test Control/Status Register (D_TST_CSR)	R/W	32
010	10000	Ethernet Control/Status Register (E_CSR)	R/W	32
010	10100	Ethernet Test Control/Status Register (E_TST_CSR)	R/W	32
010	11000	Ethernet Cache Valid Bits (E_VLD)	R/W	32
010	11100	Ethernet Base Address Register (E_BASE_ADDR)	R/W	8
101	00000	P-Port DMA Control/Status Register (P_CSR)	R/W	32
101	00100	P-Port Address Register (P_ADDR)	R/W	32
101	01000	P-Port Byte Count Register (P_BCNT)	R/W	32
101	01100	P-Port Test Control/Status Register (P_TST_CSR)	R/W	32
101	10000	P-Port Hardware Configuration Reg (P_HCR)	R/W	16
101	10010	P-Port Operation Configuration Reg (P_OCR)	R/W	16
101	10100	P-Port Parallel Data Register (P_DR)	R/W	8
101	10101	P-Port Transfer Control Register (P_TCR)	R/W	8
101	10110	P-Port Output Register (P_OR)	R/W	8
101	10111	P-Port Input Register (P_IR)	R/W	8
101	11000	P-Port Interrupt Control Register (P_ICR)	R/W	16

1. sb_a(x, y, z) = sb_a(23:22:26) on typical systems.
2. It is recommended that software access these registers with sb_a(5) = 0 for future expandability.
3. Byte and 1/2 word accesses to this register are also allowed.
4. The "NEXT" Address/Byte Count registers are accessed at these addresses using the D_EN_NEXT bit in the D_CSR.

SCSI Registers (on 53C9X ASF)

Table 3-2 SCSI Registers (on 53C9X Chip)

sb_a(x, y, z)	sb_a(5:2)	Register Accessed	Type	Size
100	0000	Transfer Count Low	R/W	8
100	0001	Transfer Count High	R/W	8
100	0010	FIFO Data	R/W	8
100	0011	Command	R/W	8
100	0100	Status/Bus ID	R/W	8
100	0101	Interrupt/Status Time-out	R/W	8
100	0110	Seq. Step/Synch transfer period	R/W	8
100	0111	FIFO flags/Synch offset	R/W	8
100	1000	Configuration	R/W	8
100	1001	Clock Conversion Factor	R/W	8
100	1010	SCSI TEST (chip test use only)	R/W	8
100	1011	Configuration (SCSI-2 only)	R/W	8

ENET Registers (on NCR92C990 ASF)

Table 3-3 ENET Registers (on NCR92C990 Chip)

sb_a(x, y, z)	sb_a(1)	Register Accessed	Type	Size
110	0	Register Data Port (RDP)	R/W	16
110	1	Register Address Port (RAP)	R/W	16

SBus Interface

The DMA2 provides for the communication between the SBus and three I/O devices, each of which are connected to separate ports of the chip. One I/O device port is tailored to interface with the NCR92C990 Ethernet Controller, one port is tailored to interface with the NCR53C90X SCSI Controller, and one port implements a programmable Centronics-type parallel port. However, it is possible to interface with other I/O devices with some external logic. In this document, the three ports will be referred to as the ENET interface, the SCSI interface, and the parallel port interface, and it will be assumed that the ENET and SCSI are connected to their respective interfaces. Also, the three buffers associated with these interfaces will be referred to as the E-cache, the D-FIFO, and the P-FIFO, respectively.

The DMA2 provides both Slave cycle and Master cycle accesses over the SBus. Address and Control registers in the DMA2 gate array, along with similar registers in both the ENET and the SCSI (or other devices in their places), can be accessed by the CPU via Slave cycles.

Master Cycles

To perform a Master Cycle the DMA2 requests the use of the SBus by asserting sb_br_, the bus request signal. On reception of sb_bg_, bus grant, the DMA2 takes control of the SBus to transfer data to or from memory, using the signals sb_d(31:0), sb_read, sb_siz(2:0), and sb_ack(2:0). The address is multiplexed onto the sb_d bus for all accesses on the SBus.

The sizes of DMA reads from memory to the E-cache, D-FIFO, and P-FIFO are determined by the E_BURST_SIZE field in the E_CSR, the D_BURST_SIZE field in the D_CSR, and the P_BURST_SIZE field in the P_ CSR, respectively. All reads to a given interface will be of the same size, either 1, 4, or 8 SBus words, depending on the value of the BURST_SIZE field in that interface's CSR. DMA writes from one of the interface buffers to memory can be any of the following sizes not greater than the corresponding read burst size: byte, half-word, word, 4-word, or 8-word. No 2-word bursts are carried out. The DMA2 SBus interface will always use the largest possible size when writing to memory. The largest possible size is determined by which bytes have been written to the DMA2 by the I/O device and the BURST_SIZE field in that device's CSR. Note that the slave device accessed by DMA2 must support the transfer sizes for which the chip is configured.

The SBus interface of the DMA2 is capable of supporting rerun acknowledgments from slaves. The requested DMA transfer will repeat until either it completes or an error indicator (sb_ack(0) asserted) is received, whereupon it will be aborted. When the DMA2 receives a rerun acknowledge, it will unassert sb_br_ for one clock to give the CPU and other DMA masters a chance to win SBus arbitration. Then it will reassert sb_br_ to retry the cycle. Once the DMA2 has asserted sb_br_, it will not unassert it until it has received sb_bg_ or sb_reset_.

Slave Cycles

The CPU accesses registers in the DMA2 chip and attached devices by the use of a geographical select signal on the SBus, sb_sel_, in conjunction with sb_as_. During slave cycles the DMA2 takes control of the sb_ack(2:0) signals. The combination of responses are as follows:

sb_ack(2)	sb_ack(1)	sb_ack(0)	Definition
1	1	1	Insert Wait States **
1	1	0	Error ack **
1	0	1	8-bit Port ack **
1	0	0	Rerun **
0	1	0	Reserved
0	0	1	16-bit Port ack **
0	1	1	32-bit Port ack **
0	0	0	Reserved
NOTE: This table represents all possible SBus responses. The DMA2 can, however, only generate those responses marked with a **			

The DMA2 will give a rerun acknowledgment if a slave access is to an I/O device (SCSI or ENET) that is currently active with a DMA transfer, with one exception. The ENET interface is considered active with DMA if it is currently asserting e_hlda_ or has asserted e_hlda_ within the one SBus clock. The SCSI interface is considered active with DMA if it is asserting d_ack_ or has asserted d_ack_ within two SBus clocks.

The DMA2 contains a 16-bit write buffer for slave writes to ENET. This write buffer is enabled by default, but can be disabled by setting the E_DSBL_BUF_WR bit in the E_CSR. When E_DSBL_BUF_WR is not set, a single slave write to the ENET can be carried out without a rerun while it is active with a DMA transfer. A second slave write to the ENET while it is still active with a DMA transfer will elicit a rerun acknowledgment. The DMA2 will not give a rerun if a slave access is to an internal register of the DMA2; these accesses can take place concurrently with DMA activity between the I/O devices and the DMA2. Forcing the SBus master (usually the CPU) to rerun its slave accesses in this manner will insure that a deadlock does not occur between the SCSI or ENET and the SBus master.

If the slave access to the DMA2 does not cause a rerun acknowledge, then the DMA2 will respond in the following way:

Acknowledge	Field accessed
32-bit port ack	Internal ID or internal register
16-port ack	ENET register or 16-bit parallel port register
8-bit port ack	SCSI register, external ID, or 8-bit parallel port register
error ack	A byte or half-word access to an on-chip register or a byte or word access to a ENET register. Note that an error ack will also be given for a register access of some size other than word, half-word, or byte.

SBus Identification

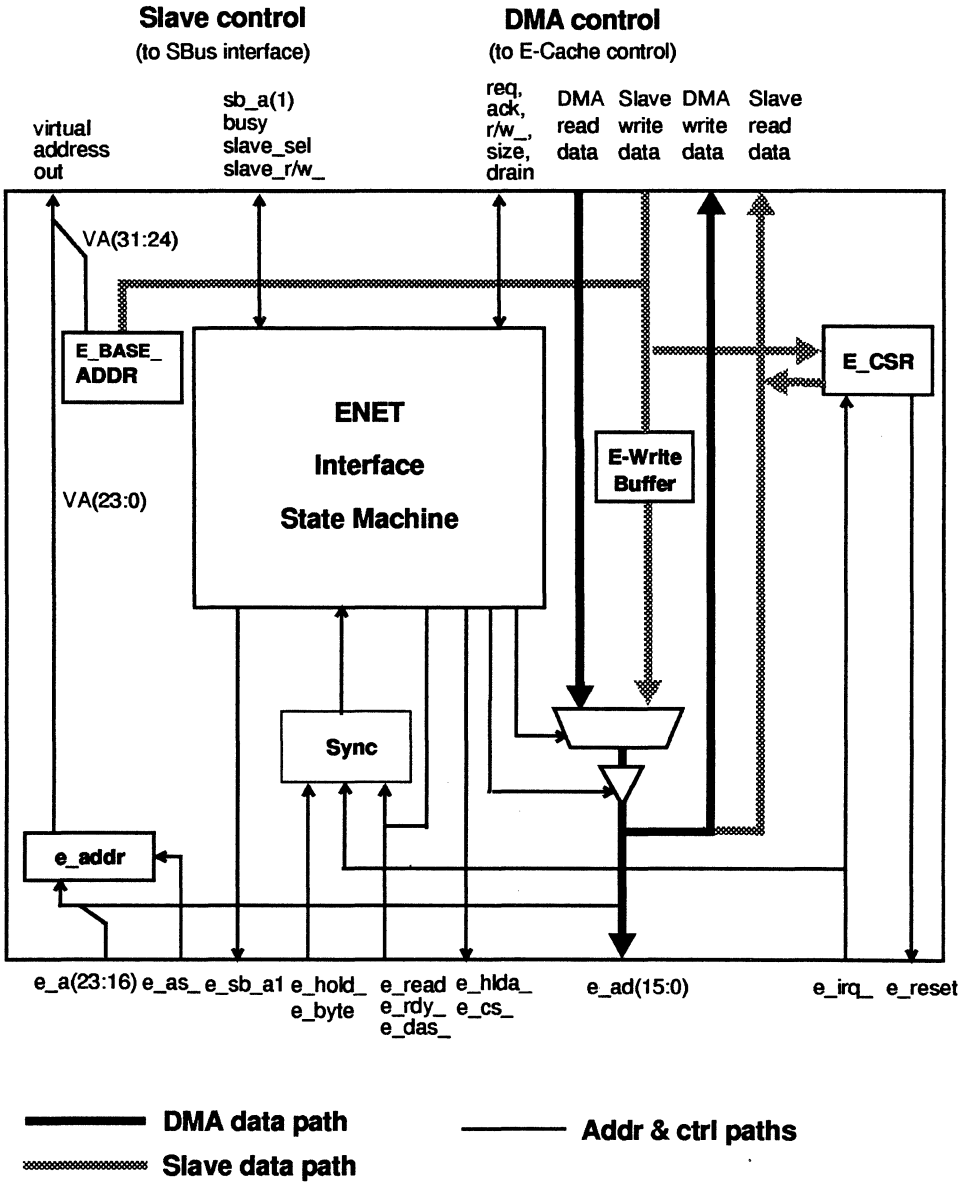
This is a mechanism which allows software to uniquely identify each SBus device, since each device can have a unique ID. The onboard ID of the DMA2 is hardwired to the 32-bit value 0xfe810103. This value will be returned when the ID field is read by the CPU if the id_cs_ pin is tied low. If the id_cs_ pin is pulled high through a large (~4.7k) resistor, then access to the ID field will cause an external access using the id_cs_ pin as an external chip select. For offboard ID reads, ID data should be driven onto p_data(7:0) when id_cs_ is asserted. The sb_a bits can be used to address the external ID PROM.

NOTE: For offboard ID reads (external to the DMA2), id_cs_ is simply an address decode from sb_a(x, y, z); therefore, it can be active for a slave read cycle or a slave write cycle.

The 89C100 provides an internal pullup on id_cs_, thus an external pull-up resistor is not required.

ENET Interface

Block Diagram



Overview

The DMA2 will provide all necessary buffering and arbitration functions to allow the ENET to access main memory over the SBus (master cycle) and the CPU to access the ENET (slave cycle). Between the ENET interface and the SBus interface is a 2 line, 8-word/line cache (E-cache) with consistency control logic. Each byte in the E-cache has its own valid/dirty bit, and each line has a bit that determines the meaning of the valid/dirty bits for that line (i.e. valid or dirty). These valid/dirty bits can be accessed directly for testing via the E_VLD register. Bytes being transferred to memory are marked dirty as they are loaded into the E-cache. Bytes being transferred from memory are marked as valid as they are loaded into the E-cache. Consistency control ensures that all data written by the ENET gets to main memory in a deterministic manner, and is handled completely in hardware.

The ENET uses multiplexed address and data, so the DMA2 demultiplexes them internally. The address supported by the ENET is 24 bits. The upper 8 bits of the 32-bit virtual address for an ENET transfer come from the E_BASE_ADDR register, and by default are driven to 0xFF. The ENET has a 16-bit data path which can accommodate 8 or 16 bit accesses using byte masking capabilities.

Data packing and unpacking is used to reduce the bus bandwidth impact of ENET transfers on the SBus. The ENET performs 2 distinct types of memory access; data transfer from or to the network, and message descriptor access for buffer management control. The ENET performs data transfers in bursts of up to 8 half-word accesses (16 bytes total), and message descriptor transfers as single half-word accesses. The end of a ENET transfer is detected by the unassertion of the e_hold_ signal.

NOTE: Any I/O device connected to the ENET interface of the DMA2 is required to unassert e_hold_ between write cycles and read cycles, and it is not allowed to unassert e_hold_ during the e_as_ or e_das_ pulses, or between these pulses.

Transfers To Memory

Data from the ENET is packed into the E-cache and marked as "dirty" until one of the draining conditions is detected. The nine conditions that can cause draining of dirty bytes to memory are as follows:

1. The ENET writes to the last byte of a E-cache line.
2. The ENET writes a message descriptor (disabled by setting E_DSBL_WR_DRN bit in E_CSR).
3. The CPU does a slave read of a ENET CSR (disabled by setting E_DSBL_RD_DRN bit in E_CSR).
4. The CPU does a slave write to a ENET CSR (disabled by setting E_DSBL_WR_INVAL bit in E_CSR).
5. The E_DRAIN bit in the E_CSR is set.
6. The E_INVALIDATE bit in the E_CSR is set.
7. The E_RESET bit in the E_CSR is set.
8. Both E-cache lines contain dirty data when the ENET writes to an address that causes a cache miss.
9. The ENET performs a DMA read from the E-cache after having performed a DMA write to the E-cache.

None of these conditions will cause draining if E_ERR_PEND=1, indicating that a memory error has occurred. If Condition 4, 6, or 7 occurs when E_ERR_PEND=1, the E-cache will be invalidated and all dirty data discarded. Condition 1 will always cause the dirty bytes in that cache line to be queued for draining to memory, while data can continue to be written into the other line if the ENET is still transferring. The last byte of a cache line is the byte whose least significant address bits are 0xF or 0x1F, depending on the E_BURST_SIZE field in the E_CSR. Condition 2, the ENET writing a descriptor, indicates that it has completed transferring a data block or has detected an error in transferring, and is detected as a single half-word written during a single assertion of e_hold_. This draining condition can be disabled by setting the E_DSBL_WR_DRN bit in the E_CSR. Condition 5 provides a way for the CPU to explicitly drain all dirty bytes in the cache by setting the E_DRAIN bit in the E_CSR. The E_DRAIN bit clears itself once draining is complete. Condition 8 occurs when the ENET attempts to write data to an address that doesn't match one of the cache tags and both cache lines contain dirty data. In this case, both cache lines will automatically be queued for draining in order to make room for the new data to be written. Condition 9 ensures that the ENET doesn't write data to the E-cache and then read it back before the E-cache has a chance to drain the data to memory.

Note that if an I/O device connected to the ENET interface makes more than one write to the same address without satisfying one of the draining conditions, the data previously stored at that address in the E-cache will be written over each time new data is written to that address.

Transfers From Memory

If the data requested by the ENET is not in E-cache or marked as not valid, then the Least Recently Used (LRU) E-cache line will be filled with the line from memory that contains the data requested. As soon as the data requested by the ENET is written to the E-cache, it is transferred on to the ENET (even if the entire E-cache line has not yet been filled). Also, the word containing the requested data will be the first one to be read from memory.

If the data requested by the ENET is in the E-cache and it is valid, then the ENET gets a cache hit, and the requested data is transferred to the ENET and marked as no longer valid in the E-cache. If part or all of the data requested by the ENET comes from a byte in an E-cache line whose least significant address bits are 0x6, and the next cache line is on the same 4k page as the current line, then a read-ahead operation will be performed. When a read-ahead occurs, the E-cache line that does not contain the data requested by the ENET will be filled with the next line from memory. The read-ahead happens concurrently with the transfer of the requested data to the ENET unless the ENET's request resulted in a cache miss. If the ENET receives a cache miss on a request that causes a read-ahead, then the e_rdy_ signal will not be asserted to the ENET until the SBus burst transfer (caused by the cache miss) has completed. Positioning the read-ahead line as it is, at byte 6 of the cache line, causes data from large blocks to be automatically pre-fetched, but does not cause superfluous pre-fetches on descriptor reads. When the last byte of a cache line is accessed, the other line is marked as Most Recently Used (MRU). Otherwise, the line that was most recently accessed by the ENET is marked as MRU. The line that is not marked as MRU is marked as LRU.

There are several conditions that will cause all valid data that has been read into the E-cache from memory to be invalidated before it can be read by the ENET. If the ENET reads the same address more than once without any intervening DMA accesses to the

DMA2, then the DMA2 detects that the ENET is polling and marks all bytes in the E-cache as invalid. This ensures consistency between the E-cache and memory for the data that is in the same cache line as the accessed data. A CPU slave write to any of the ENET's internal registers will also cause all bytes in the E-cache to be marked invalid if the E_DSBL_WR_INVAL bit in the E_CSR is 0. If E_DSBL_WR_INVAL = 1, the E-cache is unaffected by slave writes to the ENET, which has been shown to markedly increase the performance of the ENET driver software. These and the other conditions that invalidate all bytes in the E-cache are summarized as follows:

1. The ENET reads the same address more than once without intervening DMA accesses (polling).
2. The CPU does a slave write to a ENET internal CSR (disabled by setting E_DSBL_WR_INVAL in E_CSR).
3. The E_INVALIDATE bit in the E_CSR is set.
4. The E_RESET bit in the E_CSR is set.
5. sbus_reset_ is asserted.

Memory Errors

A transfer from the ENET to memory is composed of two separate transfers: one from the ENET to the DMA2, and another from the DMA2 to memory. Similarly, a transfer from memory to the ENET is composed of a transfer from memory to the DMA2, and another from the DMA2 to the ENET. A memory error can only occur during a transfer between the DMA2 and memory.

The DMA2 reports ENET-related memory errors by setting the E_ERR_PEND bit in the E_CSR. When E_ERR_PEND is set, the DMA2 forces the ENET to time out by withholding e_rdy after it has assorted e_hlda. It then generates an interrupt request directly to the CPU by asserting sb_e_irq_ when interrupts are enabled by E_INT_EN being set. This gives rise to two interrupt sources, ENET and DMA2, both of which must be cleared by software. This can be done in one of two ways, either by setting the E_RESET bit in the E_CSR, which resets the ENET and clears E_ERR_PEND, or by resetting the ENET by writing to its internal control register and then clearing E_ERR_PEND by writing E_INVALIDATE. E_ERR_PEND is not cleared by a slave write to the ENET.

To restart ENET DMA activity after an error, it will be necessary to write to the ENET's internal registers. If the E_DSBL_WR_INVAL bit in the E_CSR is set, software must invalidate the E-cache explicitly so that invalid cache data is not used when the ENET is restarted.

For a memory time-out or protection error on a transfer from the ENET to memory, the DMA2 behaves in the following way:

The E_ERR_PEND bit in the E_CSR is set as soon as the error is detected. This causes the DMA2 to generate an interrupt (assert sb_e_irq_) if interrupts are enabled by E_INT_EN being set. No further transfers from the DMA2 to memory will take place until the E_ERR_PEND bit is cleared.

For a memory time-out or protection error on a transfer from memory to the ENET, the DMA2 behaves in the following way:

The E_ERR_PEND bit in the E_CSR is set as soon as the error is detected. This causes the DMA2 to generate an interrupt (assert sb_e_irq_) if interrupts are enabled by E_INT_EN being set. No further transfers to the ENET take place until the E_ERR_PEND bit is cleared.

Registers Internal to the ENET

Slave accesses to the ENET registers will be completed with a 16-bit port acknowledge signal, sb_ack(2) & sb_ack(1), as defined in the SBus specification. The CPU accesses registers internal to the ENET in the same way as it does registers internal to DMA2, by asserting DMA2's sb_sel_ in conjunction with sb_as_. The address map for these registers is shown in Table 3-3.

Registers Internal to DMA2

Ethernet Control/Status Register (E_CSR)

Bit	Mnemonic	Description	Type
31:0	E_CSR	Control/Status bits for Ethernet Interface	R/W

The E_CSR is a 32-bit R/W register internal to DMA2 and is accessed as indicated in the DMA2 Internal Registers Address Map (Table 3-1). The bits in the E_CSR are defined in the following table:

E_CSR Bit Definitions

Bit	Mnemonic	Description	Type
0	E_INT_PEND	Set when e_irq_ active, cleared when e_irq_ inactive.	R
1	E_ERR_PEND	Set when memory time-out, protection, or late error detected on a ENET transfer.	R
3:2	E_DRAINING	Both bits set when E-cache draining dirty data to memory, otherwise both bits are 0.	R
4	E_INT_EN	When set, enables sb_e_irq_ to assert when E_INT_PEND or E_ERR_PEND is set.	R/W
5	E_INVALIDATE	When set, marks all bytes in E-cache as invalid. Resets itself. Reads as 0.	W
NOTE: All bits in the E_CSR default to 0 on an E_RESET or SBus reset with the following exceptions: E_RESET itself is cleared by an SBus reset but otherwise remains active once set until written to 0 and E_DEV_ID is hard-wired to 1010.			

Bit	Mnemonic	Description	Type
6	E_SLAVE_ERR	Set on slave access size error to a ENET-related register. Reset on write of 1.	R/W[1]
7	E_RESET	When set, invalidates the E-cache, resets the ENET interface, and asserts e_reset.	R/W
9:8	—	Unused. Read as 0.	R
10	E_DRAIN	When set, forces draining of E-cache. Resets itself when draining complete.	R/W
11	E_DSBL_WR_DRN	When set, disables draining of E-cache on descriptor writes from ENET.	R/W
12	E_DSBL_RD_DRN	When set, disables draining of E-cache on slave reads to ENET.	R/W
14:13	—	Unused. Read as 0.	R
15	E_ILACC	When set, modifies ENET DMA cycle.	R/W
16	E_DSBL_BUF_WR	When set, disables buffering of slave writes to ENET.	R/W
17	E_DSBL_WR_INVAL	Defines whether E-cache is invalidated on slave writes to ENET; 1 = no invalidate.	R/W
19:18	E_BURST_SIZE	Defines size of SBus read and write bursts for ENET transfers (see table below).	R/W
20	—	Unused.	R/W
21	E_LOOP_TEST	When set, enables Ethernet loop-back mode by tristating TP_AUI_output.	R/W
22	E_TP_AUI_	When E_LOOP_TEST = 0, selects TP or AUI Ethernet by driving TP_AUI to 1 or 0.	R/W
27:23	—	Unused. Read as 0.	R
31:28	E_DEV_ID	Device ID (For DMA2, E_DEV_ID = 1010)	R
NOTE: All bits in the E_CSR default to 0 on an E_RESET or SBus reset with the following exceptions: E_RESET itself is cleared by an SBus reset but otherwise remains active once set until written to 0 and E_DEV_ID is hard-wired to 1010.			

E_CSR Bit Function Notes

E_ERR_PEND - Bit 1

E_ERR_PEND is always reset on setting E_INVALIDATE or E_RESET.

E_DRAINING - Bits 3:2

When the E-cache is draining to memory, both bits are set. Do NOT assert E_RESET or E_INVALIDATE when set. E_DRAINING bits are not valid while E_ERR_PEND is set or during transfers to the ENET. In these cases, the E_DRAINING bits should be ignored.

E_INT_EN - Bit 4

Setting this bit allows the DMA2 to generate an interrupt by asserting sb_e_irq_ whenever E_INT_PEND is set due to the ENET asserting e_irq_, or whenever E_ERR_PEND is set due to a memory error.

E_INVALIDATE - Bit 5

Setting this bit invalidates the E-cache. If E_ERR_PEND = 0 when E_INVALIDATE is set, all dirty data in the E-cache will first be drained to memory. If E_ERR_PEND=1 when E_INVALIDATE is set, all dirty data in the E-cache will be discarded. In addition to invalidating the E-cache, setting this bit causes E_ERR_PEND to be reset.

E_RESET - Bit 7

E_RESET will remain active once written as a one until written as a zero, unless cleared by an SBus reset (sb_reset_ asserted). Setting E_RESET or asserting sb_reset_ will invalidate the E-cache, reset all ENET interface state machines to their idle states, and assert the e_reset output for as long as E_RESET or sb_reset_ remains active. If E_ERR_PEND=0 when E_RESET is set, all dirty bytes in the E-cache will first be drained to memory. When this occurs, E_RESET must not be cleared until draining is complete, as indicated by E_DRAINING = 00. If E_ERR_PEND =1 when E_RESET is set, no draining will take place and all dirty data in the E-cache will be discarded. If the CPU attempts a slave access to the ENET while E_RESET or sb_reset_ is active, it will not cause an error, but no data will actually be transferred since the e_reset output is being asserted throughout the access. In order to satisfy the ENET chip's minimum reset pulse width (200 ns), E_RESET must remain set for at least 10 SBus clocks.

E_DSBL_WR_DRN - Bit 11

When E_DSBL_WR_DRN = 0, dirty bytes in the E-cache are queued for draining to memory whenever the ENET writes to a message descriptor. This is detected as a ENET transfer (one assertion of e_hold_) in which only a single half-word is written. When E_DSBL_WR_DRN = 1, draining of the E-cache on this condition is disabled.

E_DSBL_RD_DRN - Bit 12

When E_DSBL_RD_DRN = 0, dirty bytes in the E-cache are queued for draining to memory whenever the CPU does a slave read to one of the ENET CSR's. When E_DSBL_RD_DRN = 1, draining of the E-cache on this condition is disabled.

E_ILACC - Integrated Local Area Controller Chip - Bit 15

When ENET (or something in its place) is master, and it performs a DMA cycle with ILACC set, e_rdy_ will go inactive when the ENET unasserts e_das_ and e_hlda_ will go inactive when e_hold_ is unasserted. Note that neither e_das_ nor e_hold_ can glitch when the ILACC bit is set.

E_DSBL_WR_BUF - Bit 16
Setting this bit disables the 16-bit buffer for slave writes to ENET.

E_DSBL_WR_INVAL - Bit 17
When E_DSBL_WR_INVAL = 0, the E-cache is invalidated whenever the CPU does a slave write to the ENET. When E_DSBL_WR_INVAL = 1, the E-cache is not invalidated on slave writes to ENET. The latter mode significantly increases tolerable SBus latency while the ENET is transmitting Ethernet data.

E_BURST_SIZE - Bits 19:18
This field defines the sizes of SBus read and write bursts used by DMA2 for ENET transfers and the size of the E-cache lines. All reads from memory will be one size, either a 4- or 8-word SBus burst. SBus writes to memory can be byte, halfword, or word, or one of the burst sizes given in the table. The DMA2 will always use the largest possible size for writes, which is dependent on E_BURST_SIZE and the number of dirty bytes that need to be drained. Also, E_BURST_SIZE determines the size of the E-cache lines, which are the same size as the SBus read burst. The sizes given in the following table are in SBus words:

E_BURST_SIZE	Read Burst Size	Write Burst Sizes	E-cache Line Size
00	4 words	4 words	4 words
01	8 words	4, 8 words	8 words
10	reserved	reserved	reserved
11	reserved	reserved	reserved

E_LOOP_TEST & E_TP_AUI_ - Bits 21 & 22
These bits determine the value of the TP_AUI_ output pin, which drives the NIS input pin of the external T7213 dual Ethernet interface chip. TP_AUI_ selects between the T7213's TP, AUI, and loopback Ethernet modes by driving its NIS input as indicated in the following table:

E_LOOP_TEST	E_TP_AUI_	TP_AUI_*	TP_AUI_Function
0	0	0	NIS Select-AUI mode
0	1	1	NIS Select-TP mode
1	X	Z	NIS Select-Loopback mode
* NOTE: At least 20 mS must elapse after any change to the TP_AUI_ output before transferring data to or from the T7213.			

Ethernet Address Register (E_ADDR)

Bit	Mnemonic	Description	Type
23:0	E_ADDR	Ethernet DMA Address Register	R/W

This register holds bits (23:0) of the address at which the ENET is requesting a DMA transfer. Bits (31:24) of the ENET DMA address are driven by the E_BASE_ADDR register, and have a default value of 0xff. During normal operation, the E_ADDR register is never read or written directly from the SBus, but for testing purposes it can be accessed via bits (23:0) of the E_TST_CSR as described in the following section.

Ethernet Test Control/Status Register (E_TST_CSR)

The E_TST_CSR is a 32-bit R/W register internal to DMA2 provided for testability purposes. It is accessed as indicated in the DMA2 Internal Registers Address Map (Table 3-1). The read and write functions of the bits in the E_TST_CSR are defined in the following table:

Bit	Mnemonic	Description	Type
31	LD_TAG	Writing 1 loads address tag of LRU cache line with E_TST_CSR(23:0).	W
30	REQ_OUT	Reads as 1 when cache is making a request for an SBus read or write.	R
30	RD_LINE	Writing to 1 initiates burst read from memory into LRU cache line from address in E_ADDR.	W
29	HIT	Reads as 1 when address in E_ADDR matches one of the cache address tags and corresponding valid bit is set. Not valid while E_HLDA_asserted (ENET DMA active).	R
28	LRU	Reads as which cache line is LRU (1 or 0). Writing marks specified cache line as LRU.	R/W
27	DRAIN1	Reads as 1 if cache line 1 is draining. Writing to 1 causes dirty bytes in cache line 1 to drain.	R/W
26	DRAIN0	Reads as 1 if cache line 0 is draining. Writing to 1 causes dirty bytes in cache line 0 to drain.	R/W
25	DIRTY1	Determines meaning of Valid/Dirty bits for cache line 1.	R/W
24	DIRTY0	Determines meaning of Valid/Dirty bits for cache line 0.	R/W
23:0	E_ADDR	Reads E_ADDR register.	R
23:0	E_ADDR	Writes E_ADDR register. If LD_TAG=1, also writes LRU cache address tag.	W

NOTE: The E_TST_CSR is intended for diagnostic and test use only, and should never be written while a DMA transfer is active.

Ethernet Cache Valid Bits Register (E_VLD)

Bit	Mnemonic	Description	Type
31:0	E_VLD	Valid/Dirty Bits for LRU Ethernet Cache Line	R/W

NOTE: The E_VLD register is intended for diagnostic and test use only, and should never be written while a DMA transfer is active.

This register accesses the bits which indicate whether each byte in the Least Recently Used cache line is valid, invalid, dirty, or not dirty. Each bit corresponds to the byte of the same number in the LRU cache line. The meaning of these bits for each cache line is determined by the DIRTY1 and DIRTY0 bits, accessible through the E_TST_CSR.

The following table describes the meaning of the E_VLD bits based on the value of the DIRTY1/0 bit corresponding to the LRU cache line:

DIRTY 1/0	E_VLD Bit	Meaning
0	0	Invalid data; reading this byte is a cache miss.
0	1	Valid data; reading this byte is a cache hit.
1	0	Not dirty data; this byte does not need to be drained to memory.
1	1	Dirty data; this byte needs to be drained to memory.

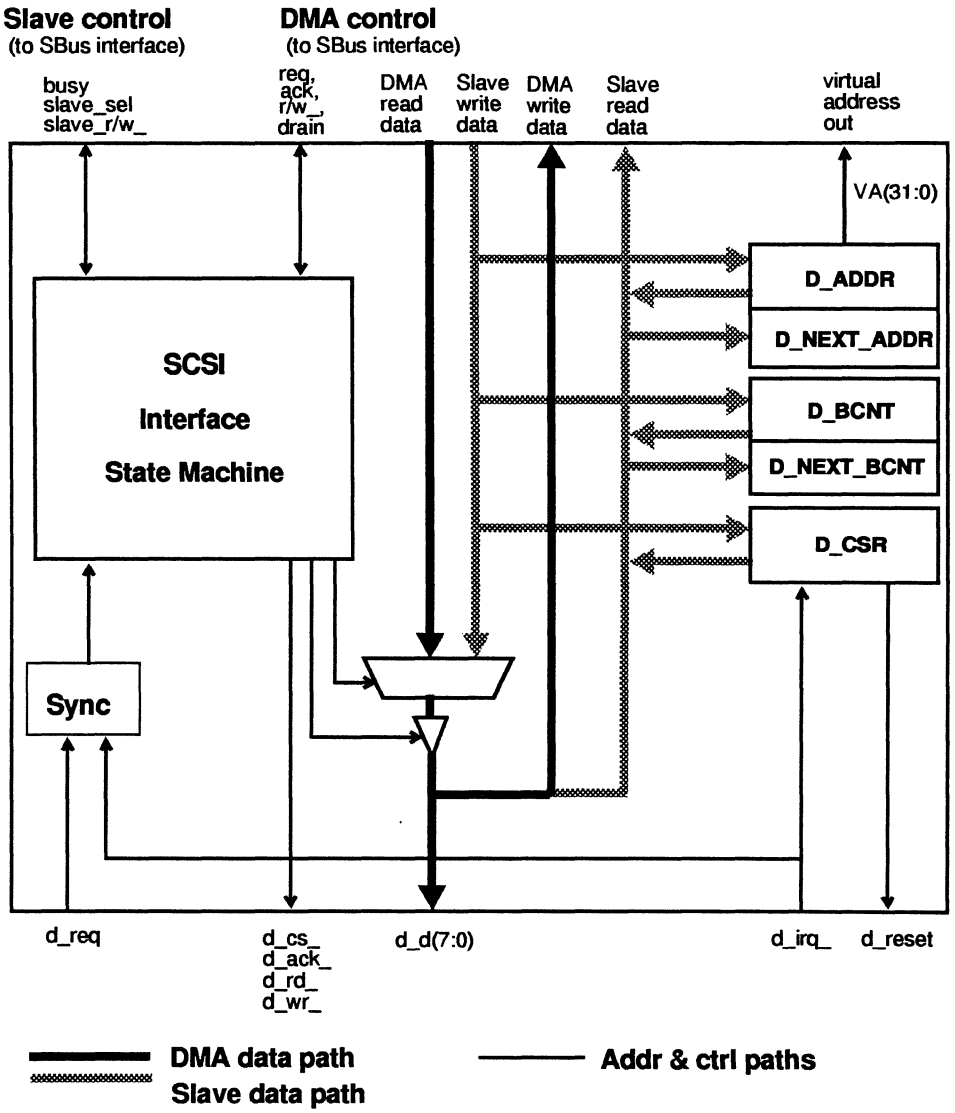
Ethernet Base Address Register (E_BASE_ADDR)

Bit	Mnemonic	Description	Type
7:0	E_BASE_ADDR	High Order 8 bits of address for Ethernet DMA transfers (defaults to 0xff).	R/W

This register makes the high order 8 address bits which are driven onto the SBus for Ethernet DMA transfers programmable by software. By default, these bit are always driven to 0xff, but for some applications it is useful to set a base address.

SCSI Interface

Block Diagram



Overview

The DMA2 gate array provides paths for the SCSI to transfer data to and from memory and for the CPU to access control and status registers on the SCSI. In order to support the DMA activity of the SCSI, the DMA2 contains two 24-bit address counters, a 32-bit control/status register, as well as two 24 bit byte counters which may be used for a SCSI generic control of future devices. The SCSI has a 16-bit transfer count, however the address counter supports a 24-bit count for future enhancements.

Between the SCSI interface and the SBus interface is a 64-byte FIFO (D-FIFO). This FIFO is bypassed for slave accesses to the SCSI.

For error recovery, the D_CSR contains a D_INVALIDATE bit for explicitly invalidating the FIFO.

Transfers To Memory

Data from the SCSI is written into the D-FIFO until the largest possible SBus burst write to memory can be carried out (1, 4, or 8 words depending on the D_BURST_SIZE field in the D_CSR). All data in the D-FIFO is then queued for draining to memory. Any time a SCSI interrupt, byte count expiration, slave write to the D_CSR, or slave access to a SCSI-internal register is detected, data in the D-FIFO is also queued for draining. The last two conditions can be disabled by setting appropriate mode bits in the D-CSR. When the D-FIFO is invalidated by setting the D_INVALIDATE or D_RESET bits in the D_CSR, all data currently in the D-FIFO will first be drained to memory unless the D_ERR_PEND bit is set. If D_ERR_PEND is set when D_INVALIDATE or D_RESET is asserted, or an SBus reset occurs (sb_reset_ is asserted), any data in the D-FIFO will be discarded. These and other conditions that cause data in the D-FIFO to be drained to memory are summarized below:

1. 4, 16, or 32 bytes (depending on D_BURST_SIZE) have been written to the D-FIFO by the SCSI.
2. The D_INT_PEND bit in the D_CSR is set.
3. The CPU does a slave write to the D_CSR (disabled by setting D_DSBL_CSR_DRN bit in D_CSR).
4. The CPU does a slave access to a SCSI-internal register (disabled by setting D_DSBL_SCSI_DRN bit in D_CSR).
5. The D_RESET or D_INVALIDATE bit in the D_CSR is set.
6. The D_ADDR register is loaded with a new address, either directly or from the D_NEXT_ADDR register (disabled by setting D_DIAG bit in D_CSR).

None of these conditions will cause draining if D_ERR_PEND=1, indicating that a memory error has occurred. If Condition 5 or 6 occurs when D_ERR_PEND=1, the D-FIFO will be invalidated and all dirty data discarded. Draining is carried out completely in hardware and until it is completed, subsequent DMA reads from the SCSI are inhibited (by withholding d_ack_). A mechanism for determining when the D-FIFO has been drained is provided in the D_CSR.

Transfers From Memory

When the SCSI initiates a read operation, the DMA2 does an SBus read burst of the size defined by D_BURST_SIZE, either 1, 4, or 8 SBus words. The word containing the requested data will be the first one to be read from memory and will be transferred on to the SCSI as soon as it is written to the D-FIFO, even before the rest of the burst read is complete.

When the SCSI is reading data and the D-FIFO has enough space to hold another read burst of data (4, 16, or 32 bytes, depending on D_BURST_SIZE), a read-ahead operation will be performed as long as it doesn't cross a 4K page boundary. When this occurs, the largest possible SBus burst read is performed beginning with the address of the word following the last word in the D-FIFO. The read-ahead happens concurrently with the transfer of data to the SCSI.

Memory Errors

A transfer from the SCSI to memory is composed of two separate transfers: one from the SCSI to the DMA2, and another from the DMA2 to memory. Similarly, a transfer from memory to the SCSI is composed of a transfer from memory to the DMA2, and another from the DMA2 to the SCSI. A memory error can only occur in a transfer between the DMA2 and memory. The DMA2 reports SCSI-related memory errors by setting the D_ERR_PEND bit in the D_CSR, which generates an interrupt request (asserts sb_d_irq_) to the CPU when enabled by D_INT_EN being set. When D_INT_EN is set, sb_d_irq_ will also be asserted (level sensitive) whenever d_irq_ is asserted by the SCSI.

For a memory time-out or protection error on a transfer from the SCSI to memory, the DMA2 behaves in the following way:

The D_ERR_PEND bit in the D_CSR is set as soon as the error is detected, causing the DMA2 to generate an interrupt (assert sb_d_irq_) if interrupts are enabled by D_INT_EN being set. After the transfer that caused the error, no further transfers will take place from the DMA2 to memory. The interrupt is active until the D_ERR_PEND bit has been cleared by setting the D_INVALIDATE or D_RESET bits.

For a memory time-out or protection error on a transfer from memory to the SCSI, the DMA2 behaves in the following way:

Since the DMA2 reads data that the SCSI has not yet requested (without crossing page boundaries), it is possible that an error could be detected on a data read that the SCSI has never requested. Whether the SCSI requested the memory read that caused the error or not, the data from that read will not be written to the SCSI. Instead the D_ERR_PEND bit is set, which causes the DMA2 to generate an interrupt (assert sb_d_irq_) if interrupts are enabled by D_INT_EN being set. No transfers to the SCSI take place until D_ERR_PEND is cleared.

Registers Internal to the SCSI

Access to control and status registers in the SCSI are achieved using the address map in Table 3-2. These addresses are external to the DMA2, however the chip does decode the appropriate higher order address bits, sb_a(x, y, z) = 100, and generate the chip select and read/write lines to the external device. The DMA2 responds to slave accesses of these registers with sb_ack(1).

SCSI Support Registers Internal to DMA2

Access to SCSI support registers within DMA2 are achieved using the address map in Table 3-1. These registers provide status and control for the SCSI and are selected by the host when sb_sel_ and sb_as_ are zero and sb_a(x, y, z) = 010. The Next feature registers are selected when D_EN_NEXT= 1. The Byte Count Registers are only used when enabled by bit 13, D_EN_CNT, in the D_CSR.

SCSI Control/Status Register (D_CSR)

Bit	Mnemonic	Description	Type
31:0	D_CSR	Control/Status bits for SCSI interface	R/W

The D_CSR is a 32-bit R/W register internal to DMA2 and is accessed as indicated in the DMA2 Internal Registers Address Map (Table 3-1). The bits in the D_CSR are defined in the following table:

D_CSR Bit Definitions

Bit	Mnemonic	Description	Type
0	D_INT_PEND	Set when d_irq active, or when D_TC set and D_TCI_DIS not set. Cleared otherwise.	R
1	D_ERR_PEND	Set when memory time-out, protection, or late error detected on SCSI DMA transfer.	R
3:2	D_DRAINING	Both bits set when D-FIFO draining SCSI data to memory, otherwise both bits are 0.	R
4	D_INT_EN	When set, enables sb_d_irq_ to assert when D_INT_PEND or D_ERR_PEND is set.	R/W
5	D_INVALIDATE	When set, invalidates the D-FIFO. Resets itself. Reads as 0.	W
6	D_SLAVE_ERR	Set on slave access size error to a SCSI-related register. Reset on write of 1.	R/W1
7	D_RESET	When set, invalidates the D-FIFO, resets the SCSI interface, and asserts d_reset output.	R/W
8	D_WRITE	DMA direction for SCSI transfers; 1 = to memory.	R/W
9	D_EN_DMA	When set, enables DMA requests from SCSI if DMA not stopped by other conditions.	R/W
12:10	—	Unused. Read as 0.	R

Bit	Mnemonic	Description	Type
13	D_EN_CNT	When set, enables internal byte counter, D_BCNT, to be decremented on byte transfers.	R/W
14	D_TC	Terminal Count; set when D_BCNT has expired. Reset on write of 1 if D_EN_NEXT=1.	R/W1
15	—	Unused. Read as 0.	R
16	D_DSBL_CSR_DRN	When set, disables draining of D-FIFO on slave writes to D-CSR.	R/W
17	D_DSBL_SCSI_DRN	When set, disables draining of D-FIFO on slave accesses to SCSI-internal registers.	R/W
19:18	D_BURST_SIZE	Defines sizes of SBus read and write bursts for SCSI transfers. (see table below)	R/W
20	D_DIAG	When set, disables draining & resetting of D-FIFO on writing of D_ADDR register.	R/W
21	D_TWO_CYCLE	When set and D_FASTER=0, sets SCSI DMA transfer rate to (2 SBus clocks)/ byte.	R/W
22	D_FASTER	When set and D_TWO_CYCLE=0, sets SCSI DMA transfer rate to (3 SBus clocks)/byte.	R/W
23	D_TCI_DIS	When set, disables D_TC from generating an interrupt. Defaults to 0.	R/W
24	D_EN_NEXT	When set, enables next address auto-load mechanism. D_EN_CNT must also be set.	R/W
25	D_DMA_ON	When set, indicates that the DMA2 is able to respond to DMA requests from the SCSI.	R
26	D_A_LOADED	Address Loaded. Set when D_ADDR written or NEXT_ADDR copied to DADDR.	R
27	D_NA_LOADED	Next Address Loaded. Set when NEXT_ADDR register written.	R
31:28	D_DEV_ID	Device ID (For DMA2, D_DEV_ID = 1010).	R

NOTE: All bits in the D_CSR default to 0 on a D_RESET or SBus reset with the following exceptions: D_RESET itself is cleared by an SBus reset but otherwise remains active once set until written to 0 and D_DEV_ID is hard-wired to 1010.

D_CSR Bit Function Notes

D_ERR_PEND - Bit 1

DMA to/from the SCSI is stopped while this bit is set. D_ERR_PEND is reset on setting D_INVALIDATE or D_RESET.

D_DRAINING - Bits 3:2

When D-FIFO is draining to memory, both bits are set. Do not assert D_RESET or D_INVALIDATE or write to D_ADDR register when set. D_DRAINING bits are not valid while D_ERR_PEND is set or during transfers to the SCSI. In these cases, the D_DRAINING bits should be ignored.

D_INVALIDATE - Bit 5

Setting this bit invalidates the D-FIFO. If D_ERR_PEND = 0 when D_INVALIDATE is set, all dirty data in the D-FIFO will first be drained to memory. If D_ERR_PEND=1 when D_INVALIDATE is set, all dirty data in the D-FIFO will be discarded. In addition to invalidating the D-FIFO, setting this bit causes D_ERR_PEND and D_TC to be reset. If D_EN_NEXT = 1, D_A_LOADED and D_NA_LOADED will also be reset.

D_RESET - Bit 7

D_RESET will remain active once written as a one until written as a zero, unless cleared by an SBus reset (sb_reset_ asserted). Setting D_RESET or asserting sb_reset_ will invalidate the D-FIFO, reset all SCSI interface state machines to their idle states, and assert the d_reset output for as long as D_RESET or sb_reset_ remains active. If D_ERR_PEND=0 when D_RESET is set, all dirty data in the D-FIFO will first be drained to memory. When this occurs, D_RESET must not be cleared until draining is complete, as indicated by D_DRAINING = 00. If D_ERR_PEND=1 when D_RESET is set, no draining will take place and all dirty data in the D-FIFO will be discarded. If the CPU attempts a slave access to the SCSI while D_RESET or sb_reset_ is active, it will not cause an error, but no data will actually be transferred since the d_reset output is being asserted throughout the access.

D_EN_CNT & D_EN_NEXT - Bits 13 and 24

The D_EN_CNT and D_EN_NEXT control the modes of operation of the DMA Address register, Next Address register, Byte count and Next Byte count registers as follows:

D_EN_CNT	D_EN_NEXT	Mode
0	0	Backward compatible with early SPARC stations.
0	1	Illegal mode. Do not use.
1	0	Backward compatible with early SPARC stations using byte count. Invalidate clears D_TC flag.
1	1	Enables loading of D_NEXT_ADDR & D_NEXT_BCNT registers and auto-load of D_ADDR & D_BCNT on byte count expiration. Halt DMA on count expiration if D_NEXT_ADDR not loaded.
NOTE: D_EN_CNT must always be set when D_EN_NEXT is set.		

D_TC - Terminal Count - Bit 14

The D_TC bit will be set when D_BCNT makes a transition from 0x000001 to 0x000000. When it is set, an interrupt will be generated on the d_irq_ pin (if enabled by D_INT_EN and not disabled by D_TCI_DIS). When D_EN_NEXT= 0, D_TC is cleared by D_INVALIDATE, D_RESET or sb_reset_. When D_EN_NEXT= 1, D_TC can also be cleared by writing a 1 to it.

D_BURST_SIZE - Bits 19:18

This field defines the sizes of SBus read and write bursts used by DMA2 for SCSI transfers. All reads from memory will be one size, either 4, 8, or 1 word (in 'no burst' mode). SBus writes to memory can be byte, half-word, or word, or one of the burst sizes given in the table. The DMA2 will always use the largest possible size for writes, which is dependent on D_BURST_SIZE and the number of bytes that need to be drained. Also, D_BURST_SIZE determines the draining level of the D-FIFO. When the D-FIFO has been filled with this amount of data, it will always be drained to memory. The sizes given in the following table are in SBus words.

D_BURST_SIZE	Read Burst Size	Write Burst Sizes	D-FIFO Draining Level
00[1]	4 words	4 words	4 words
01	8 words	4, 8 words	8 words
10	no bursts[2]	no bursts	1 word
11	reserved	reserved	reserved

[1] Default.
[2] SBus reads are always 1 word in 'no burst' mode.

D_TWO_CYCLE - Bit 21

Setting this bit modifies the timing of SCSI DMA such that one byte can be transferred every two SBus clocks, for a peak transfer rate of 10 MByte/sec in systems with a 20 MHz sb_clk. This mode should only be used with SCSI controllers supporting this data rate, such as the NCR53C9X. This bit should ONLY be set when D_FASTER=0.

D_FASTER - Bit 22

Setting this bit modifies the timing of SCSI DMA such that one byte can be transferred every three SBus clocks, for a peak transfer rate of 6.67 MByte/sec in systems with a 20 MHz sb_clk. This mode should only be used with SCSI controllers supporting this data rate, such as the NCR53C90A. This bit should ONLY be set when D_TWO_CYCLE=0.

D_DMA_ON - Bit 25

Reads as 1 when (D_A_LOADED or D_NA_LOADED) & D_EN_DMA & NOT (D_ERR_PEND); otherwise reads as 0. When set, indicates that DMA2 is able to respond to DMA requests from the SCSI.

D_A_LOADED & D_NA_LOADED - Bits 26 and 27.

These bits define the validity of the values stored in the D_ADDR and D_NEXT_ADDR registers. D_A_LOADED is set when D_ADDR is written directly or when D_NEXT_ADDR is copied to D_ADDR, and is reset by D_RESET, (D_INVALIDATE & D_EN_NEXT), or D_BCNT expiring. D_NA_LOADED is set when D_NEXT_ADDR is written. It is reset by D_RESET, D_EN_CNT = 0, D_EN_NEXT = 0, (D_INVALIDATE & D_EN_NEXT), or D_NEXT_ADDR being copied to D_ADDR. When the state is reached where a valid D_NEXT_ADDR has been loaded and the current D_ADDR has been marked as invalid (D_NA_LOADED = 1 & D_A_LOADED = 0), then the contents of the D_NEXT_ADDR register are copied to the D_ADDR register. The copy takes place on the same clock edge where the stated condition is sampled as true. So if address chaining has been set up to take place when the byte count expires, the actual sequence of events will be the following: First, D_A_LOADED will be cleared (on expiration of byte count). Second, the D_NEXT_ADDR will be copied to the D_ADDR register. These two events will probably be seen as one by software.

Address Register (D_ADDR) & NEXT Address Register (D_NEXT_ADDR)

Bit	Mnemonic	Description	Type
31:0	D_ADDR	Virtual Address Counter for SCSI access VA(31:0)	R/W
31:0	NEXT_ADDR	NEXT Address Register	W

The value in this register after a D_RESET is indeterminate.

The Address Register is a 32-bit loadable counter which always points to the next byte that will be accessed by the SCSI, independent of which bytes in memory have been accessed by the DMA2.

If the D_EN_NEXT (enable next address) bit in the D_CSR is set, then a write to the D_ADDR register will write to the D_NEXT_ADDR register instead. If D_EN_NEXT is set when the byte counter (D_BCNT) expires, and the D_NEXT_ADDR register has been written since the last time the byte counter expired, then the contents of D_NEXT_ADDR are copied into D_ADDR. If D_EN_NEXT is set when the byte counter (D_BCNT) expires, but the D_NEXT_ADDR register has not been written since the last time the byte counter expired, then DMA activity is stopped and DMA requests from the SCSI will be ignored until D_NEXT_ADDR is written, or D_EN_NEXT is cleared. (Also, the D_DMA_ON bit will read as 0 while DMA is stopped because of this). When DMA is re-enabled by writing to the D_NEXT_ADDR register, the contents of D_NEXT_ADDR are copied into D_ADDR before DMA activity actually begins.

If the D_NEXT_ADDR register is written before the D_ADDR register has been written, the address written to D_NEXT_ADDR will immediately be copied into D_ADDR. When this occurs, it also causes the value in the D_NEXT_BCNT register to be copied into the D_BCNT register. This allows for a shortcut in loading both D_ADDR and D_NEXT_ADDR along with D_BCNT and D_NEXT_BCNT by writing the registers in the following sequence: D_NEXT_BCNT, D_NEXT_ADDR, D_NEXT_BCNT, D_NEXT_ADDR. When the first value is written into D_NEXT_ADDR, it is immediately copied into D_ADDR since D_ADDR hasn't been

loaded yet. This causes the first value that was written into D_NEXT_BCNT to be copied into D_BCNT. The second values written into D_NEXT_BCNT and D_NEXT_ADDR then remain there as the actual next address and byte count. This allows the loading of both the current and next address and byte count registers without having to write the D_CSR to change the D_EN_NEXT bit in between.

NOTE: A write to the D_ADDR register will invalidate the D-FIFO. A write to the D_NEXT_ADDR register does not have this effect.

Byte Counter (D_BCNT) & NEXT Byte Counter (D_NEXT_BCNT)

Bit	Mnemonic	Description	Type
23:0	D_BCNT	Byte Count; counts down to 0, then sets the D_TC bit in the D_CSR	R/W
23:0	D_NEXT_BCNT	NEXT Byte Count Register	W

The value in this register after a D_RESET is indeterminate.

When reading this register as a word, bits 31:24 will read as 0's.

When enabled, the Byte Counter is decremented every time a byte is transferred between the DMA2 and the SCSI in a SCSI DMA cycle. It is decremented immediately after the byte has been transferred. It is not decremented on slave accesses to the SCSI or on transfers between the DMA2 and memory.

If the D_EN_NEXT bit in the D_CSR is set, then a write to the D_BCNT register will write to the D_NEXT_BCNT register instead. Whenever the D_NEXT_ADDR register is copied into the D_ADDR register, the D_NEXT_BCNT register is copied into the D_BCNT register at the same time.

If D_NEXT_ADDR is being copied into D_ADDR and D_NEXT_BCNT has not been written since the last time D_NEXT_BCNT was copied into D_BCNT, the last value that was written into D_NEXT_BCNT will again be copied into D_BCNT. This provides a shortcut in setting up consecutive DMA transfers of equal size from different addresses, in that D_NEXT_BCNT only needs to be written once as long as D_NEXT_ADDR is loaded for each successive transfer.

If D_EN_NEXT is not set when D_BCNT expires (changes from 0x000001 to 0x000000), then DMA activity between the SCSI and the DMA2 will be stopped and the D_DMA_ON bit will read as 0 until D_ADDR is written. If D_EN_NEXT is set, then DMA will be stopped on D_BCNT expiration.

NOTE: Loading D_BCNT with 0 will allow 2^{24} bytes to be transferred before it expires.

SCSI Test Control/Status Register (D_TST_CSR)

The D_TST_CSR is a 32-bit R/W register internal to DMA2 provided for testability purposes. It is accessed as indicated in the DMA2 Internal Registers Address Map (Table 3-1). The read and write functions of the bits in the D_TST_CSR are defined in the following table:

Bit	Mnemonic	Description	Type
31	LD_TAG	Writing to 1 loads FIFO DMA address register (ADDR_TAG) with value in D_ADDR.	W
30	REQ_OUT	Reads as 1 when FIFO is making a request for an SBus read or write.	R
30	RD_BURST	Writing to 1 initiates a DMA burst read from memory into FIFO from address in ADDR_TAG.	W
29	WR_CNT	Writing to 1 loads FIFO CNT register with D_TST_CSR(5:0).	W
28	WRITE	Writing to 1 puts FIFO into 'WRITING' mode. Reads as 1 when FIFO in 'WRITING' mode.	R/W
27	DRAIN	Reads as 1 if FIFO is draining. Writing to 1 forces FIFO to drain.	R/W
26	EMPTY	Reads as 1 if FIFO buffer is empty.	R
25	FULL	Reads as 1 if FIFO buffer is full.	R
24	LO_MARK	Reads as 1 if FIFO buffer has enough room for 1 SBus read burst of data.	R
23	HI_MARK	Reads as 1 if FIFO contains enough data for 1 SBus write burst.	R
5:0	COUNT	Reads CNT register containing number of bytes stored in FIFO buffer.	R
5:0	COUNT	When WR_CNT=1, writes CNT register containing number of bytes stored in FIFO buffer.	W

NOTE: The D_TST_CSR is intended for diagnostic and test use only, and should never be written while a DMA transfer is active.

Programming Notes

To setup a transfer to/from the SCSI using the DMA2

The transfer may be setup by first programming the internal byte count and address registers of the DMA2. But first, the software driver needs to ensure that no error bits are set in the D_CSR. It may do this by issuing a D_INVALIDATE and/or D_RESET command. Once this is done, the driver then programs the WRITE, D_ INT_EN, D_EN_CNT, (D_FASTER), and D_TCI_DIS bits in the D_CSR. These map to the following usages:

WRITE: determines direction of transfer
D_EN_CNT: set to override the SCSI byte count register and use that of DMA2
D_FASTER: this should be set with every D_RESET
D_TCI_DIS: set to disable interrupts upon byte count = 0.
D_INT_EN: set to enable interrupts upon error conditions and byte count = 0.

Next, the SCSI chip should be programmed with the appropriate data for the particular transfer. Finally, the ENA_DMA bit should be set. This bit acts as a gate such that the DMA2 will immediately begin to respond to SCSI requests for service.

The transfer will now complete with one of three events: an error, in which case the driver must go poll the SCSI for status, an interrupt, for which the driver must provide service, or expiration of the byte count register. This last may be used in combination with the chaining feature presented in subsequent sections of this spec.

To stop a transfer to/from the SCSI using the DMA2

The driver may suspend transfers between the SCSI and DMA2 at any time by simply clearing the ENA_DMA bit. Note that memory accesses by the SBus side of the DMA2 chip are still possible even with this bit cleared. The transfer is easily restarted by again setting the ENA_DMA bit.

Use of Internal Byte Counter with Next Address feature disabled

When using the internal Byte Counter and the D_TC flag in the D_CSR with D_EN_NEXT = 0, it is necessary to perform the following procedure for correct operation:

Load Byte Count into D_BCNT

Load DMA address into D_ADDR

Load Peripheral Device with relevant command(s)

Load D_CSR with enables and direction bits (D_EN_DMA, D_EN_CNT, and WRITE)

Data will be transferred as directed until the Byte Count expires, at which point the D_TC flag will be set in the D_CSR and an interrupt will be generated, if enabled. DMA will also be stopped at this time (D_DMA_ON will be cleared). DMA will remain stopped, independent of the value of D_EN_DMA, until D_ADDR is loaded with a new value. (NOTE: this implies that the interrupt service routine should clear D_EN_DMA before writing a new address to D_ADDR.) Once the CPU has serviced the interrupt and wishes to start another DMA operation it is necessary to proceed as follows:

Issue a D_INVALIDATE command and then repeat the above procedure. (Clears D_TC bit of D_CSR)

Use of Internal Byte Counter with Next Address feature enabled

When using the internal Byte Counter, the D_TC flag in the D_CSR and the NEXT ADDRESS feature, it is necessary to perform the following procedure for correct operation:

NOTE: This is a suggested procedure since several methods of programming the chip are possible.

INITIALIZATION

Initialization if the state of the chip is not defined such as after an error.

Write Control register

D_TCI_DIS, D_EN_DMA, D_EN_NEXT, D_RESET = 0

D_INVALIDATE, D_EN_CNT, D_INT_EN = 1

WRITE = value when read

MULTIPLE BLOCK TRANSFERS

To do a multiple block transfer with an interrupt after each block:

Write Control register

D_INT_EN, D_EN_DMA, D_EN_CNT, D_EN_NEXT = 1

D_TCI_DIS, D_INVALIDATE, D_RESET = 0

WRITE = value as read or desired

Write D_BCNT with byte count of the first block.

Write D_ADDR with the starting address of the first block. Set-up and start the SCSI chip to do its transfer.

Write D_NEXT_BCNT with the byte count of the 'next' block.

Loading of the D_NEXT_BCNT is optional because the initial loading of the D_BCNT also loads the next count register.

Write D_NEXT_ADDR with the address of the 'next' block. The transfer of the first block is enabled and the transfer of the next block will happen automatically when Terminal Count is reached, i.e.: the next address and byte counts will be used. (This assumes the loading of the NEXT count and address occur before the first block transfer is complete.)

INTERRUPT

After each interrupt:

Read D_CSR

D_TC and D_DMA_ON should both = 1.

If D_DMA_ON = 0 then, the D_NEXT_ADDR register did not get updated or there is an error pending, D_ERR_PEND. (It could also mean the DMA is not enabled, D_EN_DMA, or the next address feature is not enabled, D_EN_NEXT, but they were set = 1 so this should not be the case.)

Write D_NEXT_BCNT with the byte count of the 'next' block.

Loading of the D_NEXT_BCNT register is optional, but if loaded, must be loaded before the D_NEXT_ADDR register.

Write D_NEXT_ADDR with the address of the 'next' block.

LAST BLOCK

If no Terminal Count interrupt is desired after the last block is transferred: (because we expect an interrupt from the SCSI chip at the end of the transfer.)
On interrupt of the next to last block:

Read D_CSR

D_TC and D_DMA_ON should both = 1.

Write D_CSR

D_TCI_DIS, INT_EN, D_EN_DMA, D_EN_CNT, D_EN_NEXT = 1

D_INVALIDATE, D_RESET = 0

WRITE = value as read or desired

The D_NEXT_ADDR register is not loaded so the transfer will stop.

NEXT TRANSFER

The initialization for the next multi-block transfer is:

Write D_CSR

D_INT_EN, D_EN_DMA, D_EN_CNT, D_EN_NEXT = 1

D_TCI_DIS, D_INVALIDATE, D_RESET = 0

WRITE = value as read or desired

This assumes the chip is in a known state because, if not, the write to the D_BCNT and D_ADDR registers may go to the NEXT registers (See INITIALIZATION).

Write D_BCNT with byte count of the first block.

Write D_ADDR with the starting address of the first block.

Set-up and start the SCSI chip to do its transfer.

Write D_NEXT_BCNT with the byte count of the 'next' block.

Loading of the D_NEXT_BCNT is optional because the initial loading of the D_BCNT also loads the next count register

Write D_NEXT_ADDR with the address of the 'next' block.

TAKING ADVANTAGE OF 'NEXT' FEATURE

If an interrupt occurs and the 'next' address is not available, there is no room in the buffer or data is not available, the processor can take advantage of the fail-safe capability by not loading the D_NEXT_ADDR and the DMA will stop when the count goes to 0.

After the 'next' address becomes known the processor must determine if the transfer of the current block has completed. If it has not, the processor should load the 'next' count and address to continue the transfer. If the transfer has completed, the processor should load the current count and address and the next count and next address to restart the transfer.

NOTES:

1. There will be an interrupt generated when the count goes to zero.

2. The firmware must understand that if the transfer has stopped because the D_NEXT_ADDR was not loaded before the terminal count was reached, it is effectively loading the D_BCNT and D_ADDR registers, not the NEXT registers. This will enable the DMA to restart the transfer.

3. It is necessary to load the count before the address because the loading of the address register causes the restart of the transfer

Additional Notes

Interrupts from the SCSI are visible as D_ INT_PEND in the D_CSR. The D_INT_EN bit is provided to enable or disable the generation of an interrupt to the IU. If an error condition exists during a memory access, (this could be late errors, protection errors, time-outs etc.), the D_ERR_PEND bit will be set. This will cause an interrupt (if enabled) to the IU. Similarly, expiration of the Byte Counter will cause the D_ INT_PEND bit to be set and an interrupt (if enabled). The D_ERR_PEND bit can only be cleared by a D_INVALIDATE or D_RESET command, or sb_reset.

Software should never set the D_RESET or D_INVALIDATE bits in the D_CSR or write to the D_ADDR register while the D_EN_DMA bit is set or while the D_DRAINING bits are set.

TESTING the 'NEXT' REGISTERS

When the processor writes to the D_BCNT and D_ADDR registers the data goes through the 'NEXT' registers. This 'feature' can be used to test the 'NEXT' registers since they can not be read directly.

Bidirectional Parallel Port

The parallel port consists of four 8 bit Parallel Port interface registers, three 16-bit Configuration registers, and a 64-byte FIFO. The parallel port can operate in programmed I/O mode or be DMA driven. The interface direction, timing and protocol is programmable to meet the wide variety of "Centronics" interfaces that exist on peripheral devices.

Three functional blocks make up the interface: DMA Control, P-FIFO, and the parallel port register interface. The functional blocks, register sets, and operational modes are discussed below.

Parallel Port DMA

Parallel port DMA can operate in either a chained or unchained transfer mode. Mode selection, interrupt control and general DMA control is performed via the Parallel Port Control/Status Register (P_CSR). Transfer direction (to or from memory) is reflected in the P_CSR but is controlled via the Parallel Port Transfer Control Register (P_TCR). The operation of both modes is discussed in the following sections.

Unchained DMA Transfers

The unchained mode operates using a single address register and optionally the Byte Count Register. If the byte counter has been enabled by setting the P_EN_CNT bit of the P_CSR, the byte count will be decremented each time a byte is transferred to or from the parallel port data bus, p_data(7:0). When the byte count expires (changes from 0x000001 to 0x000000), the terminal count bit (P_TC) will be set, DMA will be halted, and an interrupt will be generated if enabled via P_INT_EN = 1 and P_TCI_DIS = 0.

Unchained DMA transfers are terminated in one of two ways. If the byte counter expires (changes from 0x000001 to 0x000000), DMA will be halted, P_TC will be set P_A_LOADED will be reset, and P_DMA_ON will be cleared until a new DMA address is loaded into the P_ADDR register. Otherwise, the transfer can be terminated through software by clearing the P_EN_DMA bit of the P_CSR. In both cases, on termination of transfers from memory to the peripheral device the P-FIFO will be invalidated, and on termination of transfers from the peripheral device to memory all valid data in the P-FIFO will be drained to memory.

Chained DMA Transfers

Chained DMA transfers also use the address and byte count registers to perform transfers. In addition, they also use the NEXT address and NEXT byte count register set to accomplish the chaining of DMA transfers. DMA transfer chaining is enabled via the P_EN_NEXT and P_EN_CNT bits in the P_CSR. Note that use of the byte counter is essential to DMA chaining, as it is byte count expiration that initiates the loading of the NEXT address and NEXT byte count into the address and byte count registers.

Chained DMA transfers are terminated in the following ways. If the byte counter expires (changes from 0x000001 to 0x000000) and DMA chaining has been disabled (**P_EN_NEXT** = 0), DMA will be halted, P_TC will be set, P_A_LOADED will be reset, and P_DMA_ON will be cleared until a new DMA address is loaded into the P_ADDR register. If the byte counter expires, DMA chaining is enabled (P_EN_NEXT = 1), and the NEXT address/byte count is invalid (P_NA_LOADED = 0), DMA will be halted until the NEXT register set has been written. Finally, the transfer can be terminated through software by clearing the P_EN_DMA bit of the P_CSR. In all three cases, on

termination of transfers from memory to the peripheral device the P-FIFO will be invalidated, and on termination of transfers from the peripheral device to memory all valid data in the P-FIFO will be drained to memory.

NEXT Register Access

The NEXT register set used for chaining can only be accessed indirectly. The P_EN_NEXT bit of the P_CSR defines whether register writes to the address/byte count registers access the address/byte counter registers or the NEXT address/byte count registers. When the P_EN_NEXT bit is set, all write accesses to the address/byte count register locations are actually writes to the NEXT registers. The NEXT registers shadow the counter registers. Note that the NEXT registers cannot be read directly. Two P_CSR bits pertain to the validity of the address and NEXT address registers. The P_A_LOADED bit indicates whether the address counter contains a valid address and the P_NA_LOADED indicates whether the NEXT address register contains a valid address.

DMA Memory Clear

The clearing of memory can be hardware assisted by setting the MEM_CLR and DATA_SRC bits of the Parallel Port Operation Configuration Register (P_OCR). The MEM_CLR bit specifies a DMA transfer to memory using the DATA_SRC bit as the data to be sourced. Once MEM_CLR has been set, the P_WRITE bit of the P_CSR will reflect the transfer direction to memory. Setting the P_EN_DMA bit will start the transfer. Either of the above DMA transfer methods can be used but in either case the byte counter must be enabled and used to control the transfer. When terminal count is reached and P_EN_NEXT is cleared the transfer will be terminated. When terminal count is reached and P_EN_NEXT is set but the NEXT address/byte count is invalid (P_NA_LOADED cleared), the transfer will be suspended until the NEXT register set has been written. Interrupts on P_TC operate according to the transfer mode selected.

Parallel Port DMA Registers

All Parallel Port DMA registers are accessed as 32 bit registers. However, not all registers have 32 significant bits. The mnemonics for all Parallel Port registers are all preceded by "P". The Addr field of the register definitions correspond to SBus physical address bits 4:0. Additionally, to access these registers the pa(x, y, z) physical address bits must be 101, respectively.

PP DMA Address Register (P_ADDR) and NEXT Address Register (P_NEXT_ADDR)

Addr	Bits	Description	Size	Type
04	31:0	PP DMA Address Register (P_ADDR).	32 Bit	R/W
04	31:0	PP DMA NEXT Address Reg (P_NEXT_ADDR)	32 Bit	W

This 32 bit R/W register contains the virtual address for Parallel Port DMA transfers. It is implemented as a 32-bit loadable counter which points to the next byte that will be accessed via the Parallel Port.

If the P_EN_NEXT (enable next address) bit in the P_CSR is set, then a write to the P_ADDR register will write to the P_NEXT_ADDR register instead. If P_EN_NEXT is set when the byte counter (P_BCNT) expires, and the P_NEXT_ADDR register has been written since the last time the byte counter expired, then the contents of

P_NEXT_ADDR are copied into P_ADDR. If P_EN_NEXT is set when the byte counter (P_BCNT) expires, but the P_NEXT_ADDR register has not been written since the last time the byte counter expired, then DMA activity is stopped and DMA requests from the Parallel Port will be ignored until P_NEXT_ADDR is written or P_EN_NEXT is cleared. (Also, the P_DMA_ON bit will read as 0 while DMA is stopped because of this). When DMA is re-enabled by writing to the P_NEXT_ADDR register, the contents of P_NEXT_ADDR are copied into P_ADDR before DMA activity actually begins.

NOTE: A write to the P_ADDR register will invalidate the P-FIFO. A write to the P_NEXT_ADDR register does not have this effect.

PP DMA Byte Count Register (P_BCNT) & NEXT Byte Count (P_NEXT_BCNT)

Addr	Bits	Description	Size	Type
08	23:0	PP DMA Byte Count (P_BCNT)	32 Bit	R/W
08	23:0	PP DMA NEXT Address Reg (P_NEXT_BCNT)	32 Bit	W

This register is implemented as a 24 bit down counter. When reading this register as a word, bits 31:24 will read as 0's. The register should be loaded with a 24 bit byte count which, if enabled via the P_EN_CNT bit in the P_CSR, will be decremented every time a byte is transferred between the DMA2 and whatever external device is connected to the parallel port. Transition of this register to zero, Terminal Count (P_TC), will generate an interrupt if not disabled via the P_TCI_DIS bit of the P_CSR.

If the P_EN_NEXT bit in the P_CSR is set, then a write to the P_BCNT register will write to the P_NEXT_BCNT register instead. Whenever the P_NEXT_ADDR register is copied into the P_ADDR register, the P_NEXT_BCNT register is copied into the P_BCNT register at the same time. If P_NEXT_ADDR is being copied int P_ADDR and P_NEXT_BCNT has not been written since the last time P_NEXT_BCNT was copied int P_BCNT, the last value that was written into P_NEXT_BCNT will again be copied into P_BCNT. This provides a shortcut in setting up consecutive DMA transfers of equal size from different addresses, in that P_NEXT_BCNT only needs to be written once as long as P_NEXT_ADDR is loaded for each successive transfer. If P_EN_NEXT is not set when P_BCNT expires (changes from 0x000001 to 0x000000), then Parallel Port DMA activity will be stopped and the P_DMA_ON bit will read as 0 until P_ADDR is written. If P_EN_NEXT is set, then DMA will be stopped on P_BCNT expiration.

NOTE: Loading P_BCNT with 0 will allow 2^{24} bytes to be transferred before it expires.

PP Test Control/Status Register (P_TST_CSR)

The P_TST_CSR is a 32-bit R/W register internal to DMA2 which provides testability functions for the parallel port interface. It is accessed as indicated in the DMA2 Internal Registers Address Map (Table 3-1). The functionality of the P_TST_CSR is identical to the D_TST_CSR, except that it affects the P-FIFO instead of the D-FIFO.

Parallel Port Control/Status Register (P_CSR)

This register provides DMA status and control information for the Parallel Port.

ADDR=0, Size=32 Bits

Bit	Mnemonic	Description	Type
0	P_INT_PEND	Set when a PP DMA or PP Control interrupt is pending or when P_TC set and P_TCI_DIS not set.	R
1	P_ERR_PEND	Set when an interrupt is pending due to an SBus error condition.	R
3:2	P_DRAINING	Both bits set when the P-FIFO is draining to memory, otherwise both bits are 0.	R
4	P_INT_EN	When set, enables sb_p_irq_ to become active when either P_INT_-PEND or P_ERR_PEND are set.	R/W
5	P_INVALIDATE	When set, invalidates the P-FIFO. Resets itself. Reads as 0.	W
6	P_SLAVE_ERR	Set on slave access size error to a PP register. Reset by P_RESET, P_IN-VALIDATE, or writing to 1.	R/W1*
7	P_RESET	When set, acts as a hardware reset to the Parallel Port only.	R/W
8	P_WRITE	DMA Direction. 1= To memory; 0=From memory.	R
9	P_EN_DMA	When set, enables DMA transfers to/ from the PP.	R/W
12:10	—	Unused. Reads as 0.	R
13	P_EN_CNT	When set, enables the PP byte counter to be decremented.	R/W
14	P_TC	Terminal count. Set when byte count expires. Reset on write of 1 if P_EN_-NEXT=1.	R/W1*
17:15	—	Unused. Reads as 0.	R
19:18	P_BURST_SIZE	Defines sizes of SBus read and write bursts for PP transfers.	R/W
20	P_DIAG	When set, disables draining and resetting of P-FIFO on loading of P_ADDR register.	R/W
22:21	—	Unused. Reads as 0.	-----
23	P_TCI_DIS	When set, disables P_TC from generat-ing an interrupt.	R/W
24	P_EN_NEXT	When set, enables DMA chaining and NEXT address/byte count auto-load mechanism. P_EN_CNT must also be set.	R/W
*R/W1 bits are readable and cleared by writing to logic 1.			

Bit	Mnemonic	Description	Type
25	P_DMA_ON	DMA On. When set, indicates that DMA transfers are not disabled due to any hardware or software condition.	R
26	P_A_LOADED	Set when the contents of the address and byte count are considered valid during chained transfers.	R
27	P_NA_LOADED	Set when NEXT Address and byte count registers have been written but have not been used for chaining.	R
31:28	P_DEV_ID	Device ID=1010	R
*R/W1 bits are readable and cleared by writing to logic 1.			

The RESET state of this register is as follows:
P_ERR_PEND=P_INT_EN=P_INVALIDATE=P_SLAVE_ERR=P_RESET=
P_WRITE=P_EN_DMA=P_EN_CNT=P_TC=P_BURST_SIZE=P_TCI_DIS=
P_EN_NEXT=P_DMA_ON=P_A_LOADED=P_NA_LOADED=0.

P_INT_PEND

Interrupt pending is the logical OR of the following enabled PP interrupt sources: (P_TC & !P_TCI_DIS), DS_IRQ, ACK_IRQ, BUSY_IRQ, ERR_IRQ, PE_IRQ, SLCT_IRQ.

P_ERR_PEND

Error pending will be set due to an SBus error acknowledge or an SBus late error. It indicates an SBus error condition. PP DMA is stopped (P_DMA_ON=0) when this bit is set. This bit can be reset by setting P_INVALIDATE or P_RESET.

P_DRAINING

When P-FIFO is draining to memory, both bits are set. Do not assert P_RESET or P_INVALIDATE or write to the P_ADDR register when set. P_DRAINING bits are not valid while P_ERR_PEND is set and should be ignored.

P_INVALIDATE

Setting this bit invalidates the P-FIFO. If P_ERR_PEND = 0 when P_INVALIDATE is set, all dirty data in the P-FIFO will first be drained to memory. If P_ERR_PEND = 1 when P_INVALIDATE is set, all dirty data in the P-FIFO will be discarded. In addition to invalidating the P-FIFO, setting this bit causes P_ERR_PEND and P_TC to be reset. If P_EN_NEXT = 1, P_A_LOADED and P_NA_LOADED will also be reset.

P_RESET

This bit functions as a hardware reset to the parallel port. It will remain active once written to one until written to zero, unless cleared by an SBus reset (sb_reset_ asserted). Setting P_RESET or asserting sb_reset_ will invalidate the P-FIFO and reset all parallel port interface state machines to their idle states. If P_ERR_PEND = 0 when P_RESET is set, all dirty data in the P-FIFO will first be drained to memory. When this occurs, P_RESET must not be cleared until draining is complete, as indicated by P_DRAINING = 00. If P_ERR_PEND=1 when P_RESET is set, no draining will take place and all dirty data in the P-FIFO will be discarded.

P_WRITE

This read only bit reflects the direction for DMA transfers. It is a logical OR of the DIR bit of the Parallel Control Register (P_CR) and the MEM_CLR bit of the parallel operation control register (P_OCR).

P_TC

This bit will be set when the byte counter (P_BCNT) transitions to from 0x000001 to 0x000000. This will generate an interrupt if enabled by P_INT_EN and not disabled by P_TCI_DIS. During unchained transfers, P_TC causes P_DMA_ON to be reset. When P_EN_NEXT = 0, P_TC is cleared by P_INVALIDATE, P_RESET, or sb_reset_. When P_EN_NEXT = 1, P_TC can also be cleared by writing a 1 to it.

P_BURST_SIZE

This field defines the sizes of SBus read and write bursts used by the DMA2 for parallel port transfers. All reads from memory will be one size, either 4, 8, or 1 word (in 'no burst' mode). SBus writes to memory can be byte, half-word, word, or one of the burst sizes given in the table. The DMA2 will always use the largest possible size for writes, which is dependent on P_BURST_SIZE and the number of bytes that need to be drained. Also, P_BURST_SIZE determines the draining level of the P-FIFO. When the P-FIFO has been filled with this amount of data, it will always be drained to memory. The sizes given in the following table are in SBus words.

P_BURST_SIZE	Read Burst Size	Write Burst Sizes	P-FIFO Draining Level
00	4 words	4 words	4 words
01	8 words	4, 8 words	8 words
10	no bursts[1]	no bursts	1 word
11	reserved	reserved	reserved

[1] SBus reads are always 1 word in 'no burst' mode.

P_DMA_ON

When set, indicates that DMA2 is able to respond to Parallel Port DMA requests. Reads as 1 when (P_A_LOADED or P_NA_LOADED) & P_EN_DMA & NOT(P_ERR_PEND); otherwise reads as 0.

Parallel Port FIFO (P-FIFO) Operation

Between the Parallel Port and the SBus interface is a 64-byte FIFO (P-FIFO). This FIFO is bypassed for slave accesses to the Parallel Port registers. Consistency control ensures that all data written by the external device gets to main memory in a deterministic manner, and is handled completely in hardware. One of the consistency control mechanisms used on transfers to memory is draining of all P-FIFO data upon the access of any parallel port register. The P-FIFO is implemented using the same logic block as the D-FIFO, and thus has identical functionality internally. Refer to the SCSI section for detailed descriptions of how the FIFO handles transfers to and from memory, invalidation conditions, and memory errors.

The conditions that cause data in the P-FIFO to be drained to memory are slightly different than those for the D-FIFO, and are as follows:

1. 4, 16, or 32 bytes (depending on P_BURST_SIZE) have been written into the P-FIFO.
2. The P_INT_PEND bit in the P_CSR is set.
3. The CPU does a slave write to a Parallel Port internal register other than the P_TST_CSR (writing P_ADDR does not cause draining if P_DIAG is set).
4. The P_RESET or P_INVALIDATE bit in the P_CSR is set.
5. The P_ADDR register is loaded from P_NEXT_ADDR when P_DIAG is not set.

None of these conditions will cause draining if P_ERR_PEND = 1, indicating that a memory error has occurred. If Condition 4 or 5 occurs when P_ERR_PEND = 1, the P-FIFO will be invalidated and all dirty data discarded.

Bidirectional Parallel Port Interface

The parallel port can operate unidirectionally or bi-directionally in either a programmed I/O mode or in a DMA mode. The hardware interface can be configured to operate with a wide range of devices through the following mechanisms:

- Bidirectional signal configuration for the interface control signals–data strobe, acknowledge, and busy. Each control signal can be individually configured as a unidirectional or as a bidirectional signal.
- Programmable pulse widths for all generated signals and programmable data setup time for data transfers.
- Programmable protocol definition for all combinations of acknowledge and busy handshaking.

This interface configuration capability will allow operation over a wide range of data transfer rates and protocol definitions. The operational modes and a description of each of the individual registers are presented below.

DMA Mode

DMA mode automates the data transfer over the port as has been discussed in the previous sections on PP DMA. Since no software intervention is required for data transfer, the interface protocol and timing required must be programmed via the configuration registers. DMA transfers are initiated/enabled by setting the **P_EN_DMA** bit of the P_CSR. The operation of the interface is dependent on the direction of transfer and the protocol selected as described below.

Unidirectional Operation (Transfers To the Peripheral Device)

This mode of operation is the "Centronics" implementation of a unidirectional parallel port. Operation of the parallel port in this mode requires the direction control bit (DIR) of the Transfer Control Register (TCR) to be 0. Timing variations are handled

via the DSS (data setup to data strobe) and DSW (data strobe width) bits of the Hardware Configuration Register. The timebase for programmability is the SBus clock. The DSS parameter (7 bits) can be programed from a minimum of 0 SBus clocks to 127 SBus clocks in steps of one SBus clock. The DSW parameter (7 bits) is also programmed in steps of one SBus clocks, however, when DSW=0,1,2, or 3, data strobe width is defined as three SBus clocks. That is, the minimum data strobe width is three SBus clocks. The following table shows the nominal range of programmability for different SBus clock speeds.

SBus Clock	DSS	DSW
16.67 MHz	0 - 7.62 us	180.0 ns - 7.62 us
20 MHz	0 - 6.35 us	150.0 ns - 6.35 us
25 MHz	0 - 5.08 us	120.0 ns - 5.08 us

The desired handshake protocol can be selected using the ACK_OP (acknowledge operation) and BUSY_OP (busy operation) bits of the Operations Configuration Register (OCR). The function of these bits is defined as follows:

ACK_OP 1 = Handshake complete with receipt of p_ack.
0 = p_ack is ignored.

BUSY_OP 1 = Handshake complete with receipt of p_bsy_.
0 = p_bsy_ is not used for handshaking.

These two bits allow selection of one of four possible protocols, however only three of these protocols make sense and are valid selections. The case of ACK_OP= BUSY_OP=1, is not supported. For all protocol selections, if p_bsy_ is active, further data transfers will not occur until p_bsy_ is negated. The following table summarizes the protocol definitions for transfers to the peripheral device:

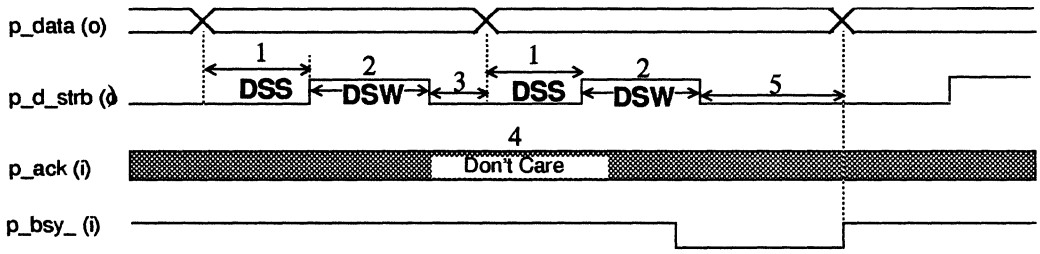
BUSY_OP	ACK_OP	Protocol Definition
0	0	No handshaking occurs.
0	1	Acknowledge is required for each byte transferred.
1	0	p_bsy_ is used as acknowledge and is required for each byte transferred. ack is ignored.
1	1	Invalid.

The transfer modes are shown and discussed in the following sections.

No Handshake (BUSY_OP=0, ACK_OP=0)

Data transfers are controlled by the use of p_d_strb and optionally p_bsy_. There is no acknowledge in this mode and p_ack is a don't care. p_bsy_ is used to gate further transfers when the peripheral device cannot receive another byte of data. p_bsy_ is sampled before data strobe becomes active and after data strobe becomes inactive, to insure that a data transfer is not attempted while the device is busy.

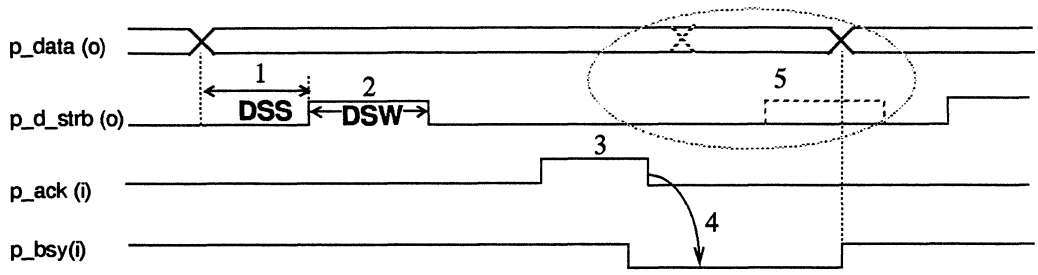
It is this mode, which provides the fastest transfer of data over the interface, the fastest cycle time is six SBus clocks per byte. This transfer time is arrived at as follows: DSS=0, DSW=3 (minimum width of DSW is three SBus clocks), and three SBus clocks between consecutive data strobes. This assumes that p_bsy_ is not asserted during the transfer cycle. Refer to the following data transfer diagram:



1. Data Setup as defined in the hardware configuration register.
2. Data strobe width as defined in the hardware configuration register.
3. There is a three SBus clock delay from the end of data strobe to the next byte of data being clocked onto the p_d data bus.
4. Acknowledge is a don't care condition for all data transfers.
5. When p_bsy_ is active it gates further data transfers.
6. All signal polarities shown are at the pins. Polarities on the interface cable should be inverted (except p_data).

Handshake with Ack (BUSY_OP=0,ACK_OP=1)

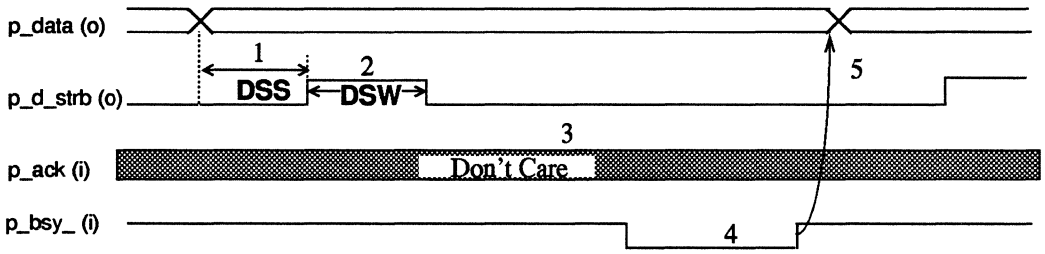
Data transfers are controlled by the use of p_d_strb, p_ack, and optionally p_bsy_. p_ack is required for each byte transferred. If p_bsy_ is active at the end of the cycle, further data transfers will be gated until p_bsy_ becomes inactive. If p_bsy_ is not present then data transfers will proceed. p_bsy_ is also sampled immediately before p_d_strb is generated to insure that a data transfer is not attempted while the device is busy. Refer to the following data transfer diagram:



1. Data Setup as defined in the hardware configuration register.
2. Data strobe width as defined in the hardware configuration register.
3. Acknowledge is required for each byte transferred.
4. When p_bsy_ is active it gates further data transfers.
5. If p_bsy_ is not present, the next data byte will be gated on to the bus following ack (there is a minimum of three SBus clocks between the trailing edge of ack and the next data byte).
6. All signal polarities shown are at the pins. Polarities on the interface cable should be inverted (except p_data).

Handshake with Busy (ACK_OP=0, BUSY_OP=1)

Data transfers are controlled by the use of p_d_strb and p_bsy_. p_ack is a don't care in this mode. p_bsy_ is required as an acknowledge after p_d_strb and will gate any further data transfers while it is active. p_bsy_ is also sampled immediately before p_d_strb is generated to insure that a data transfer is not attempted while the device is busy. Refer to the following data transfer diagram:



1. Data Setup as defined in the hardware configuration register.
2. Data strobe width as defined in the hardware configuration register.
3. Acknowledge is a don't care condition for all data transfers.
4. p_bsy_ is required as an acknowledge for each byte transferred. While p_bsy_ is present, it gates further data transfers.
5. The next byte of data will be gated on to the bus following the trailing edge of p_bsy_ (there is a minimum of three SBus clocks between the trailing edge of p_bsy_ and the next byte of data).
6. All signal polarities shown are at the pins. Polarities on the interface cable should be inverted (except p_data).

Bidirectional Operation

Bidirectional data transfer over the parallel port can be accomplished by the use of either of two master/slave protocols. The "master write" protocol or the "master read/write" protocol. The IBM implementation of a bidirectional parallel port uses the "master write" protocol in which the master always writes data to the slave and when the direction of data transfer needs to be reversed, mastership is exchanged. The Xerox implementation uses the "master read/write" protocol where data transfer is performed in either direction under control of the fixed master. The parallel port will operate as either master or slave when configured for master write protocol, and only as the master when configured for the master read/write protocol.

The selection of one of these bidirectional transfer methods is accomplished indirectly through the specification of the bidirectional nature of the data strobe signal. Since in both methods data strobe resides with the master, a bidirectional data strobe implies the IBM "master write" scheme and fixed data strobe (output only) implies the Xerox "master read/write" scheme.

The interface control signals - data strobe, acknowledge, and busy are individually configurable as bidirectional or unidirectional pins. The bidirectional signal configuration bits are located in the Operation Configuration Register. The function of the bits are as follows:

DS_DSEL
1 = Data strobe is bidirectional.
Master write transfer protocol is selected.
0 = Data strobe is fixed as an output.
Master read/write transfer protocol is selected.

ACK_DSEL
1 = p_ack is bidirectional.
0 = p_ack is fixed as an input.

BUSY_DSEL
1 = p_bsy_ is bidirectional.
0 = p_bsy_ is fixed as an input.

To allow external driver/receiver connection, each of these control signals and the data bus has a corresponding direction control pin. The DIR bit of the Transfer Control Register is used to switch the direction of the data bus and the pins that have been configured as bidirectional. The state of the DIR bit is reflected on the p_d_dir_ pin for external transceiver control and direction control communication to the attached device. While DIR=0, all pins remain in their unidirectional sense which is defined to be consistent with the unidirectional parallel port as follows:

Signal	I/O	Dir_Pin	State
p_d_strb	O	p_ds_dir	1
p_ack	I	p_ack_dir_	0
p_bsy_	I	p_bsy_dir_	0
p_data	O	p_d_dir_	1

When DIR is set to 1, the pins configured as bidirectional change direction and their corresponding direction control pins are set accordingly. Note that the input status pins (err, slct, pe) which are readable in the Input Register are not configurable. They are fixed as inputs. Similarly, the output pins (p_afxn, p_init, p_slct_in) of the Output Register are fixed as outputs.

The transfer modes are shown and discussed in the following sections.

Master Write Protocol, Slave Operation

This section describes the parallel port operation as a slave when it is configured for master write protocol (DS_DSEL=1). Operation as a master is the same as is described in the *Unidirectional Operation* section.

In this mode, acknowledge and/or busy can be generated in response to a data strobe. The width of the p_ack pulse can be defined using the DSW bits of the hardware configuration register. The p_bsy_ hold time and p_ack positioning after the trailing edge of data strobe are defined using the DSS bits. However note that in this mode, DSS has a tolerance of +3 to 4 SBus clocks, due to synchronization delays. The nominal programmability range is the same as was specified in the *Unidirectional Operation* section.

The **ACK_OP** and **BUSY_OP** bits are used to specify handshake protocol. The function of the bits take on a new meaning when the parallel port is a slave.

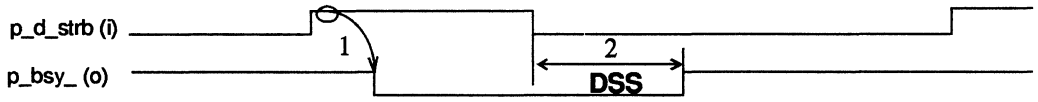
ACK_OP
1 = Generate p_ack in response to a data strobe.
0 = p_ack is not generated. p_ack is held in an inactive state.

BUSY_OP
1 = Generate p_bsy_, as an acknowledge, in response to data strobe.
0 = p_bsy_ is not generated for each byte transferred, but is asserted as required.

These two bits allow selection of one of four possible handshake protocols. The following table summarizes the protocol definitions for transfers to the parallel port from the peripheral device.

BUSY_OP	ACK_OP	Protocol Definition
0	0	No handshaking occurs.
0	1	Acknowledge is generated for each byte transferred.
1	0	p_bsy_ is generated as an acknowledge for each byte transferred. p_ack remains inactive.
1	1	Both p_bsy_ and p_ack are generated for each byte transferred.

For all protocol selections, p_bsy_ will become active if one of the following conditions occur: the P_DMA_ON bit is reset indicating DMA cannot proceed; or the P-FIFO is unable to accept more data. Internally, p_bsy_ will always be generated for these conditions. However, if the p_bsy_ pin is not configured as an output it will not be driven and the external interface will not be able to detect the busy condition. In this case data could be lost. In all cases if p_bsy_ is asserted it will have the following timing characteristics:



1. When data strobe is detected, p_bsy_ will be generated within 3 SBus clocks, if required.
2. p_bsy_ hold time after data strobe is configurable via **DSS**.

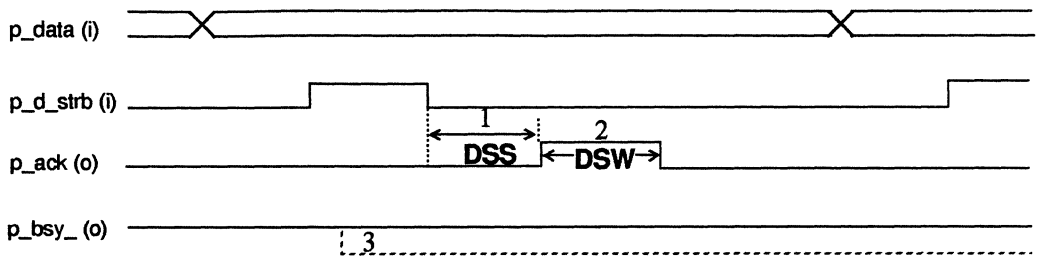
The transfer modes are shown and discussed in the following sections:

No Handshake (BUSY_OP=0, ACK_OP=0)

No handshake signals are generated in this mode. If p_ack is configured as an output it will remain low or inactive. p_bsy_ will be generated as required to gate further transfers but not as a handshake signal. The operation of the interface as defined assumes the bidirectional sense of each signal has been configured as follows: DIR=1, DS_DSEL=1, ACK_DSEL=X, BUSY_DSEL=1. If p_ack is configured as an output it will remain low or inactive. The configuration of p_bsy_ as an output is suggested to avoid potential data loss. Reference the parallel port timing section for detailed timing requirements for this mode.

Handshake with Ack (BUSY_OP=0, ACK_OP=1)

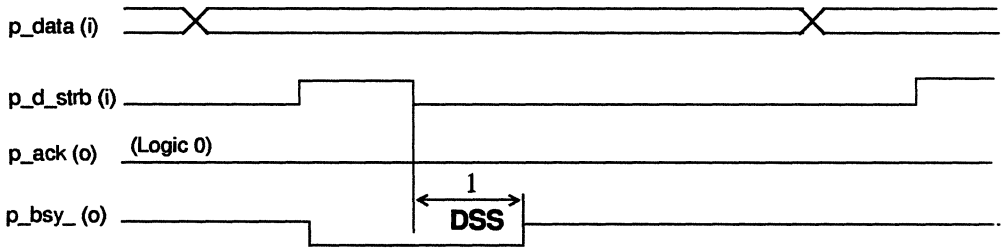
Data transfers are acknowledged using p_ack. The position of p_ack relative to the trailing edge of data strobe is set using DSS. Note that in this mode, the actual positioning of p_ack will be DSS plus 3 to 4 SBus clocks, due to synchronization delays. The width of p_ack is set using DSW. p_bsy_ will be generated as required to gate further transfers but not as a handshake signal. The operation of the interface as defined assumes the bidirectional sense of each signal has been configured as follows: DIR=1, DS_DSEL=1, ACK_DSEL=1, BUSY_DSEL=1. The configuration of p_bsy_ as an output is suggested to avoid potential data loss. Refer to the following data transfer diagram:



1. Acknowledge position relative to data strobe (DSS-Hardware configuration register).
2. Acknowledge width (DSW-Hardware configuration register).
3. p_bsy_ will be asserted if required.
4. All signal polarities shown are at the pins. Polarities on the interface cable should be inverted (except p_data).

Handshake with Busy (BUSY_OP=1, ACK_OP=0)

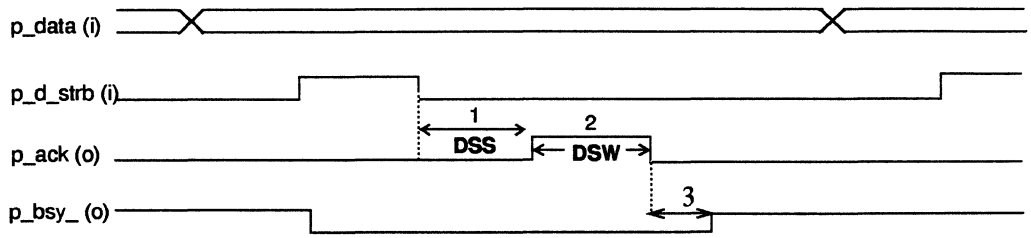
Data transfers are acknowledged using p_bsy_. p_bsy_ will be generated off of the leading edge of p_d_strb and will remain active for the period specified by DSS (plus 3 to 4 SBus clocks) beyond the end of p_d_strb. The operation of the interface as defined assumes the bidirectional sense of each signal has been configured as follows: DIR=1, DS_DSEL=1, ACK_DSEL=X, BUSY_DSEL=1. The configuration of p_ack as an input will not hinder the operation of the interface as far as handshaking is concerned. If p_ack is configured as an output it will remain low or inactive. Refer to the following data transfer diagram:



1. p_bsy_ hold time after data strobe (DSS-Hardware configuration register).
2. All signal polarities shown are at the pins. Polarities on the interface cable should be inverted (except p_data).

Handshake with Ack and Busy (BUSY_OP=1, ACK_OP=1)

Both p_ack and p_bsy_ are generated in response to a data strobe. p_bsy_ will be generated off of the leading edge of p_d_strb and will remain active for three SBus clocks beyond the end of p_ack. The position of p_ack relative to the trailing edge of data strobe is defined by DSS (again DSS has a tolerance of +3 to 4 SBus clocks). The width of p_ack is set using DSW. The operation of the interface as defined assumes the bidirectional sense of each signal has been configured as follows: DIR=1, DS_DSEL=1, ACK_DSEL=1, BUSY_DSEL=1. Reference the following data transfer diagram:



1. Acknowledge position relative to data strobe. (**DSS**-Hardware Configuration Register).
2. Acknowledge width (**DSW**-Hardware configuration register).
3. p_bsy is deasserted 3 sbus clocks following the trailing edge of p_ack.
4. All signal polarities shown are at the pins. Polarities on the interface cable should be inverted (except p_data).

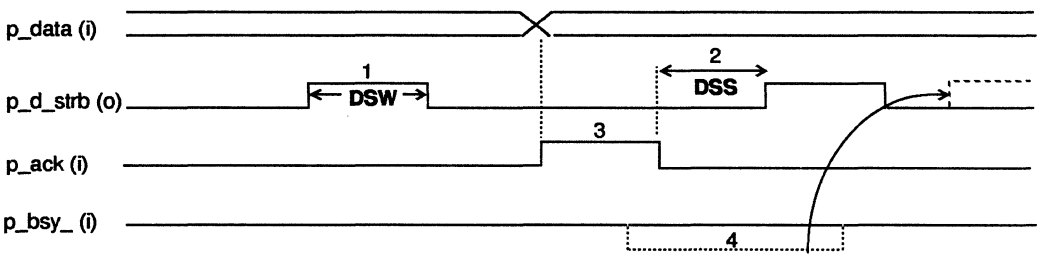
Master Read/Write Protocol (Xerox Mode)

This section describes the parallel port operation while master read cycles are performed. Operation while master write cycles are performed is the same as is described in the *Unidirectional Operation* section.

Data transfer for master read cycles is accomplished by the master generating a data strobe (request for data) with no data present on the p_data bus. The peripheral responds by placing data on the p_data bus and generating an p_ack which functions as a strobe. Only one handshake protocol is valid for master read cycles. It described below.

Handshake with Ack (BUSY_OP=0, ACK_OP=1)

Data is transferred to the parallel port by the use of p_ack. p_d_strb width is defined by DSW. DSS is used to define the required interval from p_ack to the next p_d_strb. p_bsy_ will gate further data transfers if present. The operation of the interface as defined assumes the bidirectional sense of each signal has been configured as follows: DIR=1, DS_DSEL=0, ACK_DSEL=0, BUSY_DSEL=0. Refer to the following data transfer diagram:



1. Data strobe width as defined in the hardware configuration register.
2. **DSS** is used for ack to p_d_strb timing (hardware configuration register).
3. Acknowledge is used as a strobe and is required for each byte transferred.
4. If p_bsy_ is active it gates further data transfers.
5. All signal polarities shown are at the pins. Polarities on the interface cable should be inverted (except p_data).

Programmed I/O Mode

Programmed I/O mode is intended to allow the parallel port to operate primarily under software control. Data latching, interrupt and busy generation are performed in hardware as required. The following two sections describe operation for transfers to and from the peripheral device.

PIO on Transfers To the Peripheral Device

For transfers to the peripheral device, all signals are under the control of software. There is no hardware assist other than interrupt generation.

PIO on Transfers From the Peripheral Device

The two modes of bidirectional operation previously discussed are supported with hardware assisted data latching. The bidirectional select bits (DS_DSEL, ACK_DSEL, BUSY_DSEL) should be set according to the desired configuration. The handshake protocol bits (ACK_OP, BUSY_OP) have no function in PIO mode.

During operation as a slave under the master write protocol (DS_DSEL=1, DIR=1), data is sampled and latched once data strobe has been detected. p_bsy_ becomes active at the same time that data is latched and must be made inactive under software control.

During operation under master read/write protocol (DS_DSEL=0, DIR=1), master reads are assisted by sampling and latching the data once p_ack has been detected. p_bsy_ is not generated in this mode.

Bidirectional Parallel Port Registers

The parallel port register set consists of two 16-bit configuration registers, four 8-bit parallel port interface registers, and one 16-bit interrupt control register. The two 16-bit configuration registers can be accessed as individual 16 bit registers or as a word aligned pair. Similarly, the four 8-bit registers can be accessed as 8-bit registers, 16-bit halfword aligned pairs, or as one 32-bit register. The Addr field of the register definitions corresponds to SBus physical address bits 4:0. Additionally, to access these registers the pa(x, y, z) physical address bits must be 101, respectively.

Parallel Port Configuration Registers

Hardware Configuration Register (P_HCR)

This 16-bit read/write register is used to specify the value of the two programmable timers. Reset value of this register is unknown, except the TEST bit which is reset to a 0.

ADDR=10, Size=16 Bit

Bit	Mnemonic	Description	Type
6:0	DSS	Data setup before data strobe in increments of 1 SBus clock.	R/W
7	—	Unused. Reads as 0.	RO
14:8	DSW	Data strobe width in increments of 1 SBus clock.	R/W
15	TEST	Test bit which when set, allows the buried counters to be read.	R/W

DSS

Data setup to data strobe. This 7 bit quantity is used to define several different timing specifications for the interface. The contents of this field of the register are used to load a hardware timer whose timebase is the SBus clock. The programmability range is from a minimum of 0 SBus clocks to 127 SBus clocks. Bit 0 is the lsb and bit 6 is the msb. The sections on unidirectional and bidirectional transfers should be referenced for detailed information on the use of this timer.

DSW

Data strobe width. This 7 bit quantity is used to define data strobe and acknowledge pulse widths for the interface. The contents of this field of the register are used to load a hardware timer whose timebase is the SBus clock. The programmability range is from a minimum of 3 SBus clock to 127 SBus clocks. In the case of the value being 0,1,2,or 3, the timer will be loaded with a value of 3. Bit 8 is the lsb and bit 14 is the msb. For detailed information on the use of this timer, refer to the sections on unidirectional and bidirectional transfers.

Operation Configuration Register (P_OCR)

This 16-bit read/write register is used to specify the operation of the interface. Bidirectional specification of the control signals (p_d_strb, p_ack, p_bsy_), handshake protocol, memory clear and diagnostic mode are defined in this register. The detailed function of the bits is described following the table. Reset value of this register is all bits 0, except DS_DSEL & IDLE, which are reset to 1, and bit 1, which always reads as 1.

ADDR=12 Size=16 Bit

Bit	Mnemonic	Description	Type
0	—	Reserved	R/W
1	—	Reserved. Reads as 1.	RO
2	—	Unused. Reads as 0.	RO
3	IDLE	Reads as one when the PP data transfer state machines are idle.	RO
4	—	Unused. Reads as 0.	RO
5	—	Unused. Reads as 0.	RO
6	—	Unused. Reads as 0.	RO
7	SRST	When set, resets the parallel port. Must be reset by software.	R/W
8	ACK_OP	Acknowledge operation.	R/W
9	BUSY_OP	Busy operation.	R/W
10	EN_DIAG	When set, enables diagnostic mode.	R/W
11	ACK_DSEL	Acknowledge bidirectional select. When set, ack_ is bi-dir.	R/W
12	BUSY_DSEL	Busy bidirectional select. When set, p_bsy_ is bi-dir.	R/W

Bit	Mnemonic	Description	Type
13	DS_DSEL	Data Strobe bidirectional select.When set, p_d_strb is bi-dir.	R/W
14	DATA_SRC	Data source for memory clear operation.	R/W
15	MEM_CLR	When set, enables memory clear.	R/W

IDLE
When this bit is set, it indicates that the parallel port data transfer state machines are in their idle states. The state machines should be idle when changing direction and/or configuring operational modes and when enabling a memory clear operation.

SRST
Setting this bit will place the parallel port data transfer state machines and programmable timers into reset. It will not reset any of the parallel port registers. This bit must be reset by software.

ACK_OP
Used to specify the handshake protocol to be used on the interface. The meaning of this bit differs depending on the direction of transfer. Refer to the sections on unidirectional and bidirectional transfers for detailed information on this bit. The general definition is as follows:

1 = Handshake using p_ack.

0 = p_ack is inactive.

BUSY_OP
Used to specify the handshake protocol to be used on the interface. The meaning of this bit differs depending on the direction of transfer. Refer to the sections on unidirectional and bidirectional transfers for detailed information on this bit. The general definition is as follows:

1 = Handshake performed using p_bsy_.

0 = p_bsy_ is not used for handshaking.

EN_DIAG
Setting this bit enables diagnostic mode which does three things:

1. Bits 0-2 of the output register are gated on to bits 0-2 of the input register. This allows testing of the data path and the interrupt generation logic.

2. The internal DS, ACK, and BUSY latch bits drive the internal DS_IRQ and ACK_IRQ, and BUSY_IRQ interrupt generation logic.

3. When reading the DS, ACK, and BUSY bits of the Transfer Control Register, the read data comes from the internal latches instead of the external pins. During diagnostic mode, if the DS or ACK bits are configured as outputs, the output pins will be gated to an inactive level. The BUSY- output will be driven active and the DIR output will be latched in its current state.

ACK_DSEL
This bit is a bidirectional select for the p_ack signal. When reset, p_ack is fixed as an input. When set, p_ack is a bidirectional signal. The p_ack_dir_ pin will reflect the direction of p_ack. The switching of direction is controlled by the DIR bit of the Transfer Control Register. The function of the two pins is as follows:

	DIR = 0	DIR = 1
p_ack	Input	Output
p_ack_dir	0	1

BUSY_DSEL
This bit is a bidirectional select for the p_bsy_ signal. When reset, p_bsy_ is fixed as an input. When set, p_bsy_ is a bidirectional signal. The p_bsy_dir_ pin will reflect the direction of p_bsy_. The switching of direction is controlled by the DIR bit of the Transfer Control Register. The function of the two pins is as follows:

	DIR = 0	DIR = 1
p_bsy	Input	Output
p_bsy_dir_	0	1

DS_DSEL
This bit is a bidirectional select for the p_d_strb signal. When reset, p_d_strb is fixed as an output. When set, p_d_strb is a bidirectional signal. The p_ds_dir_ pin will reflect the direction of p_d_strb. The switching of direction is controlled by the DIR bit of the Transfer Control Register. The function of the two pins is as follows:

	DIR = 0	DIR = 1
p_d_strb	Output	Input
p_ds_dir_	1	0

This bit also defines transfer protocol as follows:

1 = Data strobe is bidirectional. Master write transfer protocol is selected.

0 = Data strobe is fixed as an output. Master read/write transfer protocol is selected.

DATA_SRC
This bit specifies the data to be sourced during a memory clear operation. When set, the sourced data will be ones. When reset, the sourced data will be zeros.

MEM_CLR
This bit enables memory clear operation. Additionally, the DMA control registers need to be configured and DMA must be enabled.

Parallel Port Interface Registers

Parallel Data Register (P_DR)
The Data Register is an 8-bit read/write port used to transfer data to and from the external device. In programmed I/O mode data written to this register is presented to the I/O pins if the DIR bit of the Transfer Control Register is 0. A read of this register will result in the data previously written or if the DIR bit of the Transfer Control Register is set to 1, the latched data from the last data strobe. The data port is not accessible via slave write cycles during DMA (P_DMA_ON=1). Any write cycles

during DMA will not generate errors, the data will simply not be written.Since both DMA and PIO share the same data register, internal loopback is possible by running a single byte DMA cycle followed by a PIO cycle to verify the data. This will test both the DMA and slave data paths.

The reset state of this register is undefined.

ADDR=14, Size=8 Bit

Addr	Bits	Description	Size	Type
14	7:0	Parallel Data	8 Bit	R/W

Transfer Control Register (P_TCR)

The Transfer Control Register is an 8-bit register whose contents define/reflect the state of the external interface handshake and direction control signals. The DS, ACK, and BUSY- bits will reflect the state of the external pins, when read. Writing these bits defines a value to be driven onto the external pins if the individual direction select bits (DS_DSEL, ACK_DSEL, BUSY_DSEL) and the direction control bit (DIR) are configured such that these pins are outputs. The write bits and read bits are different. That means that values typically written to these bits may not be reflected on a read cycle. However, by setting the EN_DIAG bit of the Operation Control Register, these register bits become read/write (see the EN_DIAG bit description of the OCR).

ADDR=15, Size=8 Bit

Bit	Mnemonic	Description	Type
0	DS	Data Strobe.	R/W
1	ACK	Acknowledge.	R/W
2	BUSY	Busy (active low).	R/W
3	DIR	Direction Control. 0= Write to external Device. 1=Read.	R/W
4	—	Unused. Reads as 0.	RO
5	—	Unused. Reads as 0.	RO
6	—	Unused. Reads as 0.	RO
7	—	Unused. Reads as 0.	RO

DS

Reading this bit reflects the state of the bidirectional p_d_strb pin. Writing this bit with DS_DSEL=0 or with DS_SEL=1 and DIR=0 will cause the value written to be driven onto p_d_strb. The reset state of the output latch is 0, but the value read back from this bit after reset will reflect the input signal being driven onto p_d_strb.

ACK

Reading this bit reflects the state of the bidirectional p_ack pin. Writing this bit with ACK_DSEL=1 will cause the value written to be driven onto p_ack if DIR=1. The reset state of the output latch is 0, but the value read back from this bit after reset will reflect the input signal being driven onto p_ack.

BUSY
Reading this bit reflects the state of the bidirectional p_bsy_ pin. Writing this bit with BUSY_DSEL=1 will cause the value written to be driven onto p_bsy_ if DIR=1. The reset state of the output latch is 0, but the value read back from this bit after reset will reflect the input signal being driven onto p_bsy_.

DIR
This bit defines and controls the direction of data transfer: 0=write to external device, 1= read from external device. It is also driven externally on the p_d_dir_ pin. Note that this bit also controls the direction of DMA (except in the case of a memory clear operation, when the MEM_CLR bit defines DMA direction). The state of the DIR bit is reflected in the P_WRITE bit of the P_CSR. Reset state of this bit is 1.

Output Register (P_OR)
The Output Register is an 8-bit read/write register whose contents are driven onto the corresponding external pins. In diagnostic mode (EN_DIAG=1), bits 0-2 are gated onto Input Register bits 0-2. The external outputs remain low while diagnostic mode is enabled. All bits are 0 after reset.

ADDR=16, Size=8 Bit

Bit	Mnemonic	Description	Type
0	SLCT_IN	Select in. This bit is output on the p_slct_in pin.	R/W
1	AFXN	Auto Feed. This bit is output on the p_afxn pin.	R/W
2	INIT	Initialize. This bit is output on the p_init pin.	R/W
3	—	Reserved	R/W
4	—	Reserved	R/W
5	—	Reserved	R/W
6	—	Unused. Reads as 0	RO
7	—	Unused. Reads as 0	RO

Input Register (P_IR)
The Input Register is an 8-bit read/write register whose contents reflect the state of several external input pins and their corresponding interrupts. In diagnostic mode (**EN_DIAG**=1), bits 0-2 are driven from output register bits 0-2.

ADDR=17, Size=8 Bit

Bit	Mnemonic	Description	Type
0	ERR	Error input. This bit reflects the state of the err input pin.	RO
1	SLCT	Select input. This bit reflects the state of the slct input pin.	RO
2	PE	Paper empty. This bit reflects the state of the p_pe_ input pin.	RO
3	—	Unused. Reads as 0	RO

Bit	Mnemonic	Description	Type
4	—	Unused. Reads as 0	RO
5	—	Unused. Reads as 0	RO
6	—	Unused. Reads as 0	RO
7	—	Unused. Reads as 0	RO

ERR, SLCT, PE
These bits reflect the state of the corresponding input pins.

Interrupt Control Register

This 16-bit read/write register is used to specify operation of the parallel port interrupts. Interrupt enables, polarity, and IRQ pending bits are contained in this register. The detailed function of these bits is described following the table. Reset value of this register is all bits 0.

ADDR=18 Size=16 Bit

Bit	Mnemonic	Description	Type
0	ERR_IRQ_EN	When set, enables ERR interrupts.	R/W
1	ERR_IRP	ERR interrupt polarity. 1=on rising edge, 0=on trailing edge.	R/W
2	SLCT_IRQ_EN	When set, enables SLCT interrupts.	R/W
3	SLCT_IRP	SLCT interrupt polarity. 1=on rising edge, 0=on trailing edge.	R/W
4	PE_IRQ_EN	When set, enables PE interrupts.	R/W
5	PE_IRP	PE interrupt polarity. 1=on rising edge, 0=on trailing edge.	R/W
6	BUSY_IRQ_EN	When set, enables BUSY interrupts.	R/W
7	BUSY_IRP	BUSY interrupt polarity. 1=on rising edge, 0=on trailing edge.	R/W
8	ACK_IRQ_EN	When set, enables ACK interrupts on rising edge of ack.	R/W
9	DS_IRQ_EN	When set, enables DS interrupts on the rising edge of ds.	R/W
10	ERR_IRQ	When set, error IRQ pending.	R/W1
11	SLCT_IRQ	When set, select IRQ pending.	R/W1
12	PE_IRQ	When set, paper empty IRQ pending.	R/W1
13	BUSY_IRQ	When set, busy IRQ pending.	R/W1
14	ACK_IRQ	When set, acknowledge IRQ pending.	R/W1
15	DS_IRQ	When set, data Strobe IRQ pending.	R/W1

***_IRQ_EN**
When set, enables interrupts on the corresponding bits of the Input and Transfer Control registers. Note that the interrupt enable bit of the PD_SCR must also be enabled to allow a hardware interrupt to be generated.

***_IRP**
Defines the polarity of the edge triggered interrupts on the corresponding bits of the Input Register. 0=Interrupt generated on the 1 to 0 transition of the signal. 1= Interrupt generated on the 0 to 1 transitions of the signal. Note that when configuring the interrupt polarity of a given signal, it is possible to generate a false interrupt. It is suggested that when the interrupt polarities are being programmed, interrupts be disabled and all interrupt sources be cleared upon completion of programming.

ERR_IRQ, SLCT_IRQ, PE_IRQ, BUSY_IRQ
When set, an interrupt is pending on the corresponding bit. The interrupt is cleared and the bit is reset when a one is written to the corresponding bit. Writing a 0 to these locations leaves the bit(s) unchanged.

ACK_IRQ
When set, an interrupt is pending due to the receipt of p_ack. The interrupt is set on the 0 to 1 transition of p_ack. This interrupt is intended to facilitate PIO transfers while configured as master under master write protocol.The interrupt is cleared and the bit is reset when a one is written to this bit. Writing a 0 to this location leaves the bit unchanged.

DS_IRQ
When set, an interrupt is pending due to the receipt of p_d_strb. This interrupt is intended to facilitate PIO transfers while configured as slave under master write protocol.The interrupt is cleared and the bit is reset when a one is written to this bit. Writing a 0 to this location leaves the bit unchanged.

Parallel Port Programming Notes

Parallel Port Interrupts

The P_INT_PEND and P_ERR_PEND bits of the P_CSR reflect whether the PP has an interrupt source active or not. This will allow software to determine whether an interrupt is pending on the PP by performing one register read. If P_INT_PEND is set, then one or more of the following interrupt sources are active:

Parallel Port Control/Status Register (P_CSR):

- P_TC – Terminal Count.

Parallel Port Interrupt Control Register (P_ICR):

- ERR_IRQ – Error
- SLCT_IRQ – Select
- PE_IRQ – Paper Error
- BUSY_IRQ – Busy
- ACK_IRQ – Acknowledge
- DS_IRQ – Data Strobe

If the P_ERR_PEND bit is set, the P_RESET must be toggled to reset the parallel port to recover from the error.

Mode Selection

When switching modes of operation of the parallel port, the P_DMA_ON bit should be 0 and the IDLE bit of the Transfer Control Register should be 1 before the mode is switched. This insures that no data transfer cycles are in progress. "Mode Selection" in this context means handshake specification, control signal direction selection, interface direction selection via the DIR bit of the Transfer Control Register, and memory clear operations.

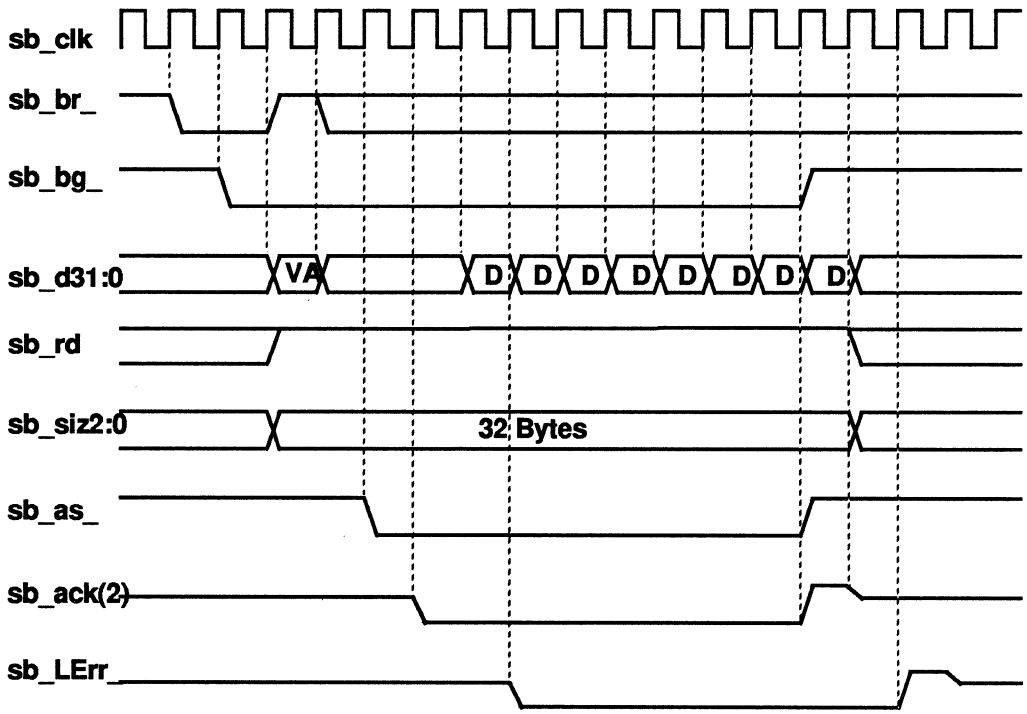
Timing Diagrams

NOTE: These timing diagrams illustrate relationships between various DMA2 signals and SBus clock pulses. Also note that there are no diagrams for slave (register) accesses to the SCSI in "slow" mode (fast/slow_ pin = 0). For now, these cycles are described in terms of their corresponding "fast" cycles:

The SCSI register write cycle in "slow" mode is the same as it is in "fast" mode except that the d_cs_ pulse is extended by three SBus clocks and the d_wr_ pulse is extended by two SBus clocks.

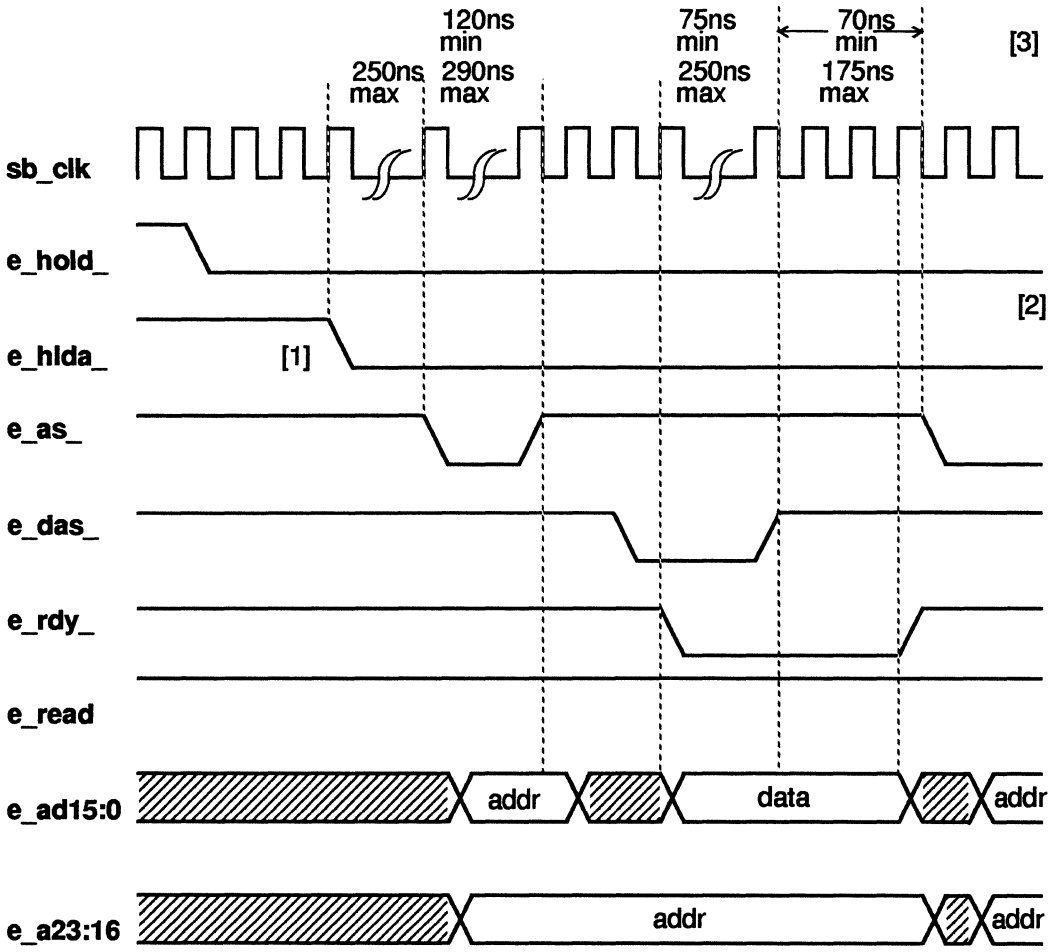
The SCSI register read cycle in "slow" mode is the same as it is in "fast" mode except that the d_cs_ pulse is extended by three SBus clocks and the d_rd_ pulse is extended by three clocks.

Note that for both of these cycles, the sb_ack(1) pulse is delayed by the same time that the d_rd_ or d_wr_ pulse is extended.



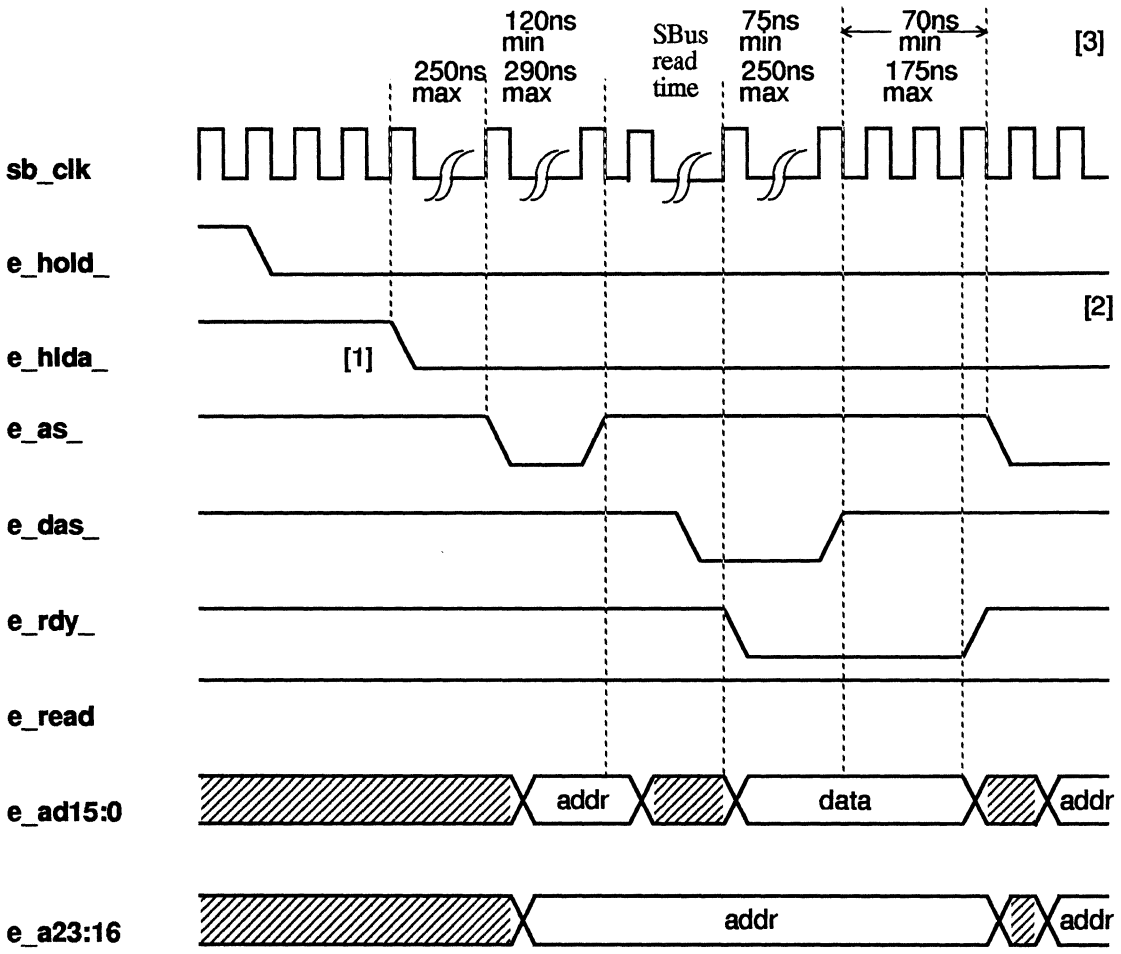
SBus DMA burst read (1 word/clk)
(Data flows from memory to the DMA2)

For other possible SBus cycles, see the SBus specification; this is just one example.



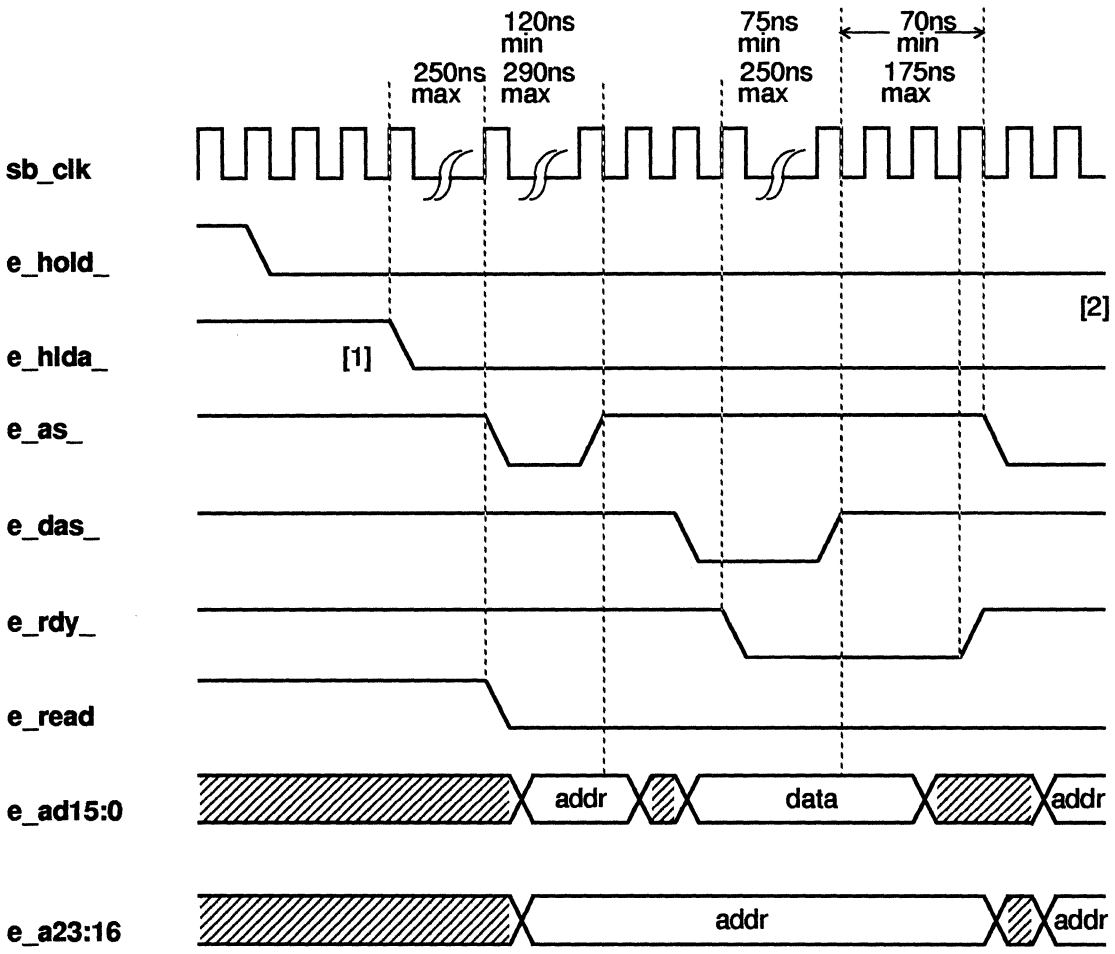
[1] e_hlda_ is only asserted when the interface is not busy.
[2] e_hold_ will stay asserted for burst mode accesses. e_hlda_ must follow it.

Ethernet DMA burst read (from memory) with valid data in E-cache



[1] e_hlda_ is only asserted when the interface is not busy.
[2] e_hold_ will stay asserted for burst mode accesses. e_hlda_ must follow it.

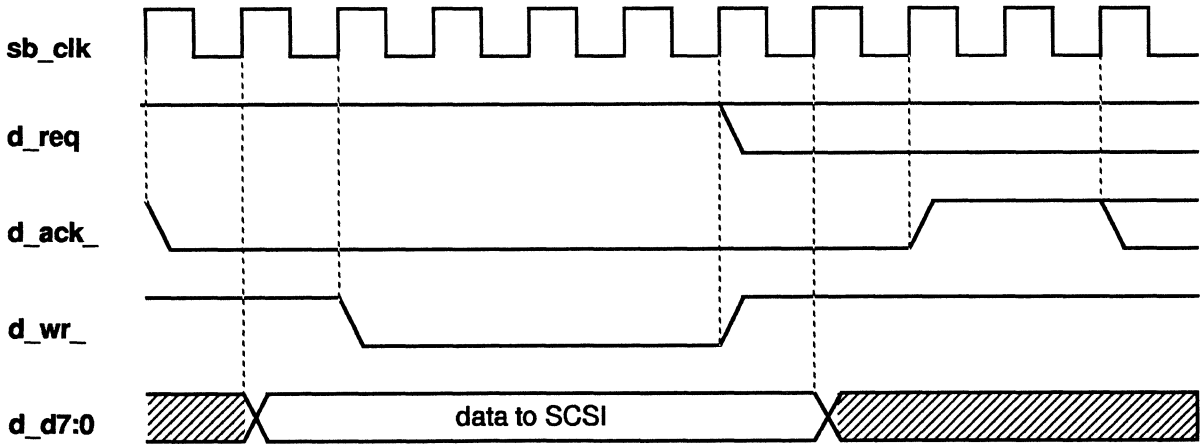
Ethernet DMA burst read (from memory) with no data inE-cache



[1] e_hlda_ is only asserted when the interface is not busy.
[2] e_hold_ will stay asserted for burst mode accesses. e_hlda_ must follow it.

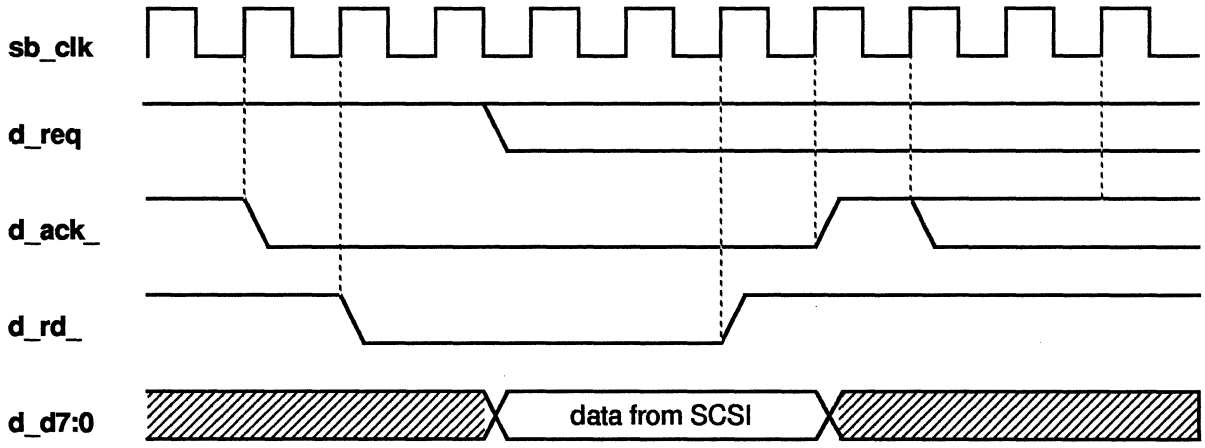
Ethernet DMA burst write (to memory)

NOTE: assumes a cache line is available for the ENET to write to



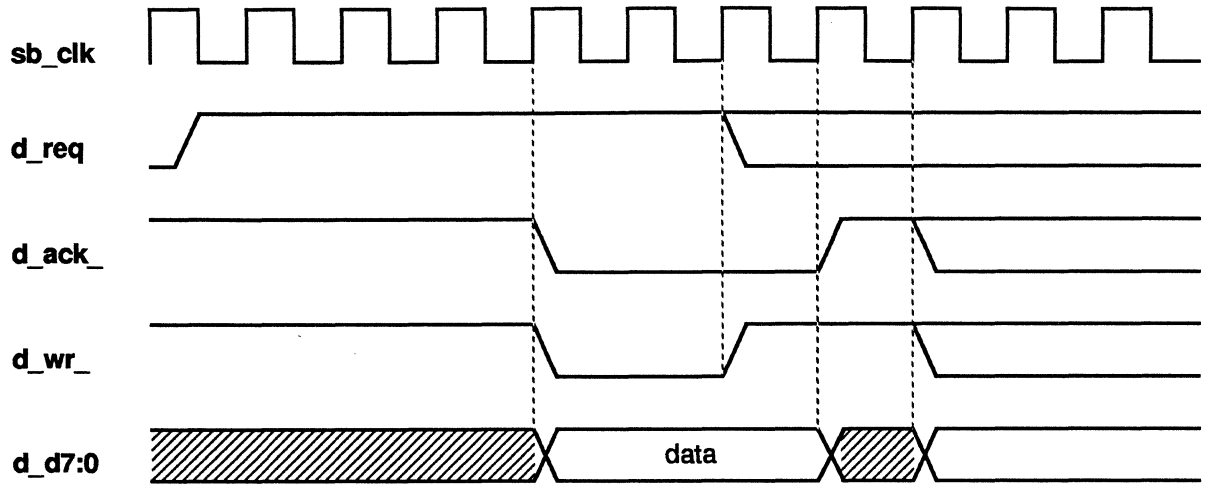
SCSI DMA read (from memory)

[fast/slow_ pin = 0V, D_FASTER = "don't care", D_TWO_CYCLE = "don't care"]



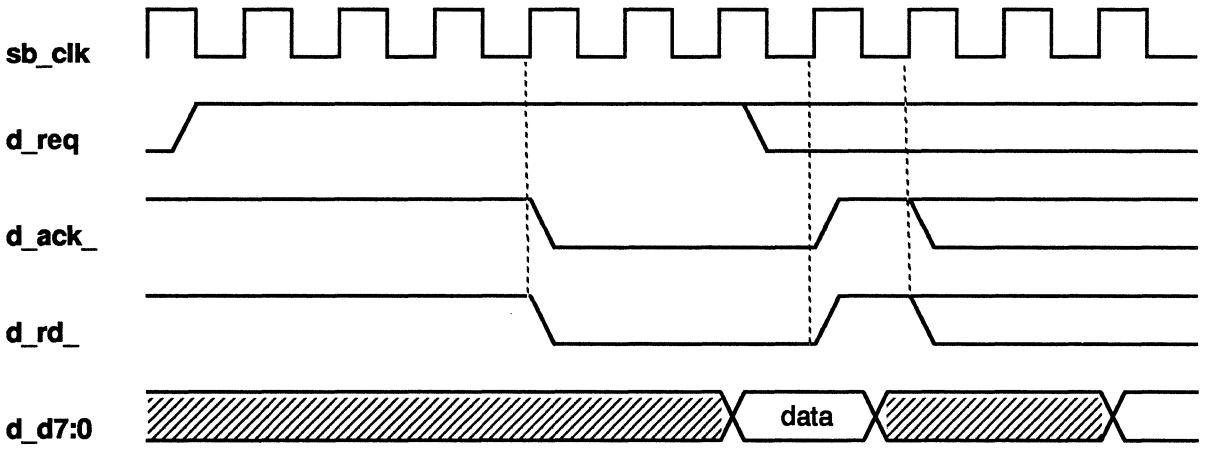
SCSI DMA write (to memory)

[fast/slow_ pin = 0V, D_FASTER = "don't care", D_TWO_CYCLE = "don't care"]



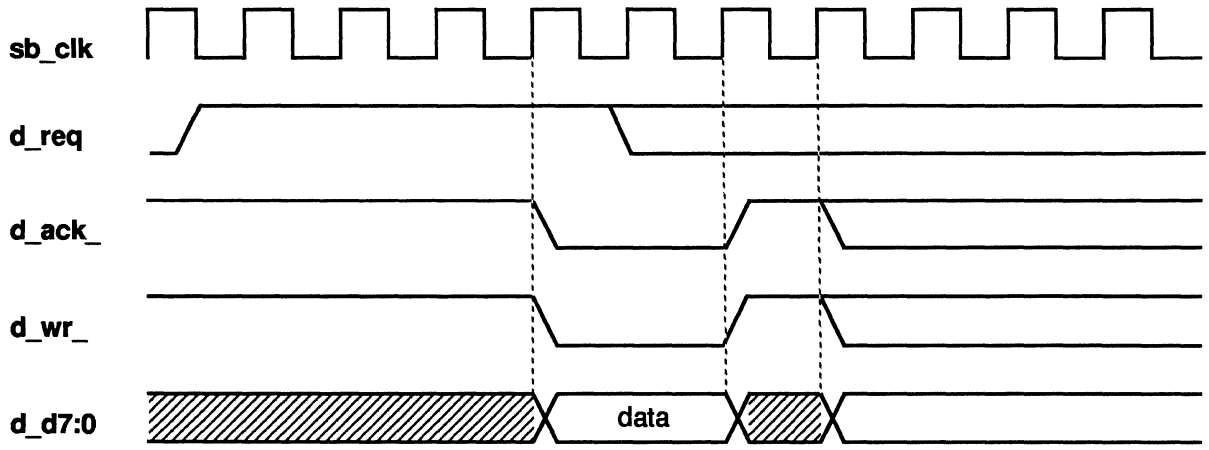
SCSI DMA read (from memory) with valid data in D-cache

[fast/slow_ pin = +5V, D_FASTER = 0, D_TWO_CYCLE = 0]



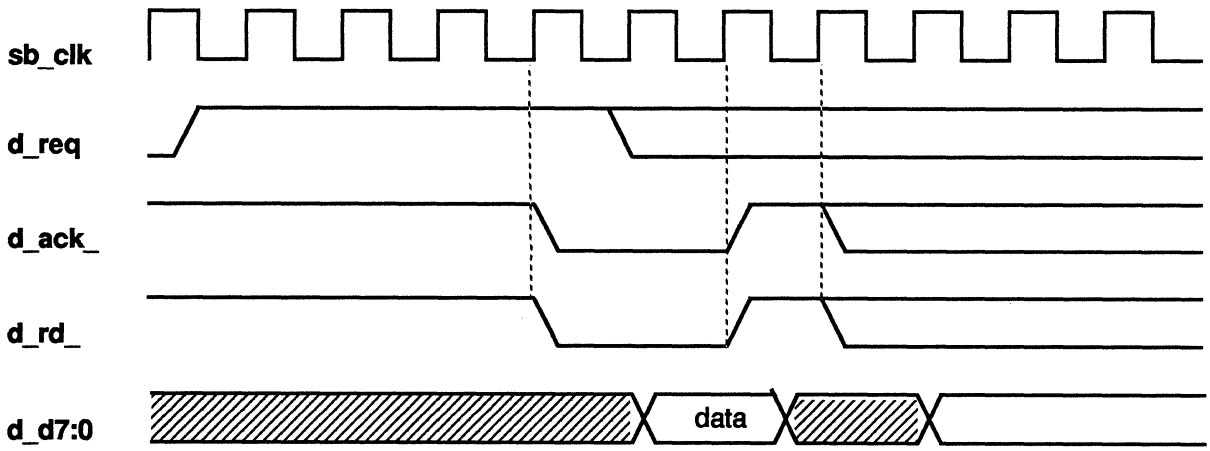
SCSI DMA write (to memory)

[fast/slow_ pin = +5V, D_FASTER = 0, D_TWO_CYCLE=0]



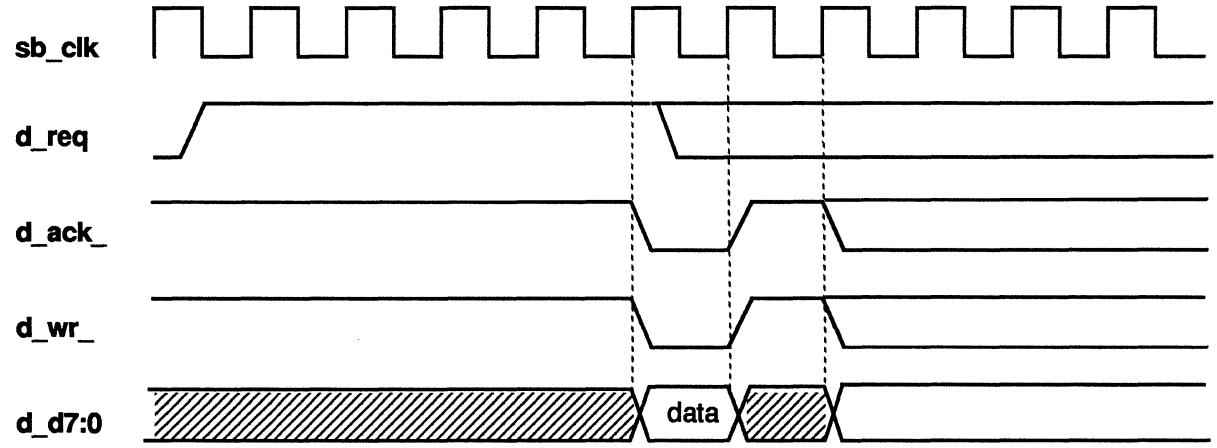
SCSI DMA read (from memory) with valid data in D-cache

[fast/slow_ pin = +5V, D_FASTER = 1, D_TWO_CYCLE = 0]



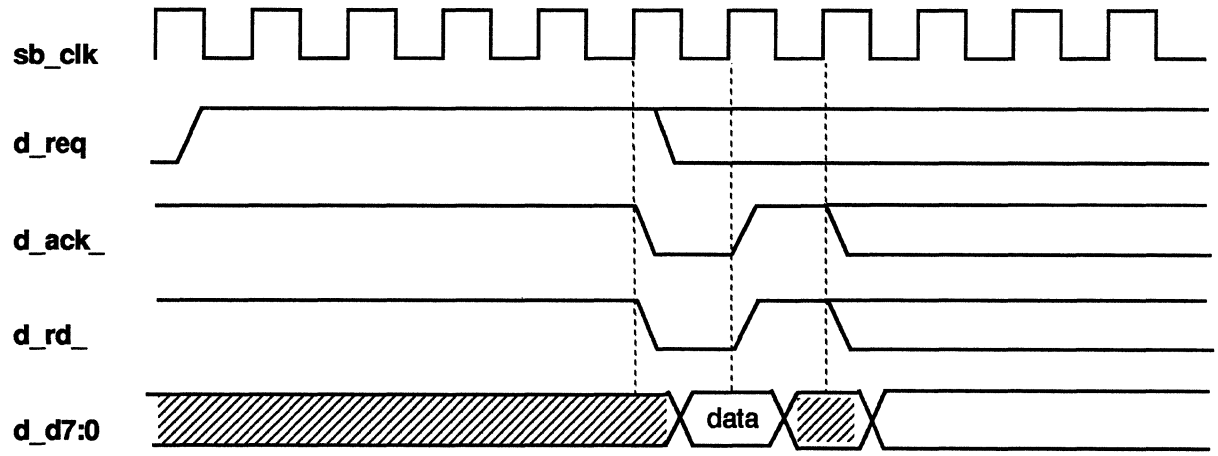
SCSI DMA write (to memory)

[fast/slow_ pin = +5V, D_FASTER = 1, D_TWO_CYCLE = 0]



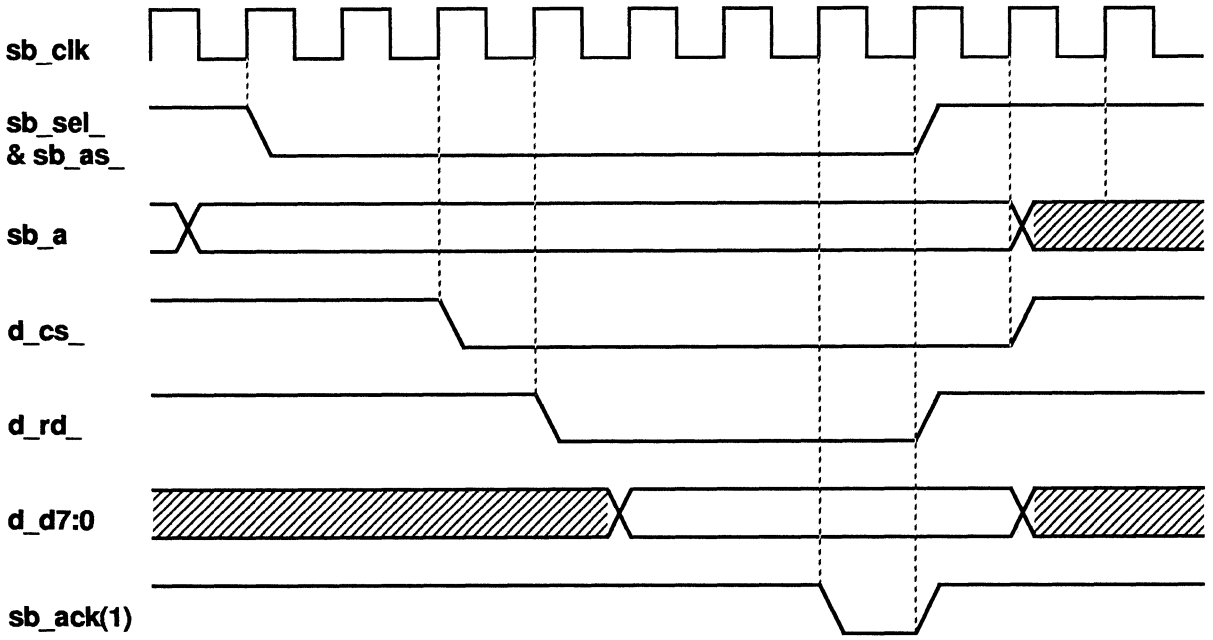
SCSI DMA read (from memory) with valid data in D-cache

[fast/slow_ pin = +5V, D_FASTER = 0, D_TWO_CYCLE = 1]

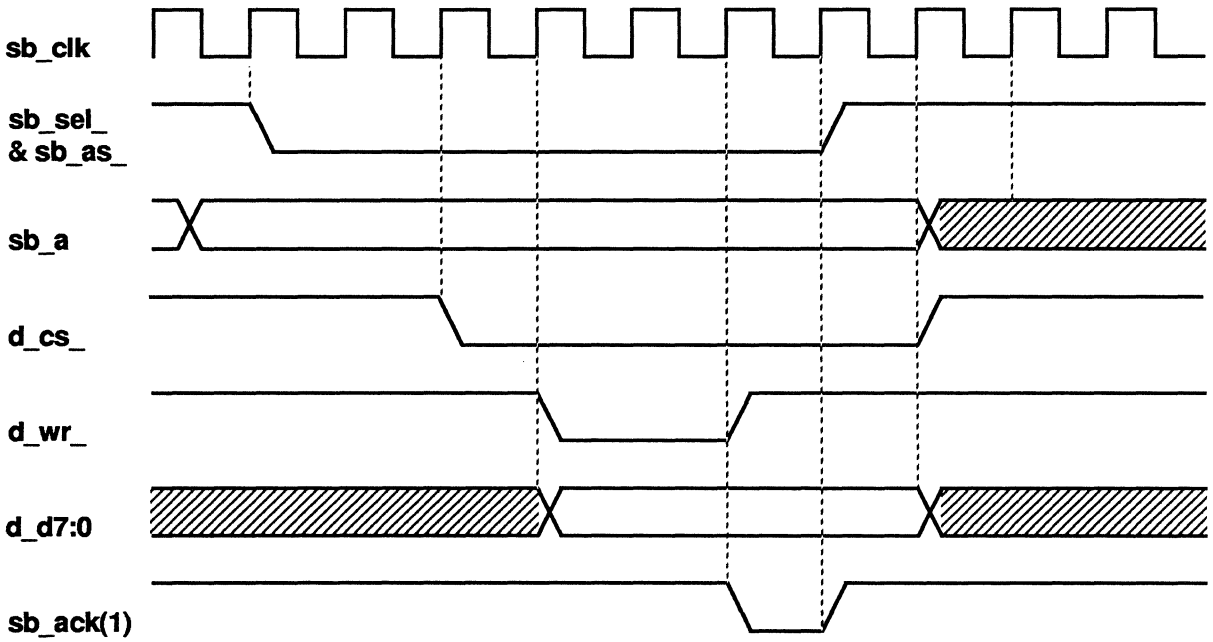


SCSI DMA write (to memory)

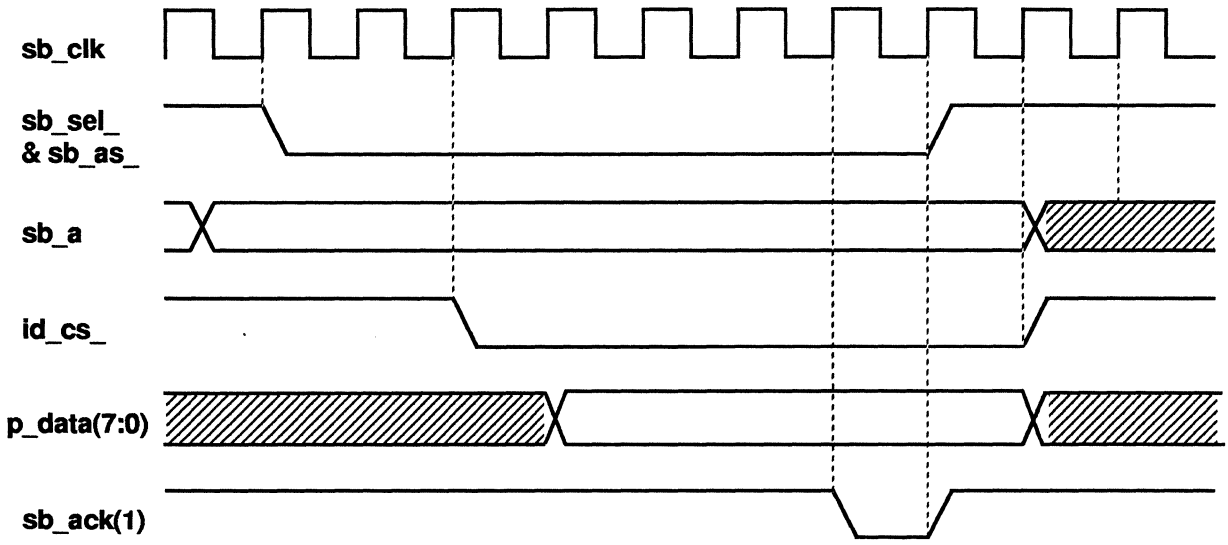
[fast/slow_ pin = +5V, D_FASTER = 0, D_TWO_CYCLE = 1]



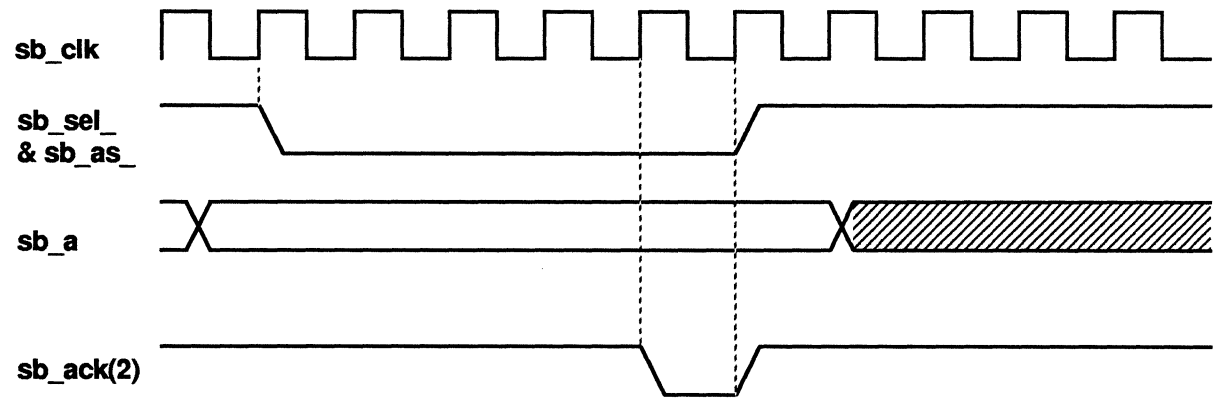
SCSI register read (slave cycle) [fast_slow_ pin = +5V]



SCSI register write (slave cycle) [fast_slow_ pin = +5V]

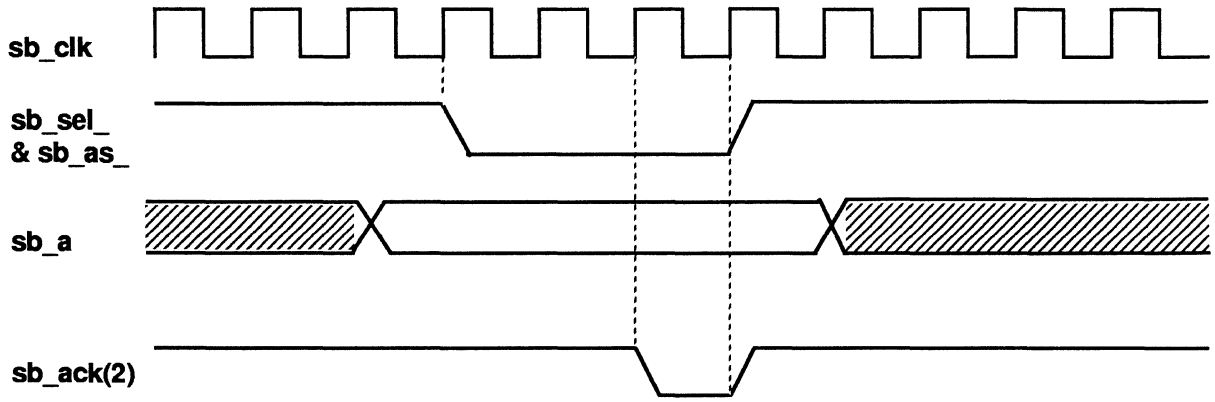


External ID PROM read (slave cycle)



Internal register slave access (D & E interfaces)

For D_CSR, D_ADDR, D_BCNT, D_TST_CSR, E_CSR, E_TST_CSR, E_VLD, & E_BASE_ADDR.
(See SBus Spec for timing of SBus signals not shown here.)



Internal register slave access (P Interface)

For P_CSR, P_ADDR, P_BCNT, P_TST_CSR, P_HCR, P_OCR, P_DR, P_TCR, P_OR, P_IR, & P_ICR
(See SBus Spec for timing of SBus signals not shown here.)

NCR92C990 Ethernet/IEEE 802.3 LAN Controller
Contents

NCR92C990 Ethernet/IEEE 802.3 LAN Controller

Features

- Register compatible with Am7990, rev. F
- Supports the IEEE 802.3 standard for 10BASE5 type A, and 10BASE2 type B
- Compatible with Am7992 Manchester Encoder/Decoder
- Multiplexed address/data bus (16-bit data bus and 24-bit linear address)
- Dual 48-byte internal FIFOs (one for transmit and one for receive)
- Data byte swapping (8086/68000 microprocessors)
- Programmable Address Latch Enable (ALE)
- Supports DMA bus master and slave modes
- Transmit-/receive-based collision detection and recovery
- Network and packet error reporting
- Diagnostics
- Internal/external loopback
- CRC
- Time Domain Reflectometry (TDR)
- Submicron CMOS cell-based technology
- ASIC macrocell core for customized designs
- Available in a 68-pin PLCC

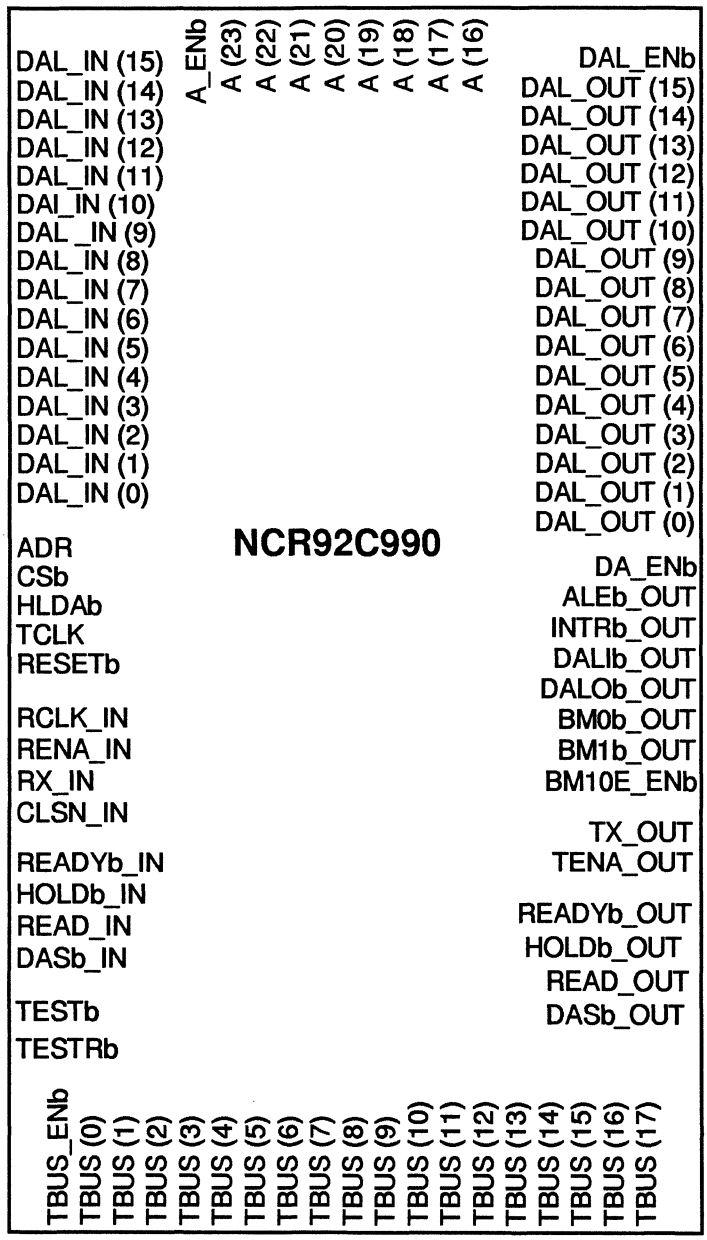


Figure 4-1 Symbol

Signal Name	Type	Description
DASb_IN	I	Data strobe
DASb_OUT*	O	Data strobe
DALIb_OUT*	O	External bus transmission control
DALOb_OUT*	O	External bus transmission control
DAL_IN (0)	I	Input data bus
DAL_IN (1)	I	Input data bus
DAL_IN (2)	I	Input data bus
DAL_IN (3)	I	Input data bus
DAL_IN (4)	I	Input data bus
DAL_IN (5)	I	Input data bus
DAL_IN (6)	I	Input data bus
DAL_IN (7)	I	Input data bus
DAL_IN (8)	I	Input data bus
DAL_IN (9)	I	Input data bus
DAL_IN (10)	I	Input data bus
DAL_IN (11)	I	Input data bus
DAL_IN (12)	I	Input data bus
DAL_IN (13)	I	Input data bus
DAL_IN (14)	I	Input data bus
DAL_IN (15)	I	Input data bus
DAL_OUT (0)	O	Output data bus
DAL_OUT (1)	O	Output data bus
DAL_OUT (2)	O	Output data bus
DAL_OUT (3)	O	Output data bus
DAL_OUT (4)	O	Output data bus
DAL_OUT (5)	O	Output data bus
DAL_OUT (6)	O	Output data bus
DAL_OUT (7)	O	Output data bus
DAL_OUT (8)	O	Output data bus
*Uses DA_Enb for Enable.		

Signal Name	Type	Description
DAL_OUT (9)	O	Output data bus
DAL_OUT (10)	O	Output data bus
DAL_OUT (11)	O	Output data bus
DAL_OUT (12)	O	Output data bus
DAL_OUT (13)	O	Output data bus
DAL_OUT (14)	O	Output data bus
DAL_OUT (15)	O	Output data bus
DAL_ENb	O	Output data bus enable
BM0b_OUT*	O	Byte selection
BM1b_OUT	O	Byte selection (uses BM10E_ENb for enable)
BM10E_ENb	O	Byte selection enable
INTRb_OUT	O	Interrupt
DA_ENb	O	Bus control signals enable
ALEb_OUT*	O	Address latch enable
READYb_IN	I	Ready acknowledge
READYb_OUT	O	
HOLDb_IN	I	Hold request
HOLDb_OUT	O	
READ_IN	I	Read/Write
READ_OUT*	O	
RCLK_IN	I	Receive clock
RENA_IN	I	Receiver enable
RX_IN	I	Receive data
CLSN_IN	I	Collision
TX_OUT	O	Transmit data
TENA_OUT	O	Transmitter enable
ADR	I	Register address port
CSb	I	Chip select
HLDAb	I	Hold acknowledge
*Uses DA_Enb for Enable.		

Signal Name	Type	Description
TCLK	I	Transmit (system) clock
RESETb	I	Reset
A(16)	O	High order address bits
A(17)	O	High order address bits
A(18)	O	High order address bits
A(19)	O	High order address bits
A(20)	O	High order address bits
A(21)	O	High order address bits
A(22)	O	High order address bits
A(23)	O	High order address bits
A_ENb	O	High order address bits enable
TBUS_ENb	O	Test bus enable
TBUS (0)	O	Test bus
TBUS (1)	O	Test bus
TBUS (2)	O	Test bus
TBUS (3)	O	Test bus
TBUS (4)	O	Test bus
TBUS (5)	O	Test bus
TBUS (6)	O	Test bus
TBUS (7)	O	Test bus
TBUS (8)	O	Test bus
TBUS (9)	O	Test bus
TBUS (10)	O	Test bus
TBUS (11)	O	Test bus
TBUS (12)	O	Test bus
TBUS (13)	O	Test bus
TBUS (14)	O	Test bus
TBUS (15)	O	Test bus
TBUS (16)	O	Test bus
TBUS (17)	O	Test bus

Signal Name	Type	Description
TESTb	I	Test enable (active low)
TESTRb	I	Test enable (active low)

Input Signal Capacitance		Output Signal Drive Capability	
DASb_IN	.035 pF	DASb_OUT	HBUF
DAL_IN (15:0)	.035 pF	DAL_OUT (15:0)	HBUF
READYb_In	.035 pF	DAL_ENb	HBUF
HOLDb_IN	.035 pF	READYb_OUT	HBUF
READb_IN	.035 pF	HOLDb_OUT	HBUF
RCLK_IN	.035 pF	READb_OUT	HBUF
RENA_IN	.035 pF	DALIb_OUT	HBUF
RX_IN	.035 pF	DALOb_OUT	HBUF
CLSN_IN	.035 pF	INTRb_OUT	HBUF
ADR	.035 pF	BM1b_OUT	HBUF
CSb	.035 pF	BM0b_OUT	HBUF
HLDAb	.035 pF	BM10E_ENb	HBUF
TCLK	.035 pF	DA_ENb	HBUF
RESETb	.035 pF	ALEb_OUT	HBUF
TESTb	.035 pF	TX_OUT	HBUF
TESTRb	.035 pF	TENA_OUT	HBUF
		A (23:16)	HBUF
		A_ENb	HBUF
		TBUS (17:0)	HBUF
		TBUS_ENb	HBUF

Description

The NCR92C990 is a Local Area Network (LAN) controller that supports IEEE 802.3 10BASE5 type A and 10BASE-T type B applications. The NCR92C990 provides a 16-bit interface for use in LAN adapter architectures and system buses. The NCR92C990 is suitable in both low-cost slave architectures and shared memory applications. The NCR92C990 controls the network parameters of an IEEE 802.3 network and supports network management with network counters and status reports.

Designed with NCR's cell-based technology, the NCR92C990's compatibility significantly reduces the difficulty in interfacing a microcomputer or minicomputer to an IEEE 802.3 Ethernet LAN. DMA and slave modes facilitate addressing to 24 bits. Bus master and slave modes are supported. The NCR92C990 has two independent, 48-byte FIFOs (one for transmit and one for receive). Network diagnostics and CRC error reporting are also supported.

NCR offers EtherCore products that are sold as standard products for quick time-to-market designs. This family of EtherCore products is developed and supported by NCR's ASIC design tools. This allows designers to maintain control of cost/performance objectives and eliminate standard products that restrict design flexibility.

Macrocell Pin Descriptions

A(23:16) — High-Order Address Bits.
During the bus master address segment of a memory transfer, these outputs contain the most significant eight bits of the 24-bit address. A_ENb is the enable for A(23:16).

ADR — Register Address Port Select.
This input selects which internal register is in use during a slave mode access. If ADR = 1, the register address port is selected. If ADR = 0, the register data port is selected.

ALEb_OUT — Address Latch Enable/Address Strobe.
This output demultiplexes the address bus. ALE is programmed through the ACON bit of Control/Status Register 3.

BM0b_OUT and Bm1b_OUT — Byte Selection

BM0_OUT	BM1b_OUT	Mode
0	0	16-bit word
0	1	upper byte
1	0	lower byte
1	1	not used

CSb — Chip Select.
To select the NCR92C990 for slave access, drive this input low. CSb must remain active throughout the complete cycle.

CLSN_IN — Collision.
This input from the Manchester Encoder/ Decoder (MENDEC) indicates a collision was detected.

DAL_OUT(15:0) — DAL_IN(15:0) — Address/Data Line.
During the address segment of a memory transfer, this split bus contains the lower 16 bits of memory address. During data read/write segments, this split bus contains 16 bits of data. DAL_ENb is the enable for DAL_OUT(15:0).

DALIb_OUT — External data bus transceiver control.
This output is active in bus master mode only. It is asserted when DAL_OUT(15:0) is driven. DALIb_OUT = 1 during a write cycle. DALIb_OUT = 0 during the DATA segment of a read cycle.

DALOb_OUT — External data bus transceiver control.
This output is active in bus master mode only. It is asserted when DAL_OUT(15:0) is driven. DALOb_OUT = 0 during a write cycle and the ADDRESS segment of a read cycle.

DASb_OUT — Data Strobe.
This output identifies the data segment of the bus cycle. DASb_OUT = 0 during the WRITE data segment of the bus master transfer. DASb_OUT = 1 during the address segment of the bus master transfer. A low-to-high transition is used to latch WRITE data for a slave cycle.

DASb_IN — Data Strobe.
This input identifies the data segment of the bus cycle. DASb_IN = 0 during the READ data segment of the bus master transfer. DASb_IN = 1 during the address segment of the bus master transfer. A low-to-high transition is used to latch data READ.

HLDAb — Hold Acknowledge.
When this input, HLDAb = 0, in response to an assertion of HOLDb_IN, the NCR92C990 is the bus master. The controller waits for HLDAb to go high before reasserting HOLDb_OUT = 0.

HOLDb_OUT — Hold Request.
The NCR92C990 asserts this active low output during memory accesses. It stays = 0 for the entire burst. This pin is programmable through bit 0 of Control/Status Register 3.

HOLDb_IN — Hold Request Sense.
The NCR92C990 looks at HOLDb_IN input to sense the HOLDb condition. If HOLDb_IN = 1 the NCR92C990 can drive HOLDb_OUT = 0 to request the bus.

INTRb_OUT — Interrupt.
This active low output activates when any interrupts are generated according to the flags set in Control/Status Register 0. INTRb_OUT is disabled through bit 6 of Control/Status Register 0. (In which case INTRb_OUT will remain high.)

RCLK_IN — Receive Clock.
This input is the 10 MHz receive clock from the MENDEC.

READ_IN — Read Data In.
This input is used to define the read/write operation. For a slave cycle READ_IN = 0 to output data (write). READ_IN = 1 to input data (read).

Cycle	READ_IN	Type
Slave	0	Data output
	1	Data input

READ_OUT — Read Data Out.
This output is used to define the read/write operation. For a master cycle READ_OUT = 1 to input data (read), READ_OUT = 0 to output data (write).

Cycle	READ_OUT	Type
Bus Master	1	Data output
	0	Data input

READYb_IN — Ready (Master mode).
This active low input in a bus master mode is used to provide an asynchronous acknowledgment that a transfer can be performed. The READ_OUT signal determines whether a read or write transfer is taking place.

READYb_OUT — Ready (Slave mode).
Asserted active low output is generated in response to DASb_IN for a Slave read or write. The READ_IN signal determines whether a read or write transfer is taking place.

RENA_IN — Receive Enable.
This buffered input from the MENDEC signifies a carrier present.

RESETb — Reset.
This active low buffered input is a system reset for the NCR92C990 macrocell.

RX_IN — Receive Data.
This is the buffered receive data input from the MENDEC data output.

TBUS(17:0) — Test Bus Outputs.
These outputs are used for testing and are always driven.

TCLK — Transmit Clock.
This buffered input is the system clock from the MENDEC.

TENA_OUT — Transmitter Enable.
This is an output to the MENDEC to enable the MENDEC transmitter.

TESTb — Test Mode Enable.
This active low input is driven low to enable the NCR92C990 internal test features. TESTb should be driven high for normal operation.

TESTRb — Test Mode Enable.
This active low input is driven low to enable the NCR92C990 internal test features. TESTRb should be driven high for normal operation.

TX_OUT — Transmit Data.
This output is the transmit data to the MENDEC.

Functional Description

Bus Interface

The NCR92C990 is designed to be compatible with a variety of microprocessors. Internal logic for byte swapping enables different bus organizations to be supported. The programmable Address Latch Enable (ALE) output supports interfaces with different microprocessors and it is register programmable. The bus master mode supports 24-bit DMA with addressing and 16-bit programmable data byte addressing.

Bus Master Mode

Data is transferred to and from the NCR92C990 through DMA transfers. The data transfers are timed by using ALEb_OUT, DASb_IN, DASb_OUT, READYb_IN and READYb_OUT.

Slave Mode

The slave mode is active when the host asserts the CSb input and initiates a read or write to a register or to the register address port.

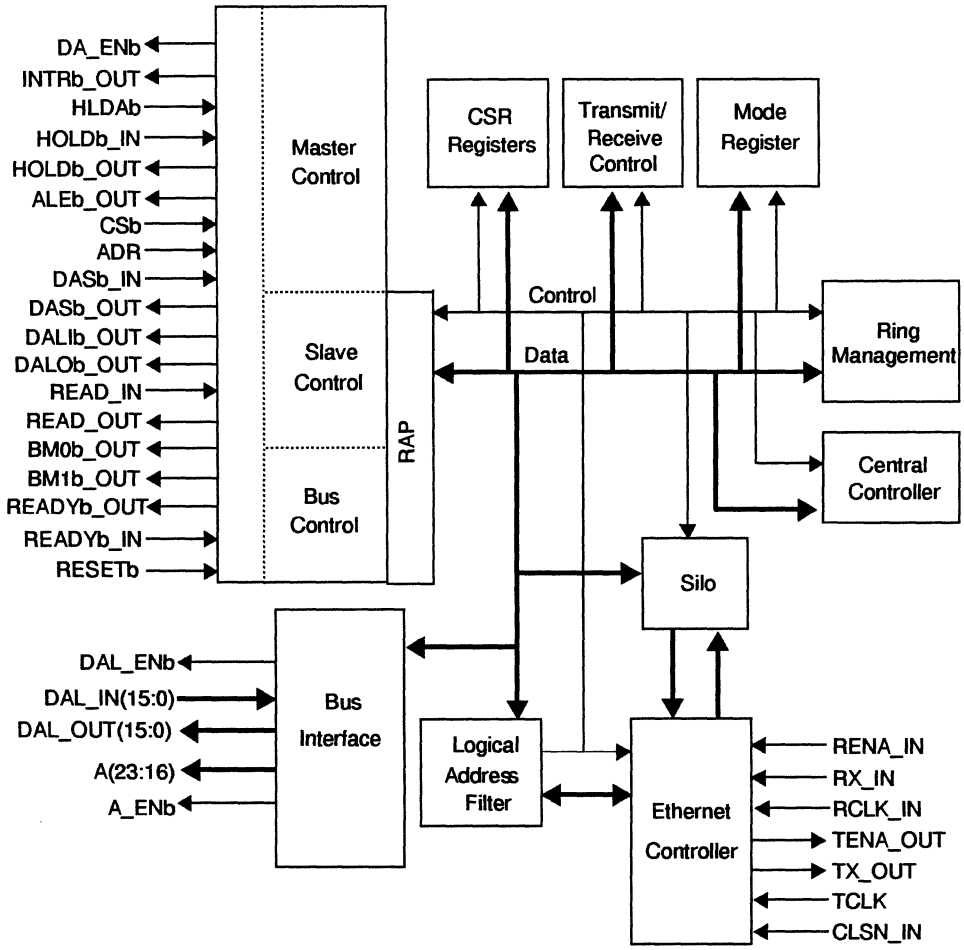


Figure 4-2 Macrocell Functional Block Diagram

LAN Interface

The NCR92C990 supports error reporting for diagnostics, addressing, collision, babbling transmitter (jabbering), packet framing, overflow and underflow.

Diagnostic Modes

Internal and external loopback modes are supported by the NCR92C990; they are configured through the internal registers.

Addressing Modes

Physical, logical and promiscuous modes are supported.

Physical

Packets can be received that have the full 48-bit destination address matching the physical address that is programmed into the NCR92C990 during initialization.

Logical

Packets can be received only if the destination address matches one of 64 logical addresses programmed in the logical address filter at initialization.

Promiscuous

A promiscuous mode allows reception of all packets. All incoming, error-free packets are accepted and stored in the buffer memory regardless of destination address or length.

Collision Detection

The NCR92C990 supports collision detection and recovery for both transmit and receive functions.

Transmitting

Upon a collision during transmit, the NCR92C990 will jam with 1 bit for 32 bit-times and then back off for a multiple number of slot-times. A slot-time is equal to 512 bit-times @ 10 MHz/100 ns periods. The delay for the next transmission is chosen from a uniformly distributed random integer in the range of 1 to 2^k where k=1,2,3,...n. Collisions during preamble, destination or source address fields, data fields, and CRC are handled in the same manner.

Receiving

Upon a collision during receive, the packet reception terminates immediately. A collision that occurs before the 64-byte interval results in a runt packet. A collision that occurs after the 64-byte interval causes a truncated packet that is transferred to memory. A CRC error is reported in this case.

Descriptor Ring Management

Buffer management for the NCR92C990 is handled by a recurrent list of assignments in memory called descriptor rings. There are separate descriptor rings for both transmit and receive. The NCR92C990 searches the descriptor rings to determine the next empty buffer. After an empty buffer is filled, the OWN bit is set in that descriptor ring. When a descriptor ring has its OWN bit set, the NCR92C990 starts a DMA transfer using the descriptor ring buffer pointer which points to the data memory buffer.

Internal Registers

Initialization Block

The NCR92C990 reads a data structure in memory to initialize pointers to memory transmit and receive buffers. This sets the mode of operation, and reads the logical address filter programming as well as the controllers physical address on the network. The initialization data structure contains the following:

- Mode register
- Physical address
- Logical address filter
- Receive ring address pointer and length
- Transmit ring address pointer and length

Mode Register

15	6	5	4	3	2	1	0
PRO	ILB	DRY	COL	DTC	LBK	DTX	DRX

Bits	Name	Description
0	DRX	Disable Receiver. When this bit is set, the NCR92C990 rejects all incoming packets because the receive descriptor ring is not accessed. DRX = 1 will clear the RON bit in Control/Status Register 0 when initialization is done.
1	DTX	Disable Transmitter. When this bit is set, the NCR92C990 does not access the transmit descriptor ring so no transmissions are attempted. DTX = 1 will clear the TON bit in Control/Status Register 0 when initialization is done.
2	LBK	Loopback. Loopback allows the NCR92C990 to transmit a packet addressed to itself which can be used to test the LAN interface at various levels. The packet size is limited to 8-32 bytes with 4 bytes allowed for CRC when DTC = 0. Runt packet detection is disabled. TBK = 1 allows concurrent transmit and receive for a packet length constrained to fit in the SILO. Transmission will begin when the entire packet is in the SILO. The entire received packet will be written to memory only after reception is complete. Transmit data chaining is not allowed in LBK mode. Receive data chaining is allowed if the receive packet length does not exceed 32 bytes.
3	DTC	Disable Transmit CRC. When DTC = 0, the transmitter generates and appends CRC to the transmitted packet. When DTC = 1, no CRC is appended to the packet. During LBK mode, DTC = 1 will cause CRC to be appended to the transmitted packet. Receive CRC will be checked by the NCR92C990 and written into memory. DTC = 1 disables CRC append during transmit. The host must append the CRC to the transmit packet in this case. Receive CRC will be checked.
4	COL	Force Collision. This bit allows testing of the collision logic. The NCR92C990 must be in internal loopback mode for COL to be valid. If COL = 1, a collision will be forced during the next transmit and can result in 16 transmission retries. Transmit Descriptor 3 RTY will be set = 1 in this case.
5	DRY	Disable Retry. When DRY = 1, the NCR92C990 attempts to transmit a packet only once. If there is a collision on this first attempt, a retry error is reported. In Transmit Descriptor 3, RTY = 1.

Bits	Name	Description
6	ILB	Internal Loopback. This is set with the LBK bit to determine where the loopback function is done. The NCR92C990 does not receive any external packets when it is in internal loopback mode. The packet size is limited to 8-32 bytes. Extend packet reception is disabled. Muticast addressing in External Loopback (LBK) is valid only when DTC = 1. Received packets will be accepted from the network. ILB is valid only when LBK = 1.

LBK	ILB	Loopback Mode
0	X	Normal Mode
1	0	External Loopback
1	1	Internal Loopback

Bits	Name	Description
14-7	res	Reserved.
15	PRO	Promiscuous mode. When PRO = 1, all incoming packets are accepted.

Physical Address (PADR)

The physical address is the unique 48-bit physical address assigned to the NCR92C990. PADR (0) must = 0.

Logical Address Filter (LADR)

The logical address filter is a 64-bit mask used by the NCR92C990 to accept logical addresses. It is composed of four 16-bit registers. The logical address filter is a 64-bit mask used to qualify incoming packets. After the 48 bits of the destination address have gone through the LADR CRC circuit, the high order 6 bits of the 32-bit LADR CRC value are set to select 1 of 64 bit positions in the Logical Address filter. If a bit position in the logical address filter = 1, the packet will be accepted. Broadcast addresses (all ones) pass transparently through the logical address filter. If the logical address filter (LADR) = 0, all packets are rejected except Broadcast packets.

Receive Ring Pointer

31-29	28-24	23-3	2	1	0
RRL	res	RRA	zero	zero	zero

Bits	Name	Description
2-0	zero	These bits must be zero. Receive Ring Pointers are quad word boundary aligned.
23-3	RRA	Receive Ring Address. These bits are the lowest address bits of the receive ring.
28-24	res	These bits are reserved for internal use.
31-29	RRL	Receive Ring Length. This is the number of entries in the receive ring expressed as a power of two.

RRL	Number of Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

Transmit Ring Pointer

31-29	28-24	23-3	2	1	0
TRL	res	TRA	zero	zero	zero

Bits	Name	Description
2-0	zero	These bits must be zero. Transmit Rings are aligned with quad word boundaries.
23-3	TRA	Transmit Ring Address. These bits are the lowest address bits of the transmit ring.
28-24	res	These bits are reserved for internal use.
31-29	TRL	Transmit Ring Length. This is the number of entries in the transmit ring expressed as a power of two.

TRL	Number of Entries
0	1
1	2
2	4

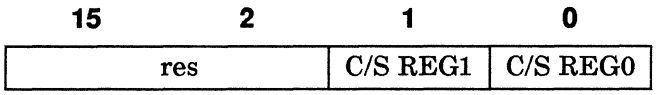
TRL	Number of Entries
3	8
4	16
5	32
6	64
7	128

Control/Status Registers

The Control/Status registers are accessed through the register address port and the register data port. The status register being accessed is determined by the value of the ADR input. When ADR = 0, the register data port is selected and when ADR = 1, the register address port is selected.

The address of the Control/Status register to be accessed is written to the register address port. All subsequent reads and writes to the data port have the same data read/written to the selected Control/Status register.

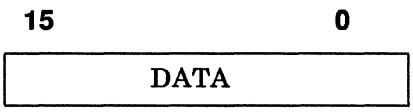
Register Address Port



Bits	Name	Description
1-0	C/S	Control/Status Register Select. The Register Address Port is READ/WRITE and selects the Control/Status Register to be accessed through the Register Data Port. RESETb clears the Register Address Port.
15-2	res	Reserved. Read as 0.

Bit 1	Bit 0	Register
0	0	Control/Status Register 0
0	1	Control/Status Register 1
1	0	Control/Status Register 2
1	1	Control/Status Register 3

Register Data Port



Bits	Name	Description
15-0	DATA	Data written to the Register Data Port is actually written to the Control/Status Register selected by the Register Address Port. The Control/Status Registers 1, 2 and 3 are only accessible when the stop bit is set in Control /Status Register 0.

Control Status Register 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERR	BAB	CE	MISS	ME	RINT	TINT	IFIN	INT	IEN	RON	TON	TD	STP	STR	INIT

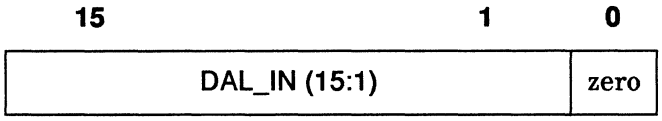
Bits	Name	Description
0	INIT	Initialize. This bit causes the NCR92C990 to start the initialization procedure and access the initialization block. Set the STP bit prior to setting INIT. INIT clears the STP bit. INIT is Read/Write with a 1 only. Writing zero bits into INIT has no effect. INIT is cleared by chip reset on setting STP.
1	STR	Start. Start enables the NCR92C990 to send and receive packets, access memory, and perform management tasks. Set the STP bit prior to setting STR. INIT clears the STR bit. Wait until INFIN = 1 to set STR. STR is Read/Write with a 1 only. STR is cleared by reset.
2	STP	Stop. STP disables the NCR92C990 from all external activity when set and clears the internal logic. STP is the same as RESETb. STP stays set until STR or INIT is set. STP is Read/Write with a 1 only. STP is set by RESETb. STP is cleared by INIT or STR.
3	TD	Transmit Demand. This bit causes the NCR92C990 to access the Transmit Ring Pointer register without waiting for a time interval. TD is write with 1 only. TD is cleared by RESETb or by STP = 1.
4	TON	Transmitter On. This bit shows that the transmitter is enabled. TON is cleared by RESETb or by setting STP. TON is set = 1 when STR = 1 and DTX = 0 in the Mode Register, for example, initialization block has been read. TON is Read only. TON is cleared by RESETb or STP = 1.
5	RON	Receiver On. This bit shows that the receiver is enabled. RON is cleared by RESETb or by setting STP. RON is set = 1 when STR = 1 and DRX = 0 in the Mode Register, for example, initialization block has been read. RON is cleared by Minor Error (ME), or, if IFIN = 1 from setting INIT=1 and DRX = 1 in the Mode Register. RON is Read only and is cleared by RESETb or STP = 1.

Bits	Name	Description
6	IEN	Interrupt Enable. IEN allows the INT bit to go low. IEN cannot be set while STP is set, but it can be set in parallel or afterwards. If IEN = 1 and INT = 1, the INTRb_OUT pin will = 0. Otherwise INTRb_OUT = 1. IEN is Read/Write and is cleared by RESETb or STP = 1.
7	INT	Interrupt. INT is set by any one of the following bits being set: BAB, MISS, ME, RINT, TINT, and IFIN. INT is Read only and is cleared by RESETb or by clearing the interrupt cause.
8	IFIN	Initialization Finished. This bit indicates that the NCR92C990 has finished the initialization procedure started by the INIT bit. All new parameters in the initialization block are now stored. When IFIN = 1, an interrupt is generated if IEN = 1. IFIN is Read/Clear and is cleared by writing a 1 to IFIN. RESETb and STP = 1 clear IFIN.
9	TINT	Transmitter Interrupt. This bit is set when the NCR92C990 updates an entry in the transmit ring description for the last buffer sent or when transmit is stopped due to an error. When TINT = 1, an interrupt is generated if IEN = 1.
10	RINT	Receiver Interrupt. This bit is set when the NCR92C990 updates an entry in the receive ring description for the last buffer received or if a receive is stopped due to an error. When RINT = 1, an interrupt is generated if IEN = 1.
11	ME	Memory Error. This bit is set when the NCR92C990 is in bus master mode and has not received a signal from READYb_IN after asserting the address on DAL_OUT(15:0). READYb_IN must be received within 102.4 microseconds after asserting DAL_OUT(15:0). The Receiver and Transmitter are turned off if ME = 1, and if IEN = 1 an interrupt is generated. ME is Read/Clear by writing a 1 to ME. RESETb or STP = 1 clear ME.
12	MISS	Missed Packet. This bit is set when the receiver loses a packet of data because it does not own any receive buffer. An interrupt is generated if IEN = 1. If MISS = 1 and IEN = 1, an interrupt is generated. MISS is Read/Clear by writing a 1 to MISS. RESETb on STP = 1 clears MISS.
13	SQE	Signal Quality Error test. This bit is set when the SQE test that follows every packet transmission fails. This indicates that the collision detection circuitry in the transceiver may be faulty. Packets still transmit, but should be retransmitted because they may have collided without the collision being detected. SQE = 1 means that on transmission the CLSN pin started to activate within 2.2 microseconds after the fall of RENA.

Bits	Name	Description
14	BAB	Babble. This bit shows a transmitter time-out error. It indicates that the transmitter has been on the channel longer than the time required to send the maximum length packet. BAB will be set after 1519 bytes have been transmitted. If IEN = 1, an interrupt will be generated. BAB is Read/Clear by writing a 1 to BAB. BAB is cleared by RESETb or STP = 1.
15	ERR	Error. ERR is set by any one of the following bits being set: BAB, SQE, MISS, and ME. ERR stays set while any of these errors are true. ERR is cleared by RESETb, by setting STP, or by clearing the individual error. ERR is Read only.

Control/Status Register 1

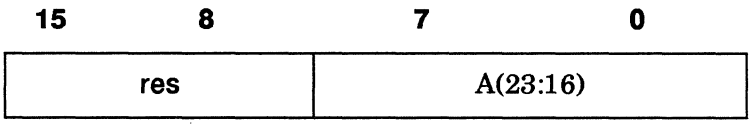
Accessible only when STP = 1 and RAP = 1. Contents are best after STP =1.



Bits	Name	Description
0	zero	This bit must be set to zero.
15-1	DAL	These 15 bits, along with bit 0 , are the low order 16 bits of the initialization block address.

Control/Status Register 2

Accessible only when STP = 1 and RAP = 2. Contents are best after STP = 1.



Bits	Name	Description
7-0	A(23:16)	This represents the high order 8 bits of the initialization block address.
15-8	res	Reserved.

Control/Status Register 3

Accessible only when STP = 1 and RAP = 3. Contents are lost after STP = 1. CSR3 is cleared by RESETb or STP = 1.

15	14 12	11 9	8 3	2	1	0
IPGS	RX-TX IPG	TX-TX IPG	res	SWAP	ACON	BCON

Bits	Name	Description
0	BCON	Byte Control. BCON redefines HOLDb_OUT, BM0b_OUT, and BM1b_OUT as shown below. BCON is Read/Write. RESETb or STP = 1 clears BCON.

BCON	Pin 17	Pin 16	Pin 15
0	HOLDb_OUT	BM1b_OUT	BM0b_OUT
1	BURSRQb_OUT	BUSAK0b_OUT	BYTE

Bits	Name	Description
1	ACON	ALE State Control. This bit defines the assertive state of ALEb_OUT when the NCR92C990 is in bus master mode. ACON is Read/Write. RESETb or STP= 1 clears ACON.

ACON	ALEb_OUT
0	Asserted High
1	Asserted Low

Bits	Name	Description
2	SWAP	Byte Swap. This bit allows the NCR92C990 to operate in systems that require bit 15:8 to be pointed to an even address and bits 7:0 to be pointed to an odd address. SWAP = 1 allows the NCR92C990 to swap the HI/LO bytes on DMA transfers. Ring entries, INIT Block data is not swapped. SWAP is Read/Write and cleared by RESETb or STP = 1.
8-3	res	Reserved. Read as 0.
11-9	TX-TX IPG	Programmable TX IPG. Transmit IPG range with Control/Status Register 3 (bit 15) = 1.
14-12	RX-TX IPG	Programmable RX IPG.

Bits	Name	Description
15	IPGS	IPG Range Select with CSR3(15) = 0 the default settings for TX-TX IPG and RX-TX IPG are selected. This is the normal power-up mode for the macrocell. If CSR3(15) = 1 then the contents of CSR3(14-7) select the programmable IPG values according to the tables shown. CSR3(15-3) are Write/Read but CSR(15-3) are read as 0.

TX IPG Range

With CSR3(15) = 1

Bit 11	Bit 10	Bit 9	TX IPG Range
0	0	0	9.6 - 10.4µs
0	0	1	11.2 - 12.0µs
0	1	0	12.8 - 13.6µs
0	1	1	14.4 - 15.2µs
1	0	0	16.0 - 16.8µs
1	0	1	17.6 - 18.4µs
1	1	0	19.2 - 20.0µs
1	1	1	20.8 - 21.6µs

With CSR3(15) = 0, the TX -TX IPG Range = 9.6 - 10.2µs.

TX-TX is defined as any transmission initiated by this node that follows either another transmission from this node, or any number of receive packets not addressed to this node, which followed a transmission from this node. The previous transmission need not have been successful.

Example:
TX - collided and backed off or was successful
RX - no address match
RX - no address match TX
TX - follows

NCR92C990 behavior:
For TX-TX, the NCR92C990 begins the IPG count as soon as RENA_IN falls after the previous transmission or receive packet that was not received. Once the IPG count begins, it will ignore any reassertions of RENA_IN. After the IPG expires, if there is at least one byte of TX data in the FIFO, the NCR92C990 will begin to transmit. If the FIFO is empty, the NCR92C990 will continue to defer until the FIFO has at least one byte and then immediately begins to transmit. If between the time the IPG expires and the FIFO obtains one byte RENA_IN is asserted, the NCR92C990 will reset the IPG counter and wait until RENA_IN again falls. Once RENA_IN falls the IPG counter will again begin to count and the cycle is repeated.

RX IPG Range

Receive followed by transmit IPG range. With Control/Status register 3(15) = 1			
Bit 14	**Bit 13**	**Bit 12**	**RX IPG Range**
0	0	0	9.6 - 10.4µs
0	0	1	11.2 - 12.0µs
0	1	0	12.8 - 13.6µs
0	1	1	14.4 - 15.2µs
1	0	0	16.0 - 16.8µs
1	0	1	17.6 - 18.4µs
1	1	0	19.2 - 20.0µs
1	1	1	20.8 - 21.6µs

With CRS3(15) = 0, the RX-TX IPG Range = 19.2 - 20.0µs.

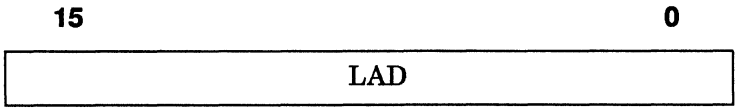
RX-RX is defined as any transmission initiated by this node when the last transaction processed by this node was any reception with a matching address whether it was a runt, missed packet, or a successful reception.

Example:
RX - addressed to this node (missed, runt or received)
RX - not address match TX
TX - follows

NCR92C990 behavior:
For RX-TX the IPG count begins as soon as RENA_IN falls for the previous receive (whether it was addressed to this node or not). If RENA_IN becomes reasserted during the first 3.2µs of the IPG (minus synchronization delay), the IPG counter will be reset and wait until RENA_IN falls again to resume counting. Once the IPG count expires and the FIFO contains at least one byte, transmission will begin. If RENA_IN becomes reasserted after the first 3.2µs of the IPG, the IPG counter will not be reset. Once the IPG counter expires and the FIFO contains at least one byte of TX data, transmission will begin. The FIFO cannot obtain 1 byte of data until RENA_IN falls or the packet is not addressed to this node and the NCR92C990 polls and begins to burst data. If between the time that the IPG expires and the FIFO contains at least 1 byte of data, RENA_IN is asserted, the IPG counter is reset, and waits until RENA_IN falls again. Once RENA_IN falls the IPG counter will again begin to count and the cycle is repeated.

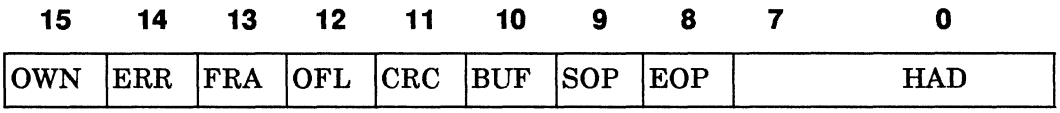
Buffer Management

Receive Descriptor 0



Bits	Name	Description
15-0	LAD	Low-order Address. These are the lowest 16 address bits of the buffer to which this descriptor points. They are written by the host.

Receive Descriptor 1



Bits	Name	Description
7-0	HAD	High-order Address bits. These are the upper 8 address bits of the buffer to which this descriptor points. They are written by the host. The NCR92C990 does not change these bits.
8	EOP	End of Packet. This is the last buffer used by the NCR92C990 for this packet. IF SOP = 1 and EOP = 1, the packet fits into one simple buffer.
9	SOP	Start of Packet. This is the first buffer used by the NCR92C990 for this packet. If EOP and SOP are both set, the packet fits into one buffer.
10	BUF	Buffer Error. This bit is set when the NCR92C990 does not own the next buffer while receiving a packet. If a buffer error occurs, an OFL error may also occur but it is not reported unless both BUF and OFL occur at the same time.
11	CRC	Cyclic Redundancy Check. This occurs when the receiver detects a CRC error on the incoming packet. CRC is valid only when EOP is set and OFL is not set.
12	OFL	Receiver Overflow. This error indicates that the receiver has lost all or part of the incoming packet data because the temporary buffer overflowed. OFL is valid only when EOP is not set.

Bits	Name	Description
13	FRA	Frame Error. This bit shows that the incoming packet contained a noninteger multiple of eight bits. FRA is not valid in internal loopback mode. FRA is valid only when EOP is set and OFL is not set. FRA should be ignored if no CRC error is reported.
14	ERR	Error. This bit is set when at least one of the following bits is set: FRA, OFL, CRC, or BUF.
15	OWN	This bit shows ownership of the decriptor entry. If OWN = 0, the host owns this entry. If OWN = 1, the NCR92C990 owns this entry. The host sets this bit after emptying the buffer to which this descriptor points. The NCR92C990 clears this bit after filling the buffer. If the host for the NCR92C990 has set this bit, for example, a buffer has been given over, the descriptor entries must not be changed.

Receive Descriptor 2

15	14	13	12	11	0
1	1	1	1	BBL	

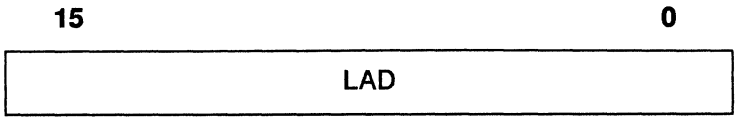
Bits	Name	Description
11-0	BBL	Buffer Byte Length. This represents the length of the buffer to which this descriptor points. This number is expressed as a two's complement. BBL is written by the host. The NCR92C990 does not change BBL in any way.
15-12	1	These bits must be ones. This field is written by the host. The NCR92C990 does not change these bits.

Receive Descriptor 3

15	12	11	0
res		MBL	

Bits	Name	Description
11-0	MBL	Message Byte Length. This represents the length of the received message in bytes. MBL is valid only when ERR is clear and EOP is set. MBL is written by the NCR92C990 and cleared by the host.
15-12	res	Reserved. These bits are reserved and read as zeros.

Transmit Descriptor 0



Bits	Name	Description
15-0	LAD	Low-order Address. These are the lowest 16 address bits of the buffer to which this descriptor points. They are written by the host. The NCR92C990 does not change these bits.

Transmit Descriptor 1

15	14	13	12	11	10	9	8	7 ... 0
OWN	ERR	res	MOR	ONE	DEF	SOP	EOP	HAD

Bits	Name	Description
7-0	HAD	High-order Address bits. These are the 8 address bits of the buffer to which this descriptor points. They are written by the host. The NCR92C990 does not change these bits.
8	EOP	End of Packet. This is the last buffer used by the NCR92C990 for this packet. If EOP and SOP are both set, the packet fits into a single buffer. EOP is set by the host. The NCR92C990 does not alter EOP.
9	SOP	Start of Packet. This is the first buffer used by the NCR92C990 for this packet. If EOP and SOP are both set, the packet fits in one buffer. SOP must = 1 in the first buffer of a packet or it will be skipped over during a pull until the OWN bit and SOP bits are set.
10	DEF	Deferred. This means that the NCR92C990 had to defer while trying to transmit a packet. This occurs if the channel is busy when the NCR92C990 is ready to transmit.
11	ONE	One. This means that exactly one retry was necessary to transmit a packet. ONE is not valid if LCOL is set.
12	MOR	More. This means that more than one retry was necessary to transmit a packet.
13	res	Reserved. The NCR92C990 writes a zero to this bit.
14	ERR	Error. This bit is set when at least one of the following bits is set: RTY, CLOS, LCOL, or UFL.

Bits	Name	Description
15	OWN	This bit shows ownership of the descriptor entry. If OWN = 0, the host owns this entry. If OWN = 1, the NCR92C990 owns this entry. The host sets this bit after emptying the buffer to which this descriptor points. The NCR92C990 clears this bit after filling the buffer. Once either the host or NCR92C990 has given over a buffer, it must not alter the descriptor entry.

Transmit Descriptor 2

15	14	13	12	11 ... 0
1	1	1	1	BBL

Bits	Name	Description
11-0	BBL	Buffer Byte Length. This represents the length of the buffer to which this descriptor points. This number is expressed as a two's complement. BBL is written by the host. The NCR92C990 does not change BBL in any way.
15-12	1	These bits must be ones. This field is written by the host. The NCR92C990 does not change these bits.

Transmit Descriptor 3

15	14	13	12	11	10	9 ... 0
BUF	UFL	res	LCOL	CLOS	RTY	TDR

Bits	Name	Description
9-0	TDR	Time Domain Reflectometry. This bit shows the state of a counter internal to the NCR92C990. This counter counts from the start of a transmission to a collision if one occurs. This value can be used to determine the approximate distance to a fault in a cable. TDR is valid and written by the NCR92C990 only if RTY is set.
10	RTY	Retry. This error indicates that the transmitter has failed because of collisions in 16 attempts to successfully transmit a packet. If the Disable Retry (DRY) bit in the mode register is set, RTY will set after only one failed transmission attempt.

Bits	Name	Description
11	CLOS	Carrier Loss. This bit is set when the carrier input to the NCR92C990 goes low during a transmission. The NCR92C990 does not try again when it loses the carrier; it transmits the entire packet until complete. CLOS is not valid during internal loopback.
12	LCOL	Late Collision. This bit means that a collision occurred after the channel slot-time (64 bytes) elapsed. The NCR92C990 does not retry on late collisions.
13	res	Reserved. The NCR92C990 sets this bit to 0.
14	UFL	Underflow. This error indicates that the transmitter has shortened a message because the data was late coming from memory. UFL indicates that the temporary buffer emptied before the end of the packet. The transmitter is turned off and TON set = 0.
15	BUF	Buffer Error. This bit is set during transmission when the NCR92C990 cannot find the EOP = 1 in the current buffer and does not own the next buffer. BUF is set by the NCR92C990 and cleared by the host. If a buffer error occurs, an underflow error also occurs. The transmitter is turned off and TON set = 0.

NCR92C990 Changes and Differences to the Am7990

In engineering the NCR92C990, NCR noticed discrepancies in the way the Am7990 operated and the way the documentation read. A summary follows of these items that are most significant to hardware and software designers.

Documented

1. The Am7990 documentation states that the initialization block must begin on an even word address. Odd-byte initialization addresses cause the Am7990 to operate in an undefined manner.

   The NCR92C990 accepts odd-byte initialization block addressing in addition to the standard even-byte addressing. In odd-byte addressing, the address increments by one after the first DMA transfer from the odd-byte address to get on an even boundary. Then the address counter increments by two, as with even-byte addressing.

2. The Am7990 documentation states that no DMA bursts occur between polling and loading of DMA cycles. Under certain circumstances, the Am7990 actually performs DMA bursts between the polling and loading DMA cycles (ring accesses).

   The NCR92C990 is designed to operate as the Am7990 does, not as the documentation indicates.

Undocumented

1. The Am7990 asserts the INTRb output 25.5 cycles after the rise of

HOLDb on the last initialization DMA read cycle.

The NCR92C990 asserts INTRb after completing 1.5 cycles after the rise of the HOLDb output on the last initialization DMA read cycle.

2. Because of timing differences in the response time to a receive-based collision and the transfer of received data to the FIFO, responses may differ on the NCR92C990 if the collision signal is asserted during the header, between the destination and source addresses of a receive packet, or during the 64th data byte. In the first case, the lack of a sync bit causes the abort. In the second case, an address mismatch causes the abort. In the third case, a runt packet causes a CRC and framing error. It is possible for one of these aborts or errors in the Am7990 to appear as one of the others in the NCR92C990.

3. When the first DMA transfer is a data word, the Am7990 starts transmission eight cycles after the rising edge of the DASb pulse. If the first DMA transfer is only a byte, the Am7990 starts the transmission 12 cycles after the rising edge of the DASb pulse.

   In both the byte and word cases, the NCR92C990 starts transmission eight cycles after the rising edge of DASb.

4. During loopback, the rising edge of the DASb pulse latches in the last byte/word of a transmission. The Am7990 begins transmission of the next packet preamble on the rising edge of DASb.

   The NCR92C990 begins transmission within five clock cycles after the rising edge of DASb.

5. During external loopback, the Am7990 SILO can lose some data if there is heavy traffic on the media.

   The NCR92C990 does not lose data, even during heavy traffic on the media.

6. The Am7990 generates a memory error if READYb has not been asserted within 254 cycles after DASb falls and both the transmitter and receiver are turned off. If READYb falls between 251 and 253 cycles after DASb falls, incorrect data is loaded into Control/Status Register 0, the memory error (ME) bit is set, but neither the transmitter nor receiver are turned off.

   The NCR92C990 matches this behavior.

7. The Am7990 requires 11 clock cycles after the rising edge of HOLDb to set the missed packet (MP) bit in Control/Status Register 0.

   The NCR92C990 sets the MP bit within two clock cycles of the rising edge of HOLDb.

8. If the SILO overflows during a receive operation, the Am7990 updates the ring with the number of bytes actually transferred to memory at that point.

The NCR92C990 updates the ring with the number of bytes transferred to the memory, plus any that may be left in the FIFO when the overflow occurred. The data in the FIFO is not actually transferred to memory. Because the packet is incomplete, the host should ignore all the data in the buffer, and the actual amount of data in the buffer should not be considered.

9. The Am7990 begins Time Domain Reflectometry (TDR) counts from the assertion of RENA.

   The NCR92C990 begins TDR counts 2-3 cycles after the assertion of TENA, as specified in the IEEE 802.3 specification.

   TDR counts are only valid after a retry. During TX updates when a retry does not occur, the Am7990 writes invalid TDR values to the ring.

   The NCR92C990 also writes invalid TDR data to the ring, but not necessarily the same values.

10. The Am7990 recognizes a defer condition when it is ready to assert TENA at the start of transmission, but RENA is already asserted.

    The NCR92C990 recognizes a defer condition when it is ready to assert TENA at the start of a transmission and either RENA is already asserted or the interpacket gap time has not elapsed since RENA went inactive.

11. If the SILO on the Am7990 contains 43 or more bytes of data from a previous reception and another packet is being sent by another node on the network, an overflow condition can occur even if the second packet is not addressed to this node. For example: 43 bytes of data remain in the SILO from a previous packet reception. The Am7990 checks the incoming packet for a valid destination address and, in the process, tries to transfer the six address bytes to the SILO. It recognizes that the address is not its own after the sixth byte, but before that happens, an overflow of the 48-byte SILO occurs (43 + 6 = 49). The SILO pointer is reset to the 44th byte in the FIFO, but the packet residing in the SILO is labeled with an overflow error.

    The NCR92C990 temporarily stores the incoming destination address in a separate FIFO which is either erased if the address is not matched, or burst into the main 48-byte FIFO if the address is matched. No overflow is reported by the NCR92C990 in the previous example.

12. Defer is reported if the first attempt to transmit a packet is delayed because RENA is asserted when the NCR92C990 is ready to transmit (the IPG has expired and the TX FIFO contains at least one byte of data); or, for an RX-TX transmission, defer is reported if the NCR92C990 is ready to transmit and the IPG time has not expired.

13. Interpacket gap: Either TX-TX or RX-TX.
    In general, three conditions are required in order to begin a transmission:

- The backoff count is 0 (only pertains to retries)
- The IPG has expired.
- The TX FIFO contains at least 1 byte of data.

The Am7990 TX FIFO is reset (and thus empty) whenever RENA is asserted except when:

- TENA is also asserted, or
- RENA is still asserted from a previous transmission, or
- An RX packet is still on the network but the RX circuitry has determined that it is not addressed to this node.

Therefore, the NCR92C990 will defer to any incoming packet, regardless of the state of the IPG counters except for a packet which causes RENA to be asserted just before TENA is to be asserted for transmission.

The NCR92C990 IPG counter will be reset if the RENA pulse is shorter than the remainder of the IPG as programmed. The IPG features control IPG timings which are different for RX-TX, TX-TX and the default IPG partition.

If the NCR92C990 is ready to transmit and is waiting for the IPG to expire, it will defer to the incoming packet thus avoiding a collision.

TX-TX

The Am7990 data sheet refers to the condition that if RENA is asserted within the first 4.1 µs of the IPG after a TX that the Am7990 will defer to the packet and receive it if the sync bit is correctly recognized after the first 4.1 µs of the IPG. The NCR92C90 will begin to look for the sync immediately.

RX-TX

The Am7990 data sheet refers to the condition that if RENA is asserted within the first 4.1 µs of the IPG after a RX that the Am7990 will defer to the packet and not receive it. The NCR92C990 will defer to the packet and will receive it.

Electrical Specifications

Absolute Maximums

Symbol	Parameter	Minimum	Maximum	Units
T_A	Ambient Temperature	0	70	°C
T_S	Storage Temperature	-55	125	°C
V_{DD}	Supply Voltage	-0.5	7.0	V
V_{IN}	Input Voltage	-0.5	V_{DD} + 0.5	V
V_{OUT}	Output Voltage	-0.5	V_{DD} + 0.5	V
T_L	Lead Temperature (Soldering 10 seconds maximum)		250	°C

DC Characteristics

(V_{DD} = 4.75 to 5.25 V, V_{SS} = 0 V, T_A = 0° C to 70° C)

Symbol	Parameter	Minimum	Maximum	Units
V_{IL}	Input Voltage, Low	V_{SS}- 0.5	0.8	Volts
V_{IH}	Input Voltage, High	2.0	V_{DD}	Volts
V_{OL}	Output Voltage, Low (V_{DD} = 4.5 V)		0.4	Volts
V_{OH}	Output Voltage, High (V_{DD} = 4.5 V)	2.4		Volts
C_{IN}	Input Capacitance		10	pF
I_{TL}	Input Leakage Current (V_{DD} = 5.5 V)		±10	µA
I_{OL}	Output Leakage Current (V_{DD} = 5.5 V)		±10	µA
I_{DD}	Power Supply Current		50	mA
I_{OL}	Output Current, Low	4		mA
I_{OH}	Output Current, High	-4		mA

AC Characteristics

The following timing parameters apply strictly to the VS700H macrocell. These timings are derived from pre-layout estimated capacitances. Post-layout timing variations may occur.

(V_{DD} = 4.75 to 5.25 V, V_{SS} = 0 V, T_A = 0° C to 70° C) External load 1 pF

Num.	Description	Fig.	Min.	Type	Max.	Units
1	TCLK period	1	99		101	ns
2	TCLK high pulse duration	1	45			ns
3	TCLK low pulse duration	1	45			ns
4	TCLK rise time	1			8	ns
5	TCLK fall time	1			8	ns
6	TENA_OUT propagation delay after rising edge of TCLK	1			25	ns
7	TENA_OUT hold time after TCLK rising	1	7			ns
8	TX_OUT propagation delay after TCLK rising	1			32	ns
9	TX_OUT hold time after TCLK rising	1	7			ns
10	RCLK_IN period	1	85		118	ns
11	RCLK_IN high pulse duration	1	38			ns
12	RCLK_IN low pulse duration	1	38			ns
13	RCLK_IN rise time	1			8	ns
14	RCLK_IN fall time	1			8	ns
15	RX_IN data rise time	1			8	ns
16	RX_IN data fall time	1			8	ns

(V_{DD} = 4.75 to 5.25 V, V_{SS} = 0 V, T_A = 0° C to 70° C) External load 1 pF

Num.	Description	Fig.	Min.	Type	Max.	Units
17	RX_IN data hold time from RCLK rising	1	5			ns
18	RX_IN data setup time to RCLK rising	1	40			ns
19	RENA_IN low duration	1	200			ns
20	CLSN_IN high duration	2	100			ns
21	Bus master driver disable after rising edge of HOLDb_OUT	3, 4			2	ns
22	Bus Master driver enable after falling edge of HLDAb (Bus Master)	3, 4			202	ns
23	Delay from HOLDb_OUT to falling edge of HLDAb (Bus Master)	3, 4	0			ns
24	RESETb pulse width low		200			ns
25	Read/Write, Address/Data cycle time		600			ns
26	A(23:16) setup time to the falling edge of ALEb_OUT	3, 4	144			ns
27	A(23:16) hold time after the rising edge of DASb_OUT	3, 4	100			ns
28	DAL_OUT(15:0) setup time to falling edge of ALEb_OUT	3, 4	144			ns
29	DAL_OUT(15:0) hold time after falling edge of ALEb_OUT	3, 4	53			ns
30	Data setup time to the rising edge of DASb_OUT (Bus Master Read)	3	2			ns
31	Data hold time from the rising edge of DASb_OUT (Bus Master Read)	3	8			ns
32	Data setup time to falling edge of DASb_OUT (Bus Master Write)	4	46			ns
33	Data setup time to the rising edge of DASb_OUT (Bus Master Write)	4	295			ns
34	Data hold time after the rising edge of DASb_OUT (Bus Master Write)	4	102			ns
35	Data driver delay after the falling edge of DASb_IN (Bus Slave Read) (RAP)	5		200		ns
36	Data driver delay after the falling edge of DASb_IN (Bus Slave Read) (CSR0, 1,2,3)	5		200		ns
37	Data hold time after the rising edge of DASb_IN (Bus Slave Read)	5	2		6	ns
38	Data hold time after the rising edge of DASb_IN (Bus Slave Write)	6	0			ns
39	Data setup time to the falling edge of DASb_IN (Bus Slave Write)	6	0			ns

(V_{DD} = 4.75 to 5.25 V, V_{SS} = 0 V, T_A = 0° C to 70° C) External load 1 pF

Num.	Description	Fig.	Min.	Type	Max.	Units
40	ALEb_OUT high pulse width	3, 4	144			ns
41	Delay from rising edge of DASb_OUT to rising edge of ALEb_OUT	3, 4	100			ns
42	DASb_OUT low pulse width	3, 4	252			ns
43	Delay from falling edge of ALEb_OUT to falling edge of DASb_OUT	3, 4	100			ns
44	Delay from rising edge of DALOb_OUT to the falling edge DASb_OUT (Bus Master Read)	3	50			ns
45	Delay from falling edge READYb_IN to rising edge of DASb_OUT (Bus Master)	3, 4	55		255	ns
46	Delay from rising edge of DALOb_OUT to falling edge of DALIb_OUT (Bus Master Read)	3	55			ns
47	DALIb_OUT setup time to rising edge of DASb_OUT (Bus Master)	3, 4	240			ns
48	DALIb_OUT hold time after the rising edge of DASb_OUT (Bus Master Read)	3	8			ns
49	Delay from rising edge of DALIb_OUT to falling edge of DALOb_OUT (Bus Master Read)	3	82			ns
50	DALOb_OUT setup time to the falling edge of ALEb_OUT (Bus Master Read)	3	150			ns
51	DALOb_OUT hold time after the falling edge of ALEb_OUT (Bus Master Read)	3	50			ns
52	Delay from rising edge of DASb_OUT to rising edge of DALOb_OUT (Bus Master Write)	4	50			ns
53	CSb hold time after the rising edge of DASb_IN (Bus Slave)	5, 6	0			ns
54	CSb setup time to the falling edge of DASb_IN (Bus Slave)	5, 6	0			ns
55	ADR hold time after the rising edge DASb_IN (Bus Slave)	5, 6	0			ns
56	ADR setup time before the falling edge of DASb_IN (Bus Slave)	5, 6	0			ns
57	Delay from ALEb_OUT to falling edge of READYb_IN to ensure a minimum bus cycle time of 600 ns (See note on page 34.)	3, 4			395	ns
58	Data setup time to the falling edge of READYb_OUT (Bus Slave Read)	5	100			ns

(V_{DD} = 4.75 to 5.25 V, V_{SS} = 0 V, T_A = 0° C to 70° C) External load 1 pF

Num.	Description	Fig.	Min.	Type	Max.	Units
59	READYb_IN hold time after the rising edge of DASb_OUT (Bus Master)	3, 4	8			ns
60	READYb_OUT driver on after the falling edge of DASb_IN (Bus Slave Write) (RAP)	5, 6		300		ns
61	READYb_OUT driven on after the falling edge of DASb_IN (Bus Slave Write) CSR0, 1, 2, 3	5, 6		300		ns
62	READYb_OUT hold time after the rising edge of DASb_IN (Bus Slave)	5, 6	0		6	ns
63	READ_IN hold time after the rising edge of DASb_IN (Bus Slave)	5, 6	0		6	ns
64	READ_IN setup time to the falling edge of DASb_IN (Bus Slave)	5, 6	0			ns
65	TCLK rising edge for low or high delay	3, 4			14	ns
66	TCLK rising to valid address	3, 4			115	ns
67	TCLK rising edge to control signals active	3, 4			115	ns
68	TCLK falling edge to ALEb_OUT low	3, 4			115	ns
69	TCLK falling edge to DASb_OUT falling edge	3, 4			115	ns
70	READYb_IN setup time to TCLK (See note on page 34.)	3, 4	0			ns
71	TCLK rising edge to DASb_OUT high	3, 4			12	ns
72	HLDAb setup to TCLK	3, 4	0			ns
73	RENA_IN hold time after the rising edge of RCLK_IN	1	0			ns
74	RENA_IN defer before TENA_OUT	1	400			ns
75	RENA_IN extended after RCLK_IN last falling	1			275	ns
76	Delay from DASb_OUT rising to HOLDb_OUT rising	3, 4	100			ns
77	READYb_OUT falling to DASb_IN rising (Slave Write/Read)	5, 6	0			ns
78	ALEb_OUT falling to next ALEb_OUT falling (DMA Burst) (Not shown in diagrams)	--		600		ns
79	DASb_IN falling to DAL_ENb falling (Slave)	5			200	ns

(V_{DD} = 4.75 to 5.25 V, V_{SS} = 0 V, T_A = 0° C to 70° C) External load 1 pF

Num.	Description	Fig.	Min.	Type	Max.	Units
80	DASb_IN rising to DAL_ENb rising (Slave)	5	2		6	ns
81	HLDAb falling to A_ENb falling (single DMA)	3		200		ns
82	HLDAb falling to DA_ENb falling	3		200		ns
83	DALOb_OUT falling to DA_ENb falling (single DMA)	3	2		6	ns
84	HLDAb falling to DAL_ENb falling (single DMA)	3		200		ns
85	DAL_ENb active to DAL_OUT stable (single DMA)	3, 4	2		6	ns
86	ALEb_OUT falling to DAL_ENb rising (single DMA)	3, 4		52		ns
87	DASb_OUT rising to A_ENb rising (single DMA)	3, 4		100		ns
88	DASb_OUT rising to DA_ENb rising (single DMA)	3		100		ns
89	DA_ENb rising to ALEb_OUT rising (single DMA)	3		100		ns
90	DAL_OUT hold time after DAL_ENb rising (Slave)	3		20		ns
91	DA_ENb falling to DASb_IN rising (Slave Read)	5		100		ns
92	DA_ENb falling from DASb_IN falling (Bus Slave Read)	5		100		ns
93	DA_ENb falling from DASb_IN falling (Bus Slave Write)	6		100		ns

NOTE: The READYb_IN setup time before negation of DASb_OUT is a function of the synchronization time of READYb_IN. The synchronization must occur within 100 ns. The setup time is 100 ns plus any accumulated propagation delays. Ready slips occur on 100 ns increments. No wait states are added by the NCR92C990 if either parameter number 57 or 70 is met.

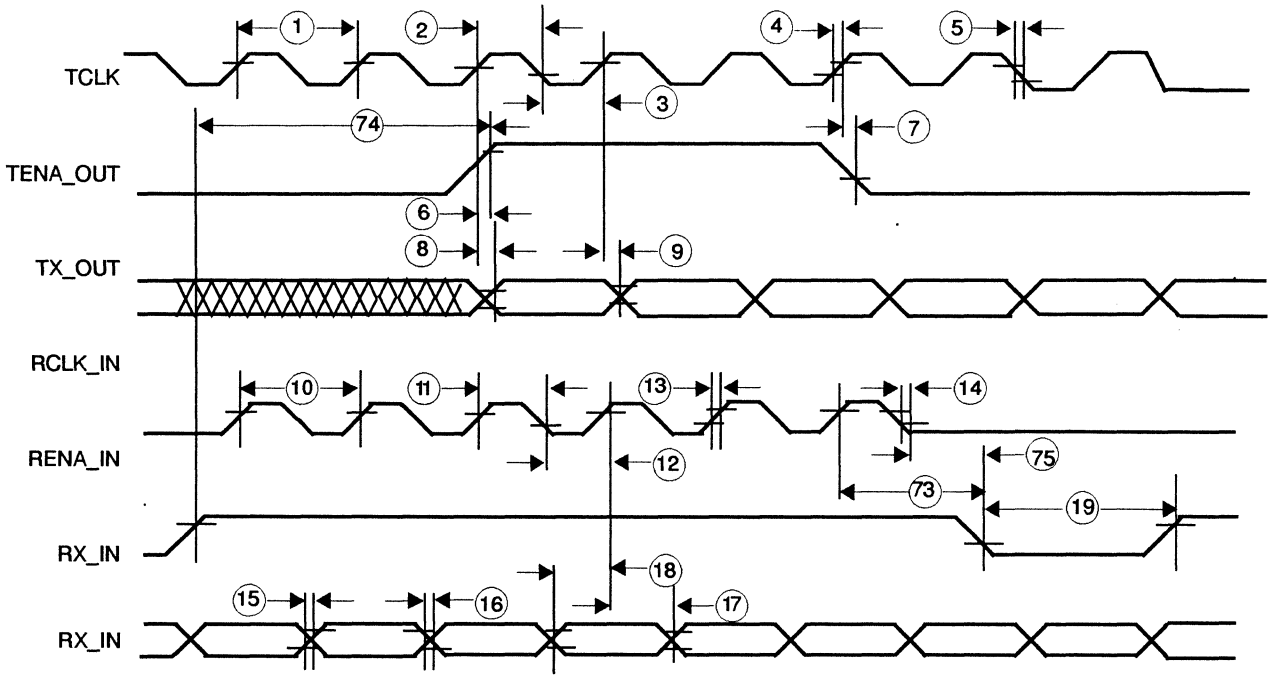


Figure 4-3 Transmit/Receive Clock

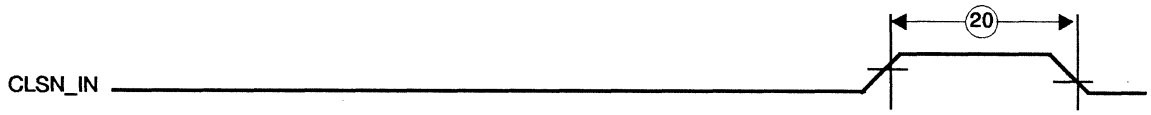


Figure 4-4 Collision

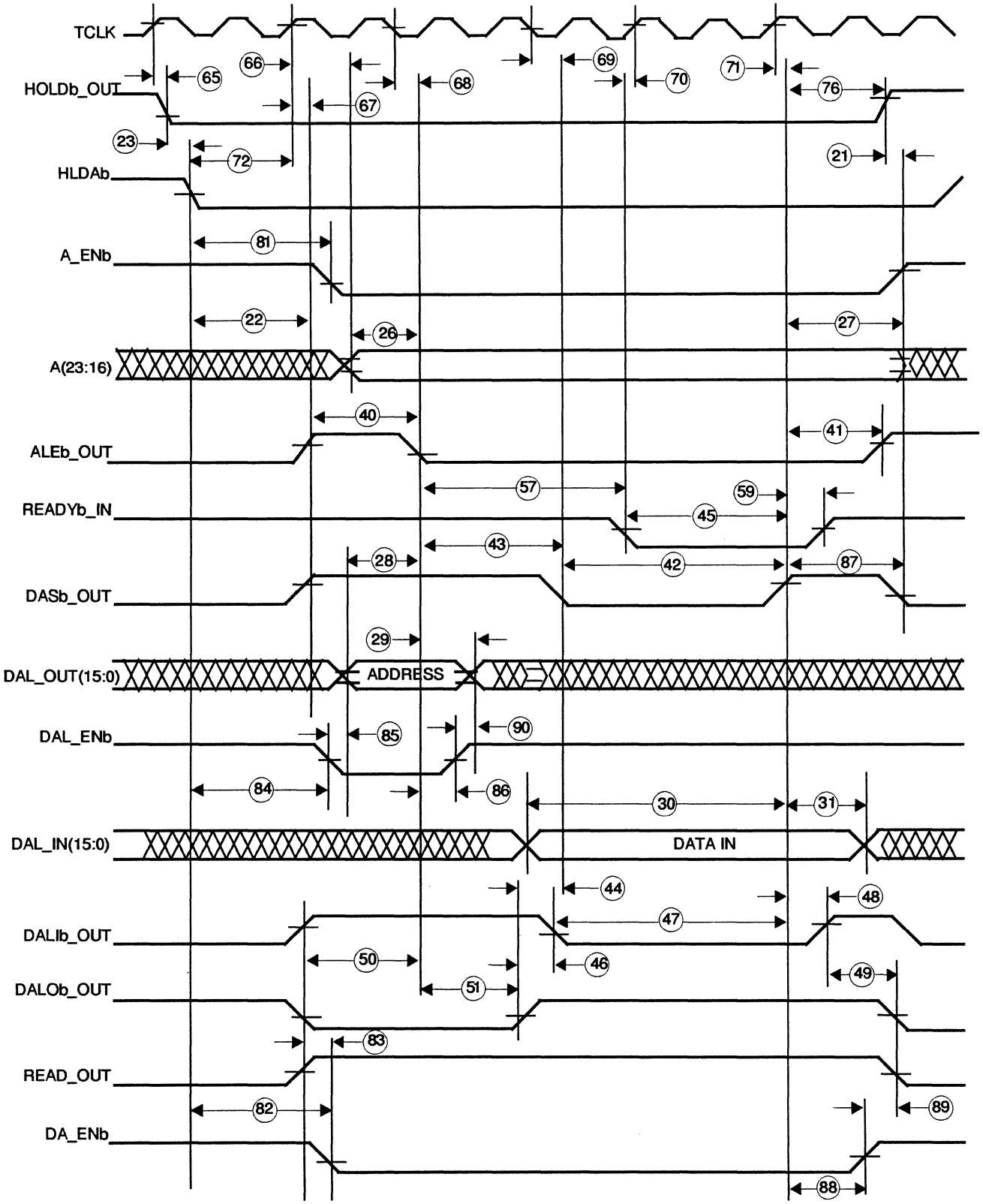


Figure 4-5 Bus Master Read (DMA Burst)

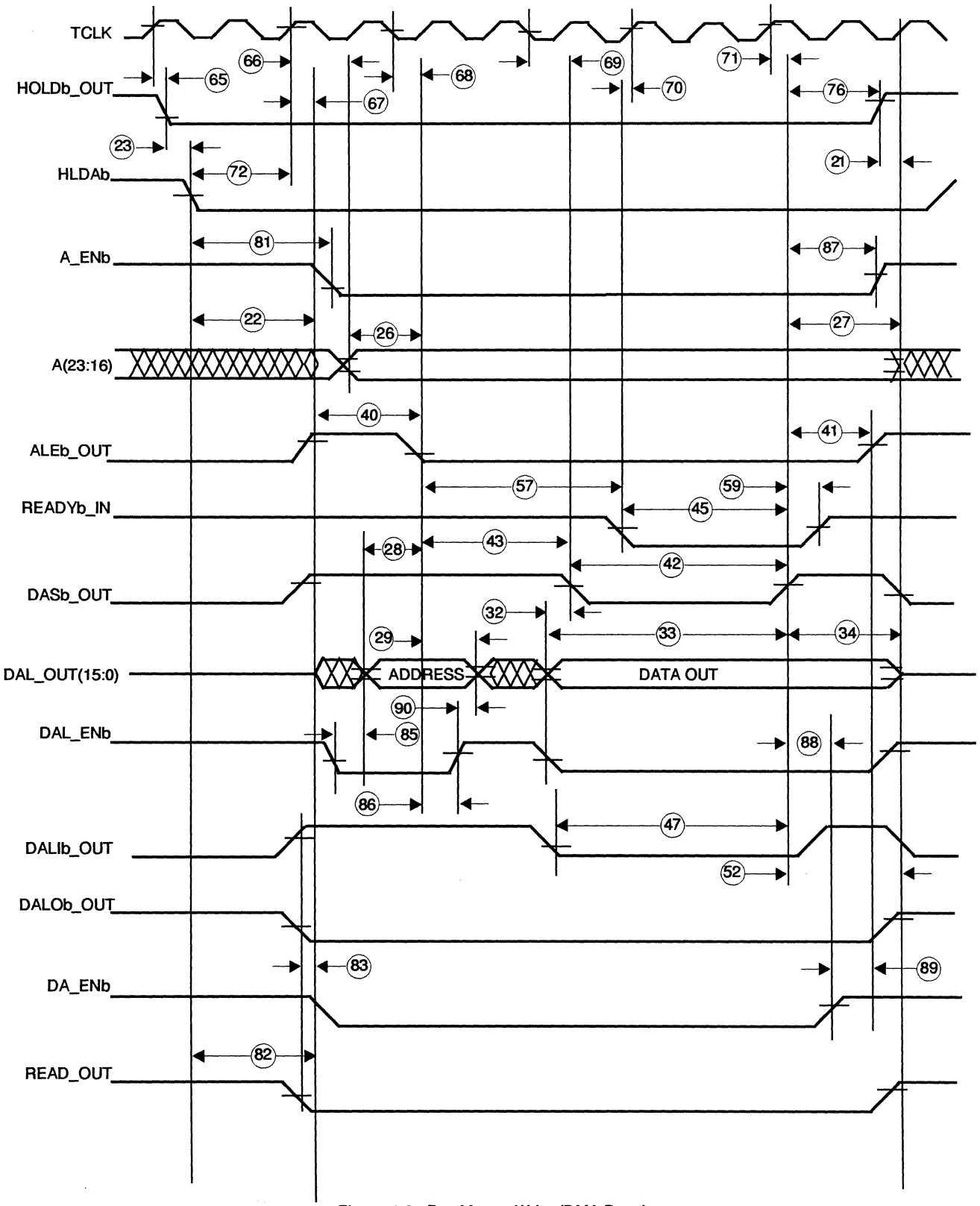


Figure 4-6 Bus Master Write (DMA Burst)

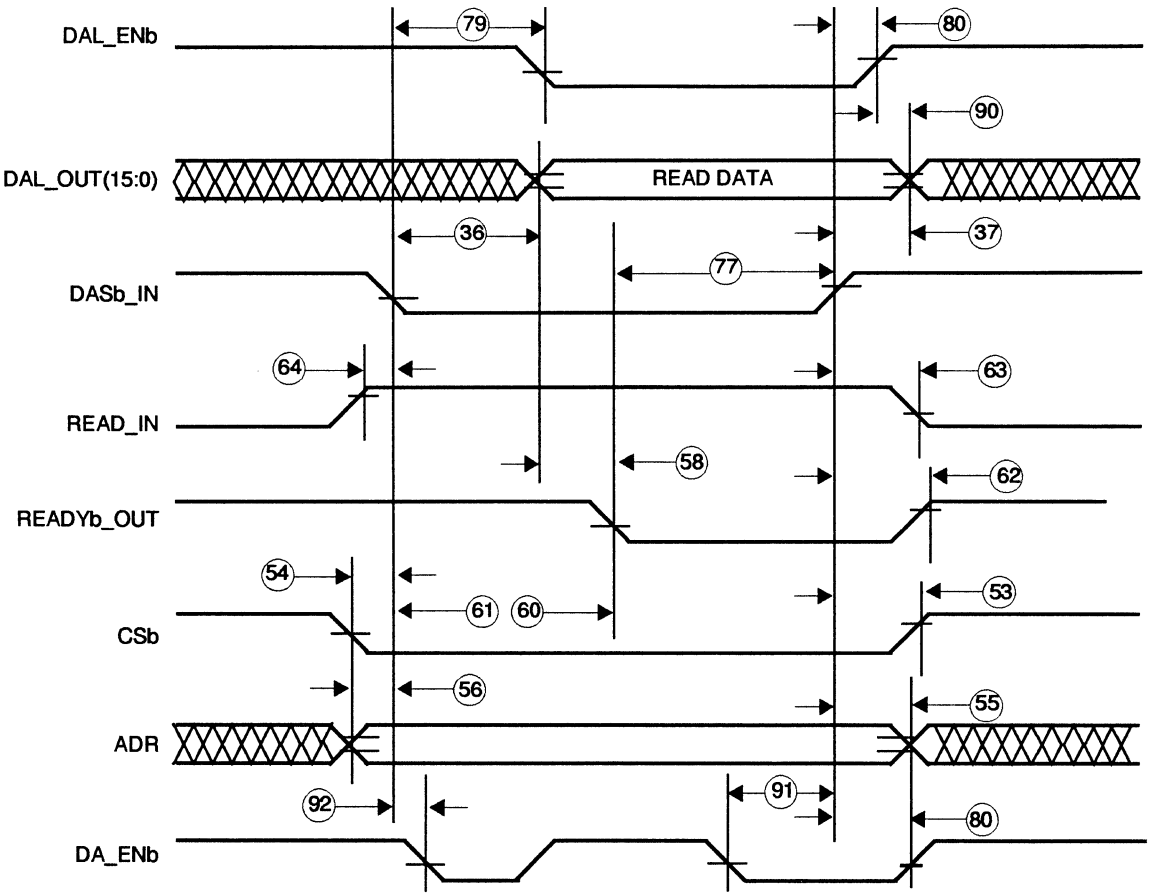


Figure 4-7 Bus Slave Read

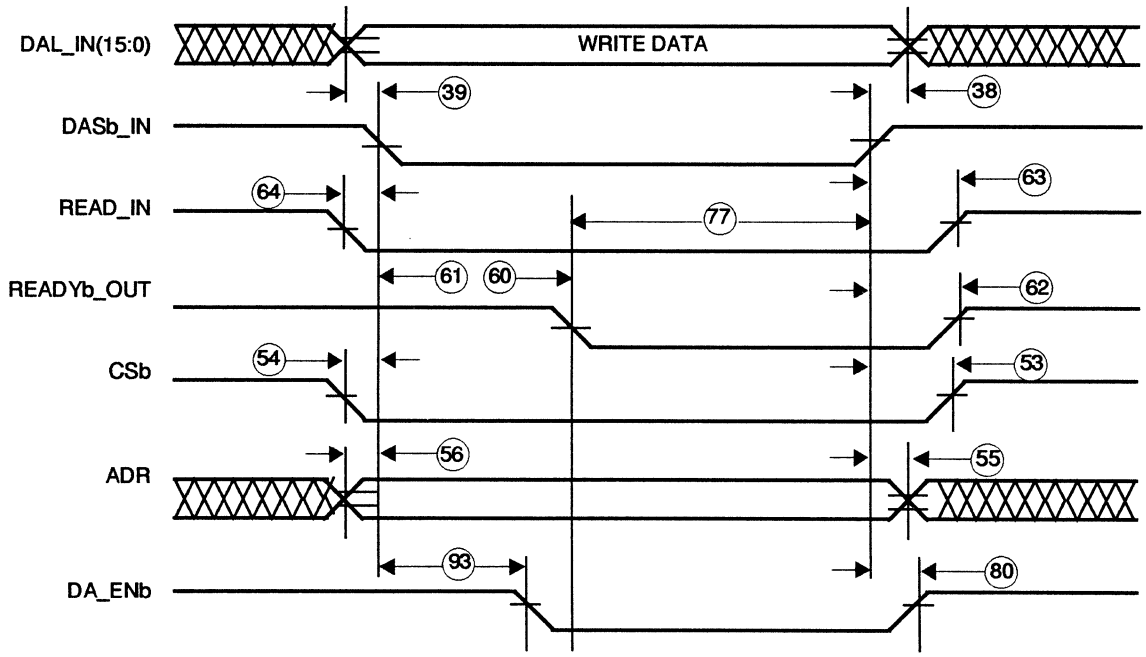


Figure 4-8 Bus Slave Write

Compatibility Verification Procedures

The design and verification of high-level macrocell is directed towards functional compatibility with the existing standard part. The macrocell is designed and verified to match data sheet functionality (when possible), then additional steps are taken to match undocumented functionality.

Creation of a Compatibility Model

A compatibility model of the new macrocell is generated from published sources of information on the standard part including functional specifications, data sheets, application notes, review sheets and errata sheets. Exceptions to standard part behavior or function will be identified.

Utilization of Industry Consultants

NCR utilizes the advice of industry consultants who are highly knowledgeable about the operation of the standard part. This additional data includes features and functions which would be apparent to someone who is very familiar with the functionality of the part but would not necessarily be covered in available documentation.

Use of Hardware Modeling Techniques and Extraction of System Patterns for Design Verification

Compatibility test vectors for the part are developed from two sources. First, a device model is created on a hardware modeling system (HML) to create test vectors based on the actual operation of the industry standard part. A second set of vectors is generated by extracting functional and timing information from the compatibility documentation and creating patterns based on this information.

In-system Test and Firmware Verification

To ensure that the macrocell is truly representative of the standard part's functionality, a system test is executed on new sample parts. The test includes the following system parameters to detect any incompatibilities or inconsistencies:

- Selection of controllers and systems
- Special test programs
- System diagnostics
- Selection of operating systems
- Selection of application programs

Final Test Plan

Throughout the compatibility verification process, final test procedures for the macrocell are emphasized. The NCR compatibility procedures provide assurance that the macrocell will function as the standard part does with documented exceptions where warranted.

Kit Parts

Provision of kit parts that are plug-compatible with the existing standard part is a very important aspect of in-system verification. The availability of kit parts allows the customer to verify macrocell operation and compatibility with the existing system design and software. Kit parts also provide a convenient way to breadboard "System Solutions in Silicon".

NCR 53C9X
Advanced SCSI Controller
Contents

NCR 53C9X
Advanced SCSI Controller

General Description

The 53C9X FSC (Fast SCSI Controller) is a high performance CMOS device designed for SCSI (Small Computer Systems Interface). It is a super-set of the 53C90A, providing fast SCSI operation, with additional commands and an additional configuration register. The C9X is intended to directly replace a C90A in an existing design, allowing an easy upgrade to SCSI-2. It is 100% compatible with existing 53C90A software.

The C90 family reduces protocol overhead by performing critical SCSI algorithms, or sequences, in response to a single C90 command. The C9X will operate at sustained data transfer rates of 10 MB/s in synchronous mode and 7 MB/s in asynchronous mode.

Independent of microprocessor intervention, the FSC performs arbitration, selection, or reselection. It also independently handles message, command, status, and data transfer between the chip's 16-byte internal FIFO, or a buffer memory.

Features

- ANSI X3.131-1986 SCSI and X3.131-199X SCSI-2 compatible
- Parity generation, optional checking
- Programmable transfer period
- Programmable offset
- SCSI-2 tagged-queueing
- 16-byte FIFO
- 12 MB/s DMA interface burst transfer rate (FASTCLK disabled)
- Up to 7 MB/s asynchronous SCSI
- Up to 10 MB/s synchronous fast SCSI and 5MB/s normal SCSI
- Up to 13.3 MBPs DMA burst transfer rate (FASTCLK enabled)
- 10 to 40 MHz clock rate
- 24-bit transfer counter eliminates inter-sector transfer delays and allows single transfers up to 16 MB
- High performance CMOS technology

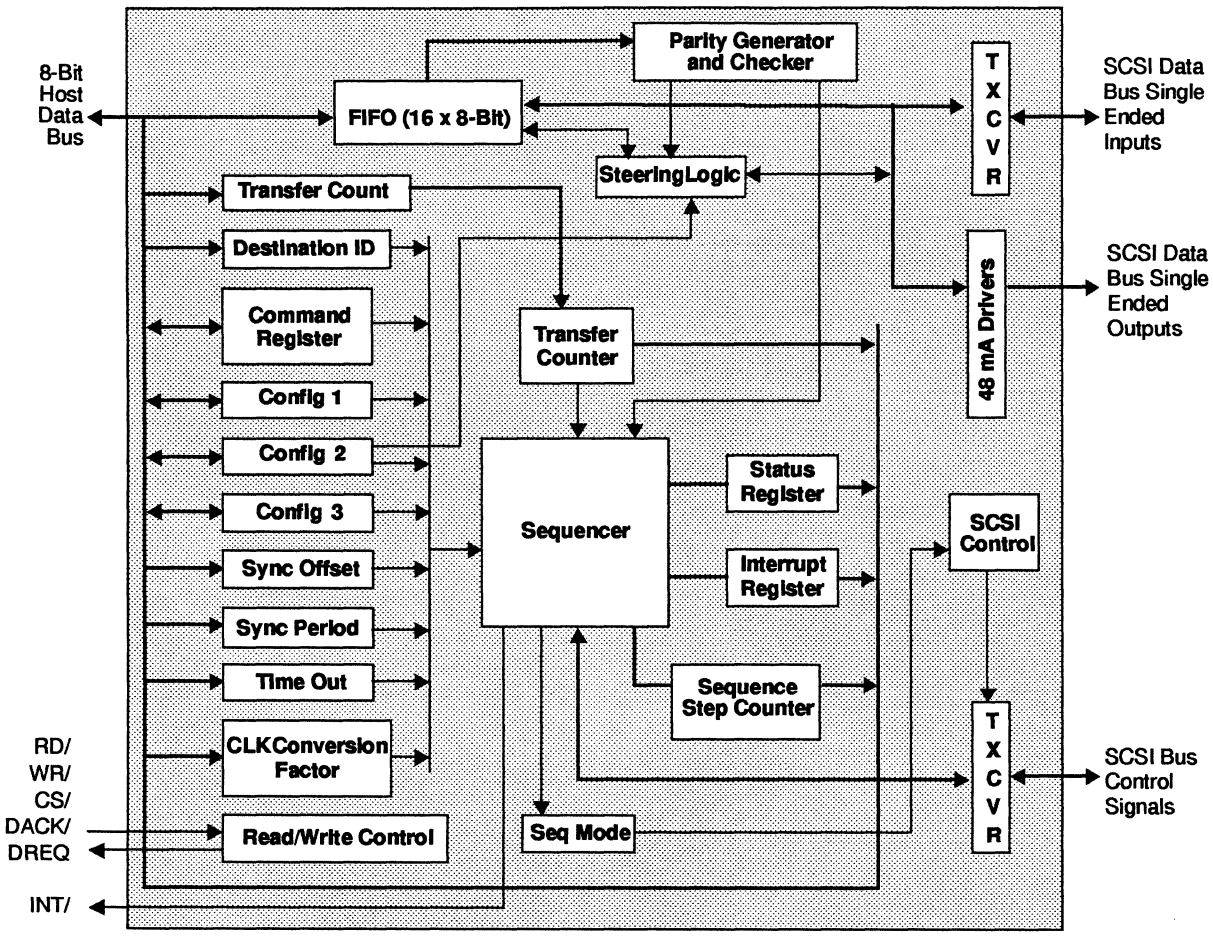


Figure 5-1 Functional Block Diagram

Pin Descriptions

Table 5-1 Host Processor and DMA Interface Pins

Signal	Type	Description
DB7-DB0	B	Active-high data bus connected to the DMA controller, CPU and buffer memory. Each pad contains a pull-up to V_{DD} (12.5K minimum).
RESET	I	Active-high chip reset. Reset must be asserted for two CLK periods, minimum, after the voltage on the power pins has reached V_{DD} minimum. This input must not be connected to RESETO.
RESETO	O	Active-high reset output. This output is always asserted when the RESET input is true OR may be asserted when the SCSI reset signal is active if bit 6 of the Config 1 register is cleared and the host has not serviced the interrupt (generated because of SCSI reset) within 1-2 ms (depending on CLK frequency and clock conversion factor). Refer to *Bus Initiated Reset*.
INT/	O	Active-low, open drain interrupt signal to the microprocessor. It is latched on the rising edge of CLK and may be cleared by reading the interrupt register or by a host hardware reset, or by a host software reset (but not by a SCSI reset). This output cannot be disabled internally.
A3-A0	I	Active-high address bus which specifies one of the FSC's internal registers for reading or writing. Used with CS/, ignored with DACK/.
CS/	I	Active-low chip-select signal that enables access to the FSC's internal registers. CS/ accesses any register, including the FIFO, while DACK/ accesses only the FIFO. CS/ and DACK/ must never be active at the same time.
RD/	I	Active-low read signal that enables FSC data onto DB7-DB0. CS/ or DACK/ must also be active.
WR/	I	Active-low write signal that strobes DB7-DB0 data into the FSC. CS/ or DACK/ must also be active.
DREQ	O	Tristate active-high DMA request to the DMA controller. DREQ will be true as long as the FIFO has at least one byte to send to memory, or has room to receive at least one byte from memory, depending on data direction.

Table 5-1 Host Processor and DMA Interface Pins (Continued) (Continued)

Signal	Type	Description
DACK/	I	Active-low DMA acknowledge from the DMA controller. DACK/ accesses the FIFO only, while CS/ accesses any register, including the FIFO. CS/ and DACK/ must never be active at the same time. DACK/ must toggle true then false for every byte transferred. Refer to *DREQ Hi Z Bit in Config 2.*
CLK	I	Clock signal. Generates internal core timing. Maximum frequency is 40 MHz (FASTCLK bit set) or 25 MHz (FASTCLK bit reset).

Table 5-2 SCSI Bus Interface

Signal	Type	Description
SDI0/-SDI7/, SDIP/	B	Schmitt trigger, active-low SCSI data/parity bus. These inputs are SCSI data bus signals.
SDO0/-SDO7/ SDOP	O O	48 mA, open drain SCSI data parity bus. These outputs are active-low SCSI data signals.
SELO/	O	48 mA, open drain SCSI select signal. This output is active-low. Asserted by the FSC to select a Target or reselect an Initiator.
BSYO/	O	48 mA, open drain SCSI busy signal. This output is active-low. Asserted by the FSC to gain use of the SCSI bus.
REQO/	O	48 mA, open drain, active-low SCSI request signal. This output is only asserted when the FSC is in target mode to request a data transfer over the SCSI bus.
ACKO/	O	48 mA, open drain, active-low SCSI acknowledge signal. This output is only asserted when the FSC is in initiator mode to acknowledge a request for a data transfer over the SCSI bus.
MSGO/ C/DO/, I/OO/	O	48 mA, open drain, active-low SCSI phase signals. These outputs are only asserted when the FSC is in target mode.
ATNO/	O	48 mA, open drain, active-low SCSI attention signal. This output is only asserted when the FSC is in initiator mode. Alerts the Target SCSI device that the FSC has a command or message transfer.

Table 5-2 SCSI Bus Interface (Continued)

Signal	Type	Description
RSTO/	O	48 mA, open drain SCSI reset signal. In single-ended mode this output is active-low. The FSC drives this signal true only when the host writes the SCSI bus reset command to the command register.
SELI/	I	Schmitt trigger, active-low SCSI select input. Selects the FSC as a Target or Reselects the FSC as an Initiator
BSYI/	I	Schmitt trigger, active-low SCSI busy input. Indicates to the FSC that the SCSI bus is in use.
REQI/	I	Schmitt trigger, active-low SCSI request input. Indicates to the FSC, as Initiator, that a Target device is requesting a data transfer on the SCSI bus.
ACKI/	I	Schmitt trigger, active-low SCSI acknowledge input. Asserted by an initiator to acknowledge a request by the FSC for a data transfer on the SCSI bus.
MSGI/	I	Schmitt trigger, active-low SCSI message input. Indicates to the FSC that a Target device has initiated a SCSI Message In or Message Out phase.
C/DI	I	Schmitt trigger SCSI control/data input. Signals the FSC that a Target device is transferring control or data information on the SCSI bus.
I/OI	I	Schmitt trigger SCSI input/output input. Alerts the FSC, as Initiator, that the direction of data movement on the SCSI bus is into the chip. This signal also distinguishes between Selection and Reselection phases.
ATNI/	I	Schmitt trigger, active-low SCSI attention input. Indicates to the FSC that an Initiator device is ready to transfer a message or a command.
RSTI/	I	Schmitt trigger, active-low SCSI reset signal. When this input is true, the FSC will automatically disconnect from the SCSI bus.
IGS	O	Active-high initiator group select signal. This pin is high whenever the FSC is in initiator mode. Used to enable external drivers for the Initiator signals ACKO/ and ATNO/.
TGS	O	Active-high target group select signal. This pin is high whenever the FSC is in target mode. Used to enable external drivers for the Target signals REQO/, MSGO/, C_DO/, and I_OO/.

Functional Description

The FSC has a command set that allows it to perform common SCSI sequences at hardware speed without host intervention. Its on-chip FIFO may be accessed simultaneously by the SCSI bus and either the host processor or the host DMA controller. All command, data, status, and message bytes pass through the FIFO on their way to or from the SCSI bus. Most FSC commands have two versions: DMA and non-DMA. When DMA instructions are used, data will pass between memory and the SCSI bus with the FIFO acting as temporary storage when the DMA channel is temporarily shut down by a higher priority event such as DRAM refresh.

The FIFO also helps speed execution during non-DMA transfers. For example, in initiator role, the host processor will load the CDB (Command Descriptor Block) and optionally one or three message bytes into the FIFO, issue one of several selection commands and wait for an interrupt. The FSC will wait for bus-free, arbitrate for the bus again and again until it acquires it, send the message bytes followed by the CDB, then generate an interrupt. Meanwhile, a multi-tasking host may continue with other tasks.

An 8-bit part unique ID code for the FSC is available in the Transfer Counter High register, at address 0Eh, when the following conditions are true:

- After power-up or chip reset
- Before the Transfer Counter High register is loaded
- The features Enable bit (bit 6 in the Configuration 2 register) is set
- A DMA NOP command (80h) has been issued

The lower three bits indicate the revision level, while the upper five bits indicate the chip family code.

In target role, the host processor will enable selection, then wait for an interrupt. Eventually, an initiator will select the FSC and will then automatically step through the arbitration, selection, and command phases before generating an interrupt. When the interrupt occurs, the entire command descriptor block will be in the FIFO along with any message bytes sent by the initiator. Combination commands such as these, are identified with the *sequence* suffix in the *Table of FSC Commands*.

After selection phase has been successfully completed, the FSC may transfer bytes in any of the SCSI information phases whether operating in initiator or a target role. The FSC supports disconnect/reselect in both initiator and target roles, making high performance multi-threaded systems easy to implement.

The FSC may transfer data phase bytes across the bus synchronously, at speeds up to 10 MB/s, or asynchronously at speeds up to 7 MB/s. Refer to *Data Transfer Rate*. The difference between the two is transparent to the user except that the Synchronous Offset and the Synchronous Transfer Period registers must be programmed prior to synchronous data transfer. The default, after hardware or software reset, is asynchronous transmission.

Data bytes will usually be transferred using DMA. The host processor will program an external DMA controller, program the FSC transfer count, issue an FSC data transfer command (there are several), and then wait for an interrupt. The DMA controller and the FSC will transfer all the data without host processor intervention.

To end the SCSI transaction, the FSC target will place a status byte and a message byte in the FIFO, then issue a single command (there are two to chose from) which will cause the FSC to first assert status phase, send the first byte, assert message in phase, send the second byte, disconnect from the SCSI bus (after the initiator releases ACK (Acknowledge)) and interrupt the host processor.

The end of a SCSI transaction is similar for an FSC initiator except that it receives two bytes into its FIFO. The initiator prevents the target from disconnecting by holding ACK asserted on the bus while the host processor examines the status and message bytes. If both bytes are good, the message accepted command is used to instruct the FSC to release ACK, which allows the target to disconnect which causes the initiator to interrupt its host and report the disconnect. If the status and message bytes are not good, the host should first issue the set ATN (Attention) command before issuing the message accepted command. This instructs the FSC to assert ATN before releasing ACK, which should cause the target to request message out phase rather than disconnect.

Bus Initiated Sequences

- Selection
- Reselection
- SCSI bus reset

Selection or reselection sequences occur in the disconnected state when the FSC is selected or reselected by another initiator or target, if the Enable Selection or Reselection command had previously been received by the FSC.

In addition to responding to bus initiated events, the FSC may initiate a bus event by using one of several selection or reselection commands. If one of these commands starts executing, it will clear Enable Selection/Reselection after arbitration has been won. Normally the host processor will have 250 ms (ANSI recommended selection time-out period) after the chip disconnects from the bus to re-enable bus initiated events. If the time-out is exceeded, an initiator or target which is attempting to connect to the FSC, may time-out and abort.

If, on the other hand, the bus initiated event occurs before the command starts executing, the FIFO and command register will be cleared, and any further writes by the host processor will be ignored until the interrupt register is read. Since a selection/reselection command requires that something be placed in the FIFO, these bytes will be lost, as will any command written to the Command register. The interrupt handler that services a selection/reselection command will have to examine the bits in the Interrupt register to determine if the FSC selected another device, or if it was selected by another device. The former case will cause a Function Complete interrupt, the latter case will cause a Selection/Reselection interrupt.

Bus Initiated Selection

When the FSC has been selected as a target, the following data will be in its FIFO:

- Bus ID
- Identify message
- Optional two-byte command queuing message
- Command Descriptor Block (CDB)

The bus ID will always be present and will always be one byte. It is an un-encoded version of the state of the bus during selection phase. Any SCSI data bits that were true during selection phase will be set. The target ID (our ID) must always be set. In arbitrating systems, the initiator ID must also be set. The initiator ID is optional in non-arbitrating systems.

The identify message, if sent, will be placed in the FIFO and will always be one byte in SCSI-1 systems but may be one or three bytes in SCSI-2 systems. If the initiator does not send an identify message (does not select with ATN), a null byte (00 hex) will be placed in the FIFO. If the FSC is selected with ATN false, it will store a null byte (00) in the FIFO behind the bus ID, then begin requesting command phase bytes. A detected parity error will cause the FSC to interrupt and stop, if parity checking is enabled.

If the initiator selects with ATN and the SCSI-2 bit is cleared, the FSC target will request one message byte and will place it in the FIFO behind the bus ID. The FSC will then begin requesting command phase bytes unless the message byte is not a valid identify message, or a parity error is detected, which will cause the FSC to interrupt and stop. The sequence step register should then be examined.

If the initiator selects with ATN and the SCSI-2 bit is set, the FSC will examine both the message byte and the ATN signal to determine how many bytes to request. If the first byte is a valid identify message and if ATN goes false after receiving the first byte, the FSC will only request one message byte. If the first byte is a valid identify message byte and ATN is still true, it will request two more message bytes. The FSC will then begin requesting command phase bytes unless the first byte was not a valid identify message, or a parity error was detected, or ATN went false between the 2nd and 3rd bytes, or ATN remained true but the SCSI-2 bit was false, which will cause the FSC to interrupt and stop. The sequence step register should then be examined.

The Command Descriptor Block (CDB) will be placed in the FIFO behind the message byte(s), assuming selection completed normally. The CDB may be 6, 10 or 12 bytes long. Thus, in SCSI-2, the entire FIFO may be filled if a tagged-queue 12-byte command is used.

Bus Initiated Reselection

The FSC will allow itself to be reselected as an initiator by a target if it has previously received the enable selection/reselection command. If the sequence completes normally, the following information will be in the FIFO:

- Bus ID
- Identify message
- Optional 2-byte queue tag message

The bus ID is the same as the selection case, described above. The identify message will always be present and always be one byte. If queue tagging is enabled, and the target is sending a queue tag message, the target will also send two queue tag message bytes.

Bus Initiated Reset

A bus initiated reset will be recognized by the FSC at any time. The FSC will then disconnect from the bus and reset its internal sequencer. If the SCSI reset reporting bit (Config 1 register) is not set, the FSC will generate a SCSI reset detected interrupt. If the host processor does not read the interrupt register within t_1 milliseconds, the FSC will assert RESETO for t_2 microseconds.

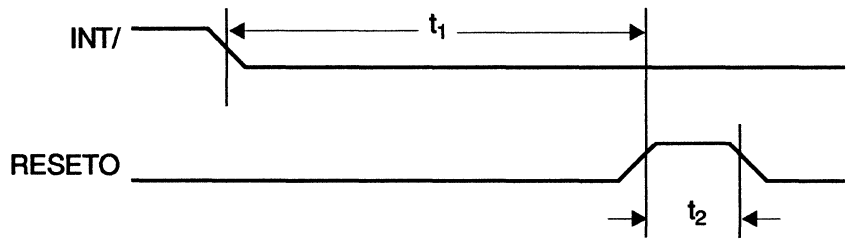


Figure 5-2 Interrupt and Reset Timing

t_1 = 2 (CLK period) (3841CCF - 1)
t_2 = 130 (CLK period) (CCF)

Where CCF = Clock Conversion Factor
Refer to *Description of Write Register 09*.

For example, at CLK = 25 MHz
t_1 = 1.5 milliseconds
t_2 = 26 microseconds

Data Transfer Rate

Performance claims for the FSC are based on it being directly connected to the SCSI bus with no external transceivers. In differential mode, external transceivers are required and will slow asynchronous transmission by the propagation delay of the chosen transceiver but will not slow synchronous transmission.

The synchronous data transmission rate is equal to the CLK input frequency multiplied by the encoded value in the Synchronous Transfer Period register. Sustained synchronous transfer rates of 10 MB/s are attainable across the commercial voltage and temperature range.

The FSC can transfer synchronous SCSI data in both initiator and target modes at transfer rates up to 10 MB/s, using an input clock frequency of 40 MHz. The SCSI-1 and Fast SCSI-2 minimum timing requirements are listed in the following table:

Mode	Setup	Hold	Assert/Negate
SCSI-1	55 ns	100 ns	90 ns
Single-ended Fast SCSI-2	25 ns	35 ns	30 ns

To support maximum Fast SCSI transfer rates and SCSI-1 transfer requirements, the FASTSCSI (bit 1) and FASTCLK (bit 0) bits have been added to the Configuration 3 register. They modify the SCSI state machine to provide fast and normal synchronous timings depending upon the clock frequency.

During synchronous SCSI transfers, the assertion and deassertion of the REQ and ACK signals are programmable using the FASTCLK bit and other bits in the Synchronous Offset register. The input clock duty cycle affects the half clock assertion/deassertion delays. Note that DMA must be used for synchronous transfers.

The asynchronous transmission rate will vary with cable length and the CLK period. The FSC can reach sustained transfer rates of 7 MB/s on short (one-foot) cables using typical devices operating at or near nominal voltage and temperature. The worst case asynchronous transmission rate over voltage, temperature, and process variations is 3 MB/s on a maximum length (six meters), single-ended cable and 4 MB/s on a one-foot cable.

The asynchronous transmission rate is only slightly affected by the CLK frequency when sending data. The FSC will drive the data bus for a minimum of one CLK period before asserting REQ or ACK. The CLK frequency does not affect the asynchronous transfer rate when receiving data.

Two termination methods for single-ended mode are described in the ANSI SCSI-2 specification. Alternative 1 reflects the SCSI-1 specification, with 220 ohms to the TERMPWR line and 330 ohms to ground. To improve the noise margins, 1% resistors should be used and TERMPWR should be between 5.0 V and 5.25 V.

Alternative 2 reflects the SCSI-2 specification, and is the recommended termination method. An adjustable voltage regulator, powered by TERMPWR, supplies 2.85 V to 110 ohm 1% resistors. This more closely matches the characteristics impedance of the cable, resulting in better signal quality. Integrated versions from multiple vendors are available.

Register Set

Table 5-3 FSC Register Set

Address (hex)	Read	Write
0	Transfer Counter Low	Transfer Count Low
1	Transfer Counter Mid	Transfer Count Mid
2	FIFO	FIFO
3	Command	Command
4	Status	Destination Bus ID
5	Interrupt	Select/Reselect Timeout
6	Sequence Step	Synchronous Transfer Period
7	FIFO Flags	Synchronous Offset
8	Configuration 1	Configuration 1
9	Reserved	Clock Conversion Factor
A	Reserved	Test Mode
B	Configuration 2	Configuration 2
C	Configuration 3	Configuration 3
E	Transfer Counter High	Transfer Counter High

Some FSC registers have different meanings during reads than writes. When CS/ is true, the register being accessed is determined by either RD/ or WR/ together with the address pins A0-3. The FIFO may be accessed using either CS/ or DACK/ together with RD/ or WR/. Address pins A0-A3 are ignored when DACK/ is active, but must be driven when CS/ is active.

Transfer Count (Write Address 0, 1)

These two registers, together with the Transfer Counter High register, form a 24-bit transfer count for DMA operations. Transfer count specifies the number of bytes that are to be transferred over the SCSI bus. Values written to these two registers will be stored internally and loaded into the transfer counter by any DMA command. These values remain unchanged while the transfer counter decrements. Thus, successive blocks of equal size may be transferred without reprogramming the count. They may be reprogrammed any time after the previous DMA operation has started, whether it has finished or not. When the Features Enable bit (bit 6 in the Configuration 2 register) is clear, disabling the Transfer Count High register, a zero in registers 00h and 01h specifies a maximum length count of 64 K. When the Features Enable bit is set, and the Transfer Count High register is enabled, zeros specify a maximum length

count of 16 MB. These registers are not changed by any reset, and their states are unpredictable after power-up.

Transfer Counter (Read Address 0, 1)

These registers combine with the Transfer Counter High register to form a 24-bit transfer counter for DMA operations. A read from these two addresses will return the value currently in the counter. DMA commands use the counter to terminate a transfer. Any DMA command will load count into the counter. A DMA NOP 80h will load the counter while the non-DMA NOP 00h will not.

With one exception, non-DMA commands do not use the counter. The exception is when the FSC has been selected, it decodes the group code field of the CDB (Command Descriptor Block), loads the counter with the number of bytes in the CDB, then decrements once for every byte received.

The transfer counter decrements on the leading edge of:

Target	Decremented by
Data in Phase	DACK/
Data out Phase	REQO/

Initiator	Decremented by
Synchronous data in	DACK/
Asynchronous data in	ACKO/
Data out	DACK/

NOTE: DACK/ can decrement the counter even if RD/ or WR/ do not go true. False DACK/s can cause the counter to get out of sync with the data stream, leading to subtle errors that are difficult to trace. When false DACK/s are expected to interfere with a temporarily suspended DMA operation, the DREQ Hi-Z bit in Config 2 should be set.

FIFO Register (Read/Write Address 02)

The FIFO is a 16 by 9-bit first-in-first-out buffer between the SCSI bus and memory. It is accessible by the host processor at this address. It is also accessible by an external DMA controller and by the SCSI bus. The DMA may access the FIFO by asserting DACK/ together with either RD/ or WR/. When accessed by CS/, the address bits must be valid. When accessed by DACK/, the address bits are ignored. The bottom FIFO element and the FIFO flags are initialized to zero during hardware reset, software reset chip and at the beginning of bus initiated selection or reselection. The contents of the rest of the FIFO are not changed by any reset, but when the flags are zero, successive FIFO reads will always access the bottom register. It is cleared if the phase changes from an output phase to Synchronous Data In during a Transfer Pad or Transfer Information command.

Command Register (Read/Write Address 03)

The command register is a two deep 8-bit read/write register used to give commands to the FSC. Up to two commands may be stacked in the command register. The second command may be written before the FSC completes (or even starts) the first. Reset chip, reset SCSI bus and target stop DMA execute immediately, all others wait for the previous command to complete. The last executed (or executing) command will remain in the command register and may be read by the host processor. Reading the command register has no effect on its contents. The command register will be cleared by any of the following conditions:

- Hardware, software or SCSI bus reset
- SCSI bus disconnect
- Bus-initiated selection or reselection
- Select command
- Reconnect command if ATN is set
- Select or reselect time-out
- Target terminate command
- Parity error detected in target mode
- Assertion of ATN in target mode
- Any phase change in initiator mode
- Illegal command

If two commands are placed in the command register, two interrupts may result. If the first interrupt is not serviced before the second finishes, the second interrupt is stacked behind the first. When the interrupt register is read by the host to service the first interrupt, the contents status register, sequence step register, and interrupt register will change to describe the second interrupt.

If the Features Enable bit is not set, a SCSI Bus reset clears the Command register FIFO, but does not cause it to be held reset until the Interrupt register is read. Then if the Command register is loaded while the register is in the reset condition, and before the Interrupt register is read, the chip attempts to execute the loaded command causing undesirable command execution. This condition is avoided by setting the Features Enable bit.

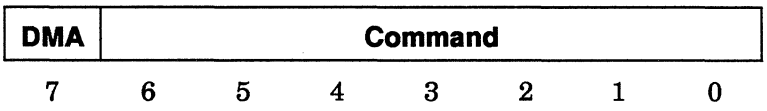


Figure 5-3 Command Register (Read/Write Address 03)

Bit 7 (Enable DMA)

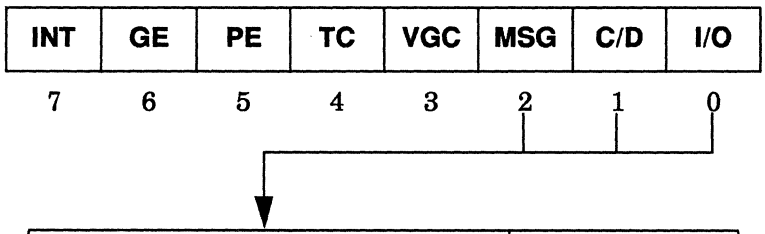
When bit 7 is not set, the command is a non-DMA instruction. When it is set, the command is a DMA instruction. DMA instructions will load the internal byte counter with the value in the transfer count register (without changing the count register) then transfer data until that count decrements to zero. If the transfer terminates prematurely, the bits in the status, sequence step, and interrupt registers will indicate why.

Bits 6-0 (Command Code)

The FSC commands are shown in Table 19. Bits 4, 5 and 6 specify a mode group. Commands from the miscellaneous group may be issued at any time. Commands from the disconnected, target or initiator groups will only be accepted by the FSC if it is in the same mode as the command when it falls to the bottom of the command FIFO. Otherwise, an illegal command interrupt will be generated. For example, after hardware or software reset, the FSC will be in the disconnected state. A command from either the target group or the initiator group will cause an illegal command interrupt. An enable selection or reselection command by itself will not change modes. However, if another SCSI device then selects the FSC, it will be in the target state; if another device reselects the FSC, it will then be in the initiator state. Similarly, any select command will place the FSC in initiator mode, while the reselect sequence command will place the FSC in target mode.

Status Register (Read Address 04)

The status register contains important flags that indicate various conditions. All but the phase bits are latched. The phase bits are live indicators of the state of the SCSI bus. All the latched bits except the terminal count are cleared by reading the interrupt register.



Bits 2	Bits 1	Bits 0	SCSI Phase
0	0	0	Data Out
0	0	1	Data In
0	1	0	Command
0	1	1	Status
1	0	0	ANSI Reserved
1	0	1	ANSI Reserved
1	1	0	Message Out
1	1	1	Message In

Figure 5-4 Status Register (Read Address 04)

Bit 7 (Interrupt)

This bit is set whenever the FSC drives the INT output true. It may be polled. It is buffered from the actual output, so that in wired-OR (shared interrupt) designs, this bit will indicate whether the FSC is attempting to interrupt the host processor. Hardware reset or software reset chip or a read from the interrupt register will release an active INT signal and also clear this bit.

Bit 6 (GrossError)

This bit is set when one of the following has occurred:

- The top of the FIFO is overwritten
- The top of the command register has been overwritten
- Direction of DMA transfer is opposite to the direction of the SCSI transfer
- An unexpected phase change in initiator role during synchronous data phase

Gross error does not cause an interrupt, it may be detected only while servicing another interrupt. The bit is cleared by reading the Interrupt register if the interrupt output is asserted. It will also be cleared by hardware reset, or software reset chip (but not SCSI reset).

Bit 5 (Parity Error)

This bit will be set if parity checking is enabled in the Config 1 register and the FSC detects a SCSI parity error on incoming command, data, status or message bytes. It will be cleared by reading the interrupt register if the interrupt output is asserted. Hardware reset or software reset chip will clear this bit (but not SCSI reset).

Bit 4 (Terminal Count)

This bit is set when the transfer counter decrements to zero. It resets when the transfer count is loaded. Since a DMA NOP 80h command will load the transfer counter, it will also clear this bit. Note that a non-DMA NOP 00h will not load the counter and will not clear this bit. Reading the interrupt register will not clear this bit. Hardware reset or software reset chip will clear it (but not SCSI reset).

Bit 3 (Valid Group Code)

When the FSC is selected, it decodes the group code field in the first byte of the command descriptor block. If the group code matches one defined in ANSI X3.131-1986, this bit will be set. An undefined group code (designated reserved by the ANSI committee) leaves it not set. If the SCSI-2 bit is set in the Config 2 register, Group 2 commands will be recognized as ten-byte commands and the bit will be set. If the SCSI-2 bit is cleared, Group 2 commands will be treated as reserved commands. Groups 3 and 4 are always treated as reserved commands. A reserved Group command will cause the FSC to request 6 command bytes. The FSC recognizes Group 6 as six-byte vendor unique commands and Group 7 as 10-byte vendor unique commands. The valid group code bit will be cleared by reading the interrupt register if the interrupt output is asserted. It will also be cleared by hardware reset or software reset chip (but not by SCSI reset).

Bits 2-0 (Phase Bits)

These bits indicate the phase on the SCSI bus at the time the register was read. These bits are live, if the phase changes, so will these bits. In target role, the FSC is driving these lines so they will not change if the read follows an interrupt. In initiator role, the FSC will generate its interrupt only after the target asserts REQ (Request). The ANSI specification requires that the phase lines be valid before asserting REQ and remain valid until the initiator asserts ACK(Anowledge). Thus, these bits can be expected to be stable during any read that follows an interrupt.

Destination ID (Write Address 04)

The least significant 3 bits of this register specify the encoded destination bus ID for a selection or reselection command. These bits are binary encoded, with 111 representing device ID 7, which appears as 80h on the SCSI bus. The most significant 5 bits are reserved by NCR. The destination ID is not changed by any reset, the states of these bits are unpredictable after power-up.

Interrupt Register (Read Address 05)

This 8-bit register is used in conjunction with the status register and sequence step register to determine the cause of an interrupt. Reading this register when the interrupt output is true will clear all three registers. The entire interrupt register will be cleared (0) by hardware reset or software reset chip (but not SCSI reset).

SCSI RST	I11	Dis	BS	FC	Re SEL	SEL ATN	SEL
7	6	5	4	3	2	1	0

Figure 5-5 Interrupt Register (Read Address 05)

Bit 7 (SCSI Reset Detected)

This bit is set if the chip detects a reset on the SCSI bus, regardless of the status of the SCSI Reset Reporting Disable bit in the Config 1 register. If the interrupt is not serviced in one or two milliseconds, the FSC resets the host. Refer to *Bus Initiated Selection.*

Bit 6 (Illegal Command)

This bit is set when an unused code is placed in the command register or when the command is from a mode group different than the mode the FSC is currently in. Command Code 7 in the Command Register will not generate an illegal command. Refer to the *Command Register* definition.

Bit 5 (Disconnect)

In initiator mode, this bit is set when the target disconnects or a selection or reselection time-out occurs. When the FSC is in target mode, this bit is set if a Terminate Sequence or Command Complete sequence command causes the FSC to disconnect from the bus.

Bit 4 (Bus Service)

This bit indicates that another device is requesting service. In target mode, it is set whenever the initiator asserts ATN (Attention). In initiator mode, it is set whenever the target is requesting an information transfer phase.

Bit 3 (Function Complete)

This bit will be set after any target mode command has completed. In initiator mode, it is set after a target has been selected (before transferring any command phase bytes), after Command Complete finishes, or after a Transfer Info command when the target is requesting message in phase.

Bit 2 (Reselected)

This bit is set during reselection phase to indicate that the FSC has been reselected as an initiator.

Bit 1 (Selected with ATN)

This bit is set during selection phase to indicate that the FSC has been selected as a target and that ATN was asserted on the SCSI bus.

Bit 0 (Selected)

This bit is set during selection phase to indicate that the FSC has been selected as a target and that ATN was false during selection.

Time-Out (Write Address 05)

This 8-bit write-only register specifies the amount of time to wait for a response during selection or reselection. (The FSC has no way to time-out if it never wins arbitration, it will keep trying indefinitely until it wins). The time-out register is normally loaded to specify a time-out period of 250 ms. The Register Value (RV) may be calculated from:

$$RV = \frac{\text{(Time-out period) (CLK frequency)}}{\text{8192 (Clock conversion factor)}}$$

For example, at 25 MHz, the register value that gives a 250 ms time-out period is 153 decimal or 99 hexadecimal. The clock conversion factor is defined in the description of write address 9. The time-out register remains unchanged by any reset, the states of these bits are unpredictable after power-up.

Synchronous Transfer Period (Write address 6)

The lower five bits of this register specify the minimum time between leading edges of successive REQ (Request) or ACK(Acknowledge) pulses. Synchronous data will be transmitted or received at the rate of one byte every N clocks (CLK). N is related to the register value as shown below.

Register Value	Clocks per Byte
0 0 1 0 0	4
0 0 1 0 1	5
0 0 1 1 0	6
0 0 1 1 1	7
—	—
—	—
—	—
1 1 1 1 1	31
0 0 0 0 0	32
0 0 0 0 1	33
0 0 0 1 0	34
0 0 0 1 1	35

Missing entries in the table above follow the binary code. The upper three bits are reserved. This register defaults to 5 after hardware reset or software reset chip (but not SCSI reset).

Sequence Step (Read Address 06)

The lower 3 bits of this register are used to indicate how far the internal sequencer was able to proceed in executing combination commands. This counter will be incremented at certain points in various algorithms to aid in error recovery if the previous command does not complete normally.

FIFO Flags (Read Address 07)

The least significant five bits of this register indicate how many bytes are currently in the FIFO. The value is binary encoded. The flags should not be polled while transferring data because they will not be stable while the SCSI interface is changing the contents of the FIFO.

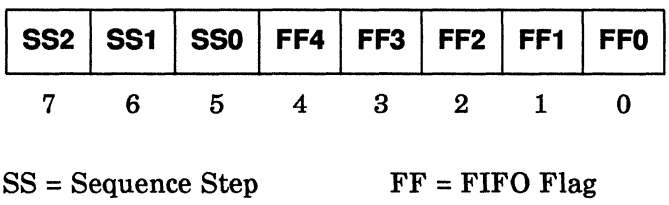


Figure 5-6 FIFO Flags (Read address 07)

The upper three bits are duplicates of the sequence step register bits when operating in normal mode.

If test mode is enabled, bit 5 is set to indicate that the offset counter is not zero. Not zero means that synchronous data may continue to be transferred. Zero means that the synchronous offset count has expired and the FSC will not transfer any more data until it receives an acknowledge.

Synchronous Offset (Write Address 07)

The least significant four bits of this register specify whether the FSC will transfer data phase bytes synchronously or asynchronously. Zero specifies asynchronous transfer. Any other value specifies the synchronous offset; the number of data phase bytes that may be sent synchronously without an acknowledge (either REQ or ACK), depending on whether the FSC is in initiator or target mode.

When transmitting to the SCSI bus, the FSC will stop sending bytes when it reaches this offset and, thereafter, send one byte for every acknowledge it receives from the other SCSI device.

When receiving from the SCSI bus, the FSC will send an acknowledge every time a byte is removed from its' FIFO on the DMA (or host processor) interface. The maximum offset of 15 allows a receiving FSC to store data in its FIFO while the external DMA controller gains control of the memory bus.

The synchronous offset is cleared (0) by hardware reset or software reset chip (but not SCSI reset).

Configuration 1 Register (Config 1) (Read/Write Address 08)

This 8-bit read/write register specifies various operating conditions for the FSC. Any bit pattern written to this register may be read back and should be identical.

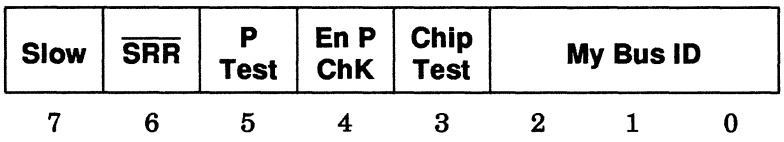


Figure 5-7 Configuration 1 Register (Config1) (Read/Write Address 08)

Bit 7 (Slow Cable Mode)

Slow cable mode will seldom be necessary. It compensates for excessive capacitive loading on the SCSI data signals by inserting an extra CLK period between data being asserted on the bus and REQ or ACK being driven true. This bit is cleared (0) by hardware reset or software reset chip (but not SCSI reset).

Bit 6 (SCSI Reset Reporting Interrupt Disable)

This bit disables the reporting of a SCSI reset. If the SCSI reset signal goes true when this bit is set, the FSC will disconnect from the SCSI bus and remain idle in the disconnected state without interrupting the host. If the bit is not set, the FSC will respond to the SCSI reset by first interrupting the host, then resetting the host if the interrupt is not serviced within 1-2 ms (depending on CLK frequency and clock conversion factor.) Refer to *Bus Initiated Reset*. This bit is cleared (0) by hardware reset or software reset chip (but not SCSI reset). The SCSI Reset Detected bit will be set if the chip detects a reset, regardless of the status of this bit.

Bit 5 (Parity Test Mode)

Setting this bit will cause the parity signal to be a duplicate of data bit 7 when unloading the FIFO to the SCSI bus. This allows parity errors to be created so that hardware and software may be tested. This bit must not be set during normal operation. Refer to Parity Checking and Generation. This bit is cleared (0) by hardware reset or software reset chip (but not SCSI reset).

Bit 4

When this bit is set, the FSC will check parity on incoming SCSI bytes during any information transfer phase except when receiving bad bytes. Detected parity errors will cause a bit to be set in the status register but will not cause an interrupt. In initiator role, bad parity will also set ATN (Attention) on the SCSI bus. When this bit is not set, parity will not be checked; the bit in the status register will not be set, and ATN will not be asserted. Refer to *Parity Checking and Generation.* This bit is cleared (0) by hardware reset or software reset chip (but not SCSI reset).

Bit 3 (Chip Test Mode Enable)

When this bit is set, the chip is placed in special test mode that enables the test register at address 0Ah. Once it has been set, the chip must be reset (hard or soft but not SCSI) before normal operation can begin. This bit should not be set during normal operation. This bit is cleared (0) by hardware reset or software reset chip (but not SCSI reset).

Bit 2-0 (My Bus ID)

This bit field is the bus ID of this device. It is the ID to which the FSC responds during bus initiated selection or reselection, and the ID that the FSC uses to arbitrate for the bus. The name of this field has changed from bus ID on the C90, but its function remains the same. This three bit field is binary encoded. These bits are cleared by a hardware reset or software reset command (but not SCSI reset).

Clock Conversion (Write Address 09)

This register must be set according to the CLK (clock) input frequency. All timings longer than 400 ns depend on this register correctly agreeing with the CLK frequency. The least significant three bits are binary encoded, and should be set to one of the following seven values:

CLK Frequency (MHz)	Clock Conversion Factor
10	010
10.01 to 15	011
15.01 to 20	100
20.01 to 25	101
25.01 to 30	110
30.01 to 35	111
35.01 to 40	000

This register must never be loaded with 1. Hardware reset or software reset chip will set the clock conversion register to 2. SCSI reset will not affect it. The upper 5 bits of this register are reserved.

Test Register (Write Address 0A)

This register is enabled by setting the special test mode bit in config-1 at address 08. After test mode has been entered, a hardware reset or software reset chip must occur before normal operation can begin.

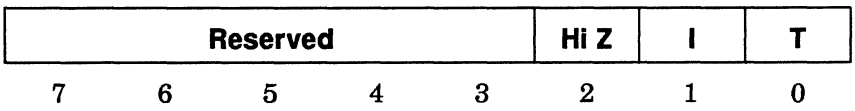


Figure 5-8 Test Register (Write address 0A

Bit 2 (All Outputs to High-impedance)

When this bit is set, all bidirectional and all output pins go to high-impedance and will not significantly load a TTL or compatible device.

Bit 1 (Initiator Mode)

When this bit is set, the FSC is artificially forced into initiator mode. Any initiator command will be accepted by the FSC. For example, a set ATN command will cause ATN to be driven on the SCSI bus even if the FSC is disconnected.

Bit 0 (Target Mode)

When this bit is set, the FSC is artificially forced into target mode. Any target command will be accepted by the FSC. For example, a DMA command will load or unload the FIFO and set the SCSI phase, data and REQ signals even if arbitration and selection have not occurred.

Configuration 2 (Config 2) (Read/Write Address 0B)

After hardware reset or software reset chip, the bits in this register are all cleared. Any bit pattern written to this register may be read back and should be identical.

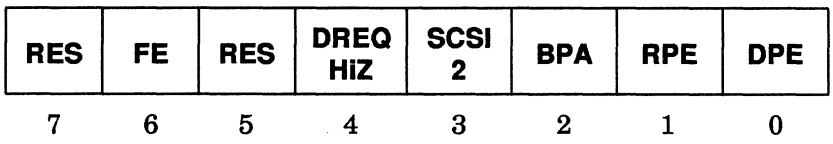


Figure 5-9 Configuration 2 (Config 2) (Read/write address 0B

Bit 7 Reserved

Bit 6 Features Enable
When set, this bit will enable the following features:

- The SCSI phase is latched at each command completion. This permits simpler software routines for stacked commands. When the Features Enable bit is not set, the phase bits (bits 2-0 in the Status register) are live indicators of the state of the SCSI phase lines.
- During differential mode operation when the SCSI phase changes from in to out, the SCSI Data In and Parity lines are delayed two or three CLKs before asserting. When the phase changes from out to in, the SCSI Data Out and Parity lines are delayed two or three CLKs before deasserting. At 40 MHz, this provides a minimum 50 ns turnoff time for the external transceivers. This will improve the SCSI Data timings in differential mode by preventing electrical bus contention between the chip and the SCSI bus transceivers when the bus changes phase.
- The Transfer Counter High register at address 0Eh is enabled, which extends the transfer counter from 16 to 24 bits.
- If other conditions are met, setting the Features Enable bit also allows the chip revision code to be read (see the Transfer Counter High register description for more information on this feature).

Bit5 Reserved

Bit 4 (DREQ High-impedance)
When this bit is set, the DREQ output (DMA Request) goes to high impedance and will not significantly load a TTL compatible device. This is useful when several devices share the DMA request line (known as wired-OR). When this bit is set, the FSC will ignore any activity on the DACK/ (DMA Acknowledge) input.

When this bit is cleared, the DREQ output will be driven to TTL high or low voltages. When this bit is cleared, DACK/ is enabled to decrement the transfer counter and load or unload the FIFO, depending or WR/ or RD/. DACK/ should not pulse true without RD/ or WR/ because the transfer counter may decrement without transferring any data. Refer to *Transfer Counter Register.*

Bit 3 (SCSI-2)
Allows the FSC to support two new features adopted in SCSI-2: the three-byte message exchange for tagged-queuing and Group 2 commands.

Tagged-Queuing
When this bit is set, and the FSC is selected with ATN (Attention), it will request either one or three message bytes depending on whether ATN remains true or goes false. If ATN is still true after the first byte has been received, the FSC may request two more message bytes before switching to command phase. If ATN goes false, it will request only one message byte then switch to command phase. When the bit is not set it will request a single message byte (as a target) when selected with ATN; and abort the selection sequence (as an initiator) if the target does not switch to command phase after one message byte has been transferred. Refer to Bus Initiated Selection.

Group 2 Commands

When the SCSI-2 bit is set, Group 2 commands are recognized as 10-byte commands. Receiving a Group 2 command with this bit set will set the valid group code bit in the status register. If the SCSI-2 bit is not set the FSC will treat Group 2 commands as reserved commands, it will request only 6 bytes in command phase and will not set the valid group code status bit.

Bit 2 (Target Bad Parity Abort)

When this bit is set, the FSC will abort a receive command or receive data sequence when the FSC detects a parity error.

Bit 1 (Register Parity Enable)

When this bit is set, parity from the host DBP pin (53C90B only) will be loaded into the FIFO when CS/ and WR/ are both true. When this bit is not set the FSC generates parity from the host data bus when CS/ and WR/ are both true and places it in the FIFO along with the data from which it was generated.

When the FSC is moving data from the FIFO to the SCSI bus, it will flag outgoing parity errors if either this bit or the DMA parity enable bit is set.

Bit 0 (DMA parity Enable)

When this bit is set, parity from the host DBP pin (53C90B only) will be loaded into the FIFO when DACK/ and WR/ are both true. When this bit is not set, the FSC generates parity from the host data bus when DACK/ and WR/ are both true and places it in the FIFO along with the data from which it was generated.

When the FSC is moving data from the FIFO to the SCSI bus, it will flag outgoing parity errors if either this bit or the register parity enable bit are set.

Configuration Register 3 (Read/Write Address 0C)

After hardware reset or a Reset Chip command the bits in this register are all cleared. Any bit pattern written to this register may be read back and should be identical.

RES	RES	RES	IDM	QTE	CDB	FSCSI	FCLK
7	6	5	4	3	2	1	0

Bits 7-5 Reserved

Bit 4 ID Message Reserved Check

This bit allows a second level of checking for the validity of an ID message. The most significant bit of an ID message byte is always checked, and must be one, or the chip interrupts. When this bit is set, bits 5-3 of the ID message are also checked and must be zero, or the chip interrupts. This check occurs in two cases: if the chip is selected with ATN true, or during reselection. If the validation check fails, the selection or reselection sequence halts and the chip generates an interrupt.

Bit 3 Queue Tag Enable

When this bit is set, the FSC can receive 3-byte messages during bus-initiated select with ATN. This feature is also enabled by setting bit 3 in the Configuration 2 register. The message bytes consist of a 1-byte Identify message and a 2-byte Queue Tag message. The middle byte is the tagged queue message itself and the last byte is the tag value (0 to 255). When this bit is set, the second byte is checked to see if it is a valid queue tagging message. If the value of the byte is not 20h, 21h, or 22h, the sequence halts and an interrupt is generated. When this bit is not set, the chip aborts the select with ATN sequence after it receives one Identify message byte, if ATN is still asserted.

Bit 2 CDB10

When this bit is set, 10-byte Group 2 commands are recognized as valid Command Descriptor Blocks (CDB). The target command sequence receives ten Group 2 command bytes and sets the Valid Group Code bit (Status register, bit 3). When this bit is not set, the target command sequence receives only six Group 2 command bytes and does not set the Valid Group Code bit. The group code defines how many bytes to request while driving command phase. This feature is also enabled or disabled by setting or clearing bit 3 in the Configuration 2 register.

Bit 1 FASTSCSI

Bit 0 FASTCLK

Bits one and zero in this register are used to inform the device that it is connected to a fast clock, and to select between Fast SCSI timings and SCSI-1 timings. Fast SCSI operation requires a 40 MHz clock. A fast clock is one with a frequency greater than 25 MHz. These bits affect the SCSI transfer rate as follows:

Bit 1	Bit 0	Min clocks/byte		SyncTransfer (MB/s)
		async	sync	
X	0	2	5	5
0	1	3	8	5
1	1	3	4	10

Transfer Counter High Register (Read Address 0E)

This register extends the transfer counter to 24 bits. Like the other transfer counter registers, this register is not affected by any reset condition. After power-up or a chip reset, and until Transfer Counter High register is loaded, the FSC part unique ID code is readable from this register. The Transfer Counter High register is only enabled when the Features Enable bit (bit 6 in the Configuration 2 register) is set. Refer to the descriptions for Transfer Counter Low and Transfer Counter Mid for more information on the transfer counter.

An 8-bit part unique ID code for the FSC is available in the Transfer Counter High register when the following conditions are true:

- After power-up or chip reset
- Before the Transfer Counter High register is loaded
- The features Enable bit (bit 6 in the Configuration 2 register) is set
- A DMA NOP command (80h) has been issued

Bits 7-3 indicate the chip family code. Bits 2-0 indicate the revision level of the chip. At power-up, the FSC family code is zero and the revision level is two.

Transfer Counter High Register (Read Address 0E)

This register extends the Transfer Count to 24 bits. Like the other transfer count registers, this register is not affected by any reset condition. This register is only enabled when the Features Enable bit (bit 6 in the Configuration 2 register) is set. Refer to the descriptions for Transfer Count Low and Transfer Count Mid for more information on the transfer count.

Parity Checking and Generation

Table 5-4 Parity Control

Register	Bit	Bit Name
Configuration 1	4	Enable Parity Checking
Configuration 1	5	Parity Test Mode
Status	5	Parity Error
Configuration 2	0	DMA Parity Enable (N/A)
Configuration 2	1	Register Parity Enable (N/A)
Configuration 2	2	Target Bad Priority Abort

The FSC has six bits that control parity generation and checking. If parity checking is disabled, the FSC does not check for parity errors. In this document, the word "detected," in conjunction with "parity error," should be understood to imply that parity checking has previously been enabled.

In target role, detected parity errors will set the parity error status bit and clear the command register. In initiator role, detected parity errors will set the parity error bit and assert ATN (Attention) prior to releasing ACK(Acknowledge). Parity errors occurring on the first few bytes after a phase change to synchronous data in are handled slightly differently in initiator mode. Refer to *Initiator Commands*.

If parity test mode is enabled, the parity bit is a duplicate of bit 7. This is true both for data flowing from the FIFO to the SCSI Data Bus (SDB) or data flowing from the FIFO to the Host Data Bus (DB).

It may pass parity between SCSI and host buses without changing it or flagging errors; or may generate parity from the data byte. Whether generated internally or externally, the parity bit is always loaded into the FIFO along with the data byte. From there on it moves through the FIFO along with the byte. The FIFO may be accessed by three busses: SCSI bus, host processor bus, or host DMA bus.

When checking parity, the FSC checks at the edge of the board. Parity errors are flagged as data comes into the FIFO from the SCSI bus, or as it leaves the FIFO on its way out to the SCSI bus.

Command Set

From the programmers point of view, DMA commands will move data between memory and the SCSI bus, non-DMA commands will move data between the FIFO and the SCSI bus. Non-DMA commands require the host processor to move data between the FIFO and memory. DMA commands require an external DMA controller to move data between the FIFO and memory. A command with bit 7 set is a DMA command. A command with bit 7 not set is a non-DMA command. DMA commands will load the transfer counter with whatever value is in the transfer count register, so the value must be correct before issuing the command.

Table 5-5 FSC Command Set

Command Register		Command Mnemonic	Interrupt
7 6 5 4	**3 2 1 0**	**Miscellaneous Group**	
X 0 0 0	0 0 0 0	NOP	No
X 0 0 0	0 0 0 1	Flush FIFO	No
X 0 0 0	0 0 1 0	Reset Chip	No
X 0 0 0	0 0 1 1	Reset SCSI Bus	No*
		Disconnected State Group	
X 1 0 0	0 0 0 0	Reselect Sequence	Yes
X 1 0 0	0 0 0 1	Select without ATN Sequence	Yes
X 1 0 0	0 0 1 0	Select with ATN Sequence	Yes
X 1 0 0	0 0 1 1	Select with ATN and Stop Sequence	Yes
X 1 0 0	0 1 0 0	Enable Selection/Reselection	No
X 1 0 0	0 1 0 1	Disable Selection/Reselection	Yes
X 1 0 0	0 1 1 0	Select with ATN3	Yes
X 1 0 0	0 1 1 1	Reselect3 Sequence	Yes
		Target State Group	
X 0 1 0	0 0 0 0	Send Message	Yes
X 0 1 0	0 0 0 1	Send Status	Yes
X 0 1 0	0 0 1 0	Send Data	Yes
X 0 1 0	0 0 1 1	Disconnect Sequence	Yes
X 0 1 0	0 1 0 0	Terminate Sequence	Yes

* The command itself does not cause an interrupt, however, external connection of the RSTO/ pin to RSTI/ pin causes an interrupt if the SCSI reset reporting is not disabled in the configuration register.

** The command itself does not cause an interrupt, however, it may allow a stalled command to finish and generate an interrupt

Table 5-5 FSC Command Set (Continued)

Command Register		Command Mnemonic	Interrupt
X 0 1 0	0 1 0 1	Target Command Complete Sequence	Yes
X 0 1 0	0 1 1 1	Disconnect	No
X 0 1 0	1 0 0 0	Receive Message	Yes
X 0 1 0	1 0 0 1	Receive Command Sequence	Yes
X 0 1 0	1 0 1 0	Receive Data	Yes
X 0 1 0	1 0 1 1	Receive Command Sequence	Yes
X 0 0 0	0 1 0 0	Target Abort DMA	No**
7 6 5 4	**3 2 1 0**	**Initiator State Group**	
X 0 0 1	0 0 0 0	Transfer Information	Yes
X 0 0 1	0 0 0 1	Initiator Command Sequence Complete	Yes
X 0 0 1	0 0 1 0	Message Accepted	Yes
X 0 0 1	1 0 0 0	Transfer Pad	Yes
X 0 0 1	1 0 1 0	Set ATN	No
X 0 0 1	1 0 1 1	Reset ATN	No

* The command itself does not cause an interrupt, however, external connection of the RSTO/ pin to RSTI/ pin causes an interrupt if the SCSI reset reporting is not disabled in the configuration register.

** The command itself does not cause an interrupt, however, it may allow a stalled command to finish and generate an interrupt

Miscellaneous Commands

Table 5-6 Miscellaneous Commands

DMA	Non-DMA	Mnemonic
80	00	No-Operation (NOP)
81	01	Flush FIFO
82	02	Reset Chip
83	03	Reset SCSI Bus

NOP

No-Operation. The 53C9X requires this command only after hardware reset or software reset chip. A DMA NOP 80h may be used to load the transfer counter with the value in transfer count register. No interrupt is generated from this command.

Flush FIFO

The Flush FIFO command initializes the FIFO to the empty condition by resetting the FIFO flags and setting the bottom byte of the FIFO to zero.

Reset Chip

This command resets all functions in the chip and returns it to a disconnected state. The command has the same effect as a hardware reset, with the exception that reset chip cannot change between single-ended mode or differential mode.

Reset SCSI Bus

This command will assert the SCSI Reset Output (RSTO) signal for 25µs, depending on CLK frequency and clock conversion factor. Refer to *Bus Initiated Reset*. This command does not cause an interrupt; however, since RSTI will be externally connected to RSTO, an interrupt will be generated unless it is disabled in the Config 1 register.

Disconnected State Commands

Table 5-7 Disconnected State Commands

DMA	Non-DMA	Mnemonic
C0	40	Reselect Sequence
C1	41	Select without ATN Sequence
C2	42	Select with ATN Sequence
C3	43	Select with ATN and Stop Sequence
C4	44	Enable Selection or reselection
C5	45	Disable Selection and Reseleciton
C6	46	Select with ATN3
C7	47	Reselect3 Sequence

If any of the Disconnected State commands are received by the FSC when it is not in the disconnected state, the command will be ignored, the command register will be cleared, and the FSC will generate an illegal command interrupt.

Reselect Sequence

This command will cause the FSC target to arbitrate for the bus then enter reselection phase when it wins arbitration. The identify message, required by SCSI protocol, must either be placed in the FIFO by the host processor before issuing the command or must be transferred by DMA, which involves setting the transfer count to one and setting up the external DMA controller. In either case, the time-out and destination ID must have previously been set. The sequence will terminate early if a reselect time-out occurs.

Select Without ATN Sequence

This command will cause the FSC initiator to arbitrate for the bus, enter selection phase when it wins, and send the CDB (Command Descriptor Block). The 6, 10, or 12 byte CDB must have either been placed in the FIFO previously by the host processor or must be transferred by DMA, which involves setting the transfer count to 6, 10 or 12 and programming the external DMA controller. In either case, the time-out and destination registers must have previously been set. This command terminates early if a reselection time-out occurs, or the target does not assert command phase, or the target removes command phase too early. If it terminates normally, a function complete/bus service interrupt will be generated.

Table 5-8 Initiator Select without ATN

Sequence Step	Interrupt Register	Interpretation
2 1 0	7 6 5 4 3 2 1 0	
0 0 0	0 0 1 0 0 0 0 0	Arbitration complete; selection time-out; disconnected
0 1 0	0 0 0 1 1 0 0 0	Arbitration and selection complete; stopped because target did not assert command phase
0 1 1	0 0 0 1 1 0 0 0	Stopped during command transfer because target prematurely changed phase
1 0 0	0 0 0 1 1 0 0 0	Select sequence complete

Table 5-9 Target Selected without ATN

Sequence Step	Interrupt Register	Interpretation
2 1 0	7 6 5 4 3 2 1 0	
0 0 0	0 0 0 0 0 0 0 1	Selected, loaded bus ID into FIFO, loaded null-byte message into FIFO
0 0 1	0 0 0 0 0 0 0 1	Stopped in command phase due to parity error; some command descriptor block bytes may not have been received; check FIFO flags
0 0 1	0 0 0 1 0 0 0 1	Same as above, initiator asserted ATN in command phase
0 1 0	0 0 0 0 0 0 0 1	Selected, received entire command descriptor block; check valid group status bit
0 1 0	0 0 0 1 0 0 0 1	Same as above, initiator asserted ATN in command phase

Select with ATN Sequence

This command will cause the FSC initiator to arbitrate for the bus, select a device with ATN true then send one message phase byte followed by 6, 10 or 12 command phase bytes. The message and command bytes must have either been placed in the FIFO by the host processor or must be transferred by DMA, which involves setting the

transfer count to 7, 11 or 13 and programming the external DMA controller. In either case, the time-out and destination ID registers must have previously been programmed. This command terminates early if: a select time-out occurs, target does not assert message phase followed by command phase, or target removes command phase early. If it completes normally, a function complete and bus service interrupt will be generated.

Table 5-10 Initiator Select with ATN

Sequence Step	Interrupt Register	Interpretation
2 1 0	7 6 5 4 3 2 1 0	
0 0 0	0 0 1 0 0 0 0 0	Arbitration complete; selection time-out; disconnected
0 0 0	0 0 0 1 1 0 0 0	Arbitration and selection complete; stopped because target did not assert message out phase; ATN still driven by FSC
0 1 0	0 0 0 1 1 0 0 0	Message out complete; sent one message byte with ATN true, then released ATN; stopped because target did not assert command phase after message byte was sent
0 1 1	0 0 0 1 1 0 0 0	Stopped during command transfer due to premature phase change; some CDB bytes may not have been sent; check FIFO flags
1 0 0	0 0 0 1 1 0 0 0	Selection with ATN sequence complete

Table 5-11 Target Selected with ATN SCSI-2 Bit Not Set

Sequence Step	Interrupt Register	Interpretation
2 1 0	7 6 5 4 3 2 1 0	
0 0 0	0 0 0 0 0 0 1 0	Selected with ATN, stored bus ID and one message byte; stopped due either to parity error or invalid ID message
0 0 0	0 0 0 1 0 0 1 0	Selected with ATN, stored bus ID and one message byte; stopped because ATN remained true after first message byte
0 0 1	0 0 0 0 0 0 1 0	Stopped in command phase due to parity error; some CDB bytes not received; check valid group code bit and FIFO flags
0 0 1	0 0 0 1 0 0 1 0	Stopped in command phase; parity error and ATN true
0 1 0	0 0 0 0 0 0 1 0	Selection complete; received one message byte and the entire command descriptor block
0 1 0	0 0 0 1 0 0 1 0	Same as above, initiator asserted ATN during command phase

Table 5-12 Target Selected with ATN SCSI-2 Bit Set

Sequence Step	Interrupt Register	Interpretation
2 1 0	7 6 5 4 3 2 1 0	
0 0 0	0 0 0 0 0 0 1 0	Selected with ATN, stored bus ID and one message byte; stopped due either to parity error or invalid ID message
1 0 0	0 0 0 0 0 0 1 0	Parity error during second or third message byte
1 0 0	0 0 0 1 0 0 1 0	ATN remained true after third message byte
1 0 1	0 0 0 0 0 0 1 0	Received three message bytes; then stopped in command phase due to parity error; some CDB bytes not received; check valid group code bit and FIFO flags
1 0 1	0 0 0 1 0 0 1 0	Stopped in command phase; parity error and ATN true
1 1 0	0 0 0 0 0 0 1 0	Selection complete; received 3 message bytes and the entire command descriptor block

Select with ATN and Stop

This command should be used in place of the one above when multiple message phase bytes are to be sent. The command will select a target with ATN asserted, send one message phase byte, and generate bus service and function complete interrupts, and stop.

Table 5-13 Initiator Select with ATN and Stop

Sequence Step	Interrupt Register	Interpretation
2 1 0	7 6 5 4 3 2 1 0	
0 0 0	0 0 1 0 0 0 0 0	Arbitration complete; selection time-out; disconnected
0 0 0	0 0 0 1 1 0 0 0	Arbitration and selection complete; stopped because target did not assert message out phase; ATN still asserted by FSC
0 0 1	0 0 0 1 1 0 0 0	Message out complete; sent one message byte; ATN on

Enable Selection/Reselection

After receiving this command, the FSC will respond to bus initiated selection or reselection. A command that causes the FSC to select or reselect will cancel this command. The command must be re-issued within 250 ms after the FSC disconnects to preserve ANSI recommended timings. If DMA is enabled, incoming Command Descriptor Block will be placed in memory. If DMA is not enabled, incoming information will remain in the FIFO.

Disable Selection/Reselection

This command disables an earlier Enable Selection/Reselection command. If bus initiated selection or reselection had not yet begun when this command is received by the FSC, it will generate a function complete interrupt. If bus initiated selection or reselection had already begun, this command (and every other command) will be ignored. Refer to *Bus Initiated Selection* and *Bus Initiated Reselection.*

Select with ATN3 Sequence

This command is similar to the select with ATN command, but sends three message bytes instead of one. It will cause the FSC initiator to arbitrate for the bus, select a device with ATN true, send three message phase bytes, deassert ATN, then send 6, 10, or 12 command phase bytes. The message and command bytes must have either been placed in the FIFO by the host processor or must be transferred by DMA; this involves setting the transfer count to 7, 11, or 13 and programming the external DMA controller. In either case, the time-out and destination ID registers must have previously been programmed. This command terminates early if: a select time-out occurs, target does not assert message phase followed by command phase or target removes command phase early. If it completes normally, a function complete and bus service interrupt will be generated.

Table 5-14 Initiator Select with ATN3

Sequence Step	Interrupt Register	Interpretation
2 1 0	7 6 5 4 3 2 1 0	
0 0 0	0 0 1 0 0 0 0 0	Arbitration complete; selection time-out; disconnected
0 0 0	0 0 0 1 1 0 0 0	Arbitration and selection complete; stopped because target did not assert message out phase; ATN still driven by FSC
0 1 0	0 0 0 1 1 0 0 0	Sent 1, 2, or 3 message bytes; stopped because target prematurely changed from message out phase or did not assert command phase after third message byte; ATN released only if third message byte was sent
0 1 1	0 0 0 1 1 0 0 0	Stopped during command transfer due to premature phase change; Some CDB bytes may not have been sent; check FIFO flags
1 0 0	0 0 0 1 1 0 0 0	Selection with ATN3 sequence complete

Initiator Commands

Table 5-15 Initiator Commands

DMA	Non-DMA	Mnemonic
90	10	Transfer Information
91	11	Initiator Command Complete Sequence
92	12	Message Accepted
98	18	Transfer Pad
9A	1A	Set ATN (Attention)
—	1B	Reset ATN (Attention

If the FSC is not in initiator state when it receives any of these commands, the command will be ignored, an illegal command interrupt will be generated, and the command register will be cleared. Refer to *Command Register.*

If BSY goes false while the FSC is connected as an initiator, it will generate a disconnected interrupt. The interrupt output will occur 1.5 to 3.5 CLK cycles after BSY goes false.

When the FSC receives the last byte of a message in phase, it will leave ACK (Acknowledge) asserted on the bus to prevent the target from sending any more bytes until the initiator decides to accept or reject the message. For non-DMA commands, every byte is presumed to be the last byte. For DMA commands, the transfer counter signals the last byte.

If parity checking is enabled and the FSC detects a parity error while in initiator mode, it will automatically assert ATN prior to deasserting ACK for the byte which has the error. The one exception is after a phase change to synchronous data in, described below.

If the synchronous offset register is non-zero (synchronous) and the phase changes to data in, the DMA interface is immediately disabled and the reporting of a parity error during data in phase is delayed. The phase, change to data, in will: latch the FIFO flags to indicate how many bytes were in the FIFO (these bytes will be lost), clear the FIFO, load the FIFO with the first data in byte, generate an interrupt, and continue to load the FIFO with incoming data in bytes as long as the target sends them, but not more than the specified offset. To continue receiving data in bytes, the host processor would normally issue the transfer information command to re-enable the DMA interface. If parity checking is enabled, and a parity error occurred on a previous input phase (message in or status), then the parity error flag will be set in the status register, and ATN (Attention) will be set on the SCSI bus. If a parity error occurred during the data in phase, the parity bit will not be set, nor will ATN be asserted until after the FSC receives the subsequent transfer information command.

Transfer Information

This command can be used to send or receive any information phase bytes, but is most often used for data transfer. For synchronous transfer, DMA must be used. The FSC will continue to transfer information until one of the following terminating events occurs:

- Transfer is complete. This successful completion will create a bus service interrupt. For a DMA transfer info, the transfer is complete when the transfer count decrements to zero and the FIFO is empty and the target asserts REQ (Request) for the next byte. For non-DMA transfer info in which the FSC is sending bytes to the SCSI bus, transfer is complete when the FIFO empties and the target asserts REQ for the next byte. For non-DMA transfer info in which the FSC is receiving bytes from the SCSI bus, transfer is complete after one byte is received and the target asserts REQ for the next byte. Thus, non-DMA transfer info commands will generate an interrupt for every byte received.

- If the phase is message out, the FSC removes ATN prior to asserting ACK for the last byte of the message. For non-DMA, the FIFO flags indicate the last byte. For DMA, the transfer counter indicates the last byte.

- Target changes phase. The FSC clears the command register and generates a bus service interrupt, after the target asserts REQ for the next byte.

- Target releases BSY (Busy). The FSC generates a disconnected interrupt.

- The FSC receives the last byte of a message in phase. (For non-DMA every byte is assumed to be the last byte. For DMA, the transfer counter signals the last byte.) The FSC leaves ACK(Acknowledge) asserted and generates a function complete interrupt.

All message in and status phase transfers are handled one byte at a time. If DMA is enabled, the next byte will not be received until the current byte has been written to buffer memory and the FIFO is empty. If DMA is not enabled, each byte will create an interrupt.

Initiator Command Complete Sequence

This command will cause the FSC to receive a status byte followed by a message byte. It terminates early if the target does not assert message in phase, or if the target disconnects. After receiving the message byte, the FSC leaves ACK asserted on the bus to allow the initiator to assert ATN if the message is unacceptable.

Message Accepted

This command releases the ACK signal on the SCSI bus. Any of the commands that receive bytes during message phase will leave ACK asserted after receiving the last message byte. To accept the message, issue this command. To reject the message, set ATN, then issue this command.

Transfer Pad

Transfer Pad is usually an error recovery technique. It is useful when a target requests more bytes than an initiator has to send, or when an initiator must receive and discard a number of bytes from a target.

When transmitting to the SCSI bus, Transfer Pad will fill the FIFO with null bytes and send them to the SCSI bus. When receiving from the SCSI bus, Transfer Pad will receive bytes, place them on the top of the FIFO and discard them from the bottom of the FIFO.

When sending pad bytes to the SCSI bus, DMA must be enabled. No DMA requests are actually made, but the FSC uses the transfer counter to end the transfer.

The command terminates under the same conditions as the transfer info command, except that the FSC does not leave ACK asserted on the last byte of a message in phase. If the command terminates early (due to phase change or disconnect) the FIFO may contain bad bytes.

Set ATN

This command asserts attention on the SCSI bus. No interrupt is generated from this command. ATN stays asserted until the last byte of a message out phase. DMA commands use the transfer counter to indicate the last byte. For non-DMA commands, every byte is assumed to be the last byte. ATN will also be released if the target prematurely disconnects.

Reset ATN

This command causes ATN to be released. It does not cause an interrupt.

This command must not be used when connected to a device supporting the Common Command Set (CCS). The FSC obeys CCS protocol by releasing ATN on the last byte of a message out phase. The Reset ATN command is provided for older devices that do not respond properly to the ATN condition.

Target Commands

Table 5-16 Target Commands

DMA	Non-DMA	Mnemonic
A0	20	Send Message
A1	21	Send Status
A2	22	Send Data
A3	23	Disconnect Sequence
A4	24	Terminate Sequence
A5	25	Target Command Complete Sequence
A7	27	Disconnect
A8	28	Receive Message Sequence
A9	29	Receive Command
AA	2A	Receive Data
AB	2B	Receive Command Sequence
84	04	Target Abort DMA

If the FSC receives any of these commands when it is not in target state, it will ignore the command, clear the command register, and generate an illegal command interrupt. Refer to *Command Register*.

Normal completion of these commands will cause a function complete interrupt. If ATN is asserted, the bus service bit will be set in the status register. If the FSC was idle when ATN was asserted, a bus service interrupt will be generated, the function complete bit will be zero, and the command register will be cleared.

Send Message

This command will cause the FSC to assert message in phase and send bytes until the FIFO is empty and the transfer counter is zero (if DMA).

Send Status

This command will cause the FSC to assert status phase and send bytes until the FIFO is empty and the transfer counter is zero (if DMA).

Send Data

This command will cause the FSC to assert data in phase and send bytes until the FIFO is empty and the transfer counter is zero (if DMA).

Disconnect Sequence

This command will cause the FSC to assert message in phase, send two bytes, then disconnect from the SCSI bus. Normally, the first byte will be a save data pointers message and the second will be a disconnect message. If ATN is asserted by the initiator, the bus service and function complete bits will be set; an interrupt will be generated, but the FSC will not disconnect.

Terminate Sequence

This command will cause the FSC to first assert status phase, send one byte; then assert message in phase and send one more byte. If ATN is asserted by the initiator, the bus service and function complete bits will be set, an interrupt will be generated, but the FSC will not disconnect.

Table 5-17 Target Terminate Sequence

Sequence Step	Interrupt Register	Interpretation
2 1 0	7 6 5 4 3 2 1 0	
0 0 0	0 0 0 1 1 0 0 0	Sent status byte; stopped because initiator set ATN
0 0 1	0 0 0 1 1 0 0 0	Sent status and message bytes; stopped because initiator set ATN
0 1 0	0 0 1 0 1 0 0 0	Terminate sequence complete; disconnected; bus is free

Target Command Complete Sequence

This command is similar to terminate sequence, but is used for linked commands. It will cause the FSC to first assert status phase, send one byte, then assert message in phase and send one more byte. The message byte will normally be a command complete message. If ATN is asserted by the initiator, the bus service and function complete bits will be set; an interrupt will be generated, but the FSC will not disconnect.

Table 5-18 Target Command Complete Sequence

Sequence Step	Interrupt Register	Interpretation
2 1 0	7 6 5 4 3 2 1 0	
0 0 0	0 0 0 1 1 0 0 0	Sent status byte; stopped because initiator set ATN
0 0 1	0 0 0 1 1 0 0 0	Sent status and message bytes; stopped because initiator set ATN
0 1 0	0 0 0 0 1 0 0 0	Command complete sequence complete

Disconnect

This command causes the FSC to release all SCSI bus signals except RSTO. The FSC returns to the disconnected state without generating an interrupt.

Table 5-19 Target Disconnect Sequence

Sequence Step	Interrupt Register	Interpretation
2 1 0	7 6 5 4 3 2 1 0	
0 0 0	0 0 0 1 1 0 0 0	Sent one message byte; stopped because initiator set ATN
0 0 1	0 0 0 1 1 0 0 0	Sent two message bytes; stopped because initiator set ATN
0 1 0	0 0 1 0 1 0 0 0	Disconnect sequence complete; disconnected; bus is free

Receive Message Sequence

This command allows the target to request message bytes from the initiator SCSI device. The SCSI bus phase lines are set to the Message Out phase, and the target receives bytes from the initiator through the SCSI bus. A function complete interrupt is generated upon command completion; if ATN is still asserted, the bus service interrupt is set and the Command Register is cleared. If a parity error is detected, the 53C9X receives message bytes and discards them until ATN is false, a function complete interrupt is generated, and the Command Register is cleared.

Receive Command

This command will cause the FSC to assert command phase and receive bytes from the initiator. For non-DMA Receive command, only one byte per interrupt may be received. DMA Receive command will interrupt after the transfer counter decrements to zero.

Table 5-20 Target Receive Command

Sequence Step	Interrupt Register	Interpretation
2 1 0	7 6 5 4 3 2 1 0	
0 0 1	0 0 0 0 1 0 0 0	Stopped during command transfer due to parity error; check FIFO flags
0 0 1	0 0 0 1 1 0 0 0	Stopped during command transfer due to parity error; ATN asserted by initiator
0 1 0	0 0 0 0 1 0 0 0	Received entire command descriptor block
0 1 0	0 0 0 1 1 0 0 0	Received entire CDB, initiator asserted ATN

Receive Data

This command will cause the FSC to assert data out phase and receive bytes from the initiator. For non-DMA Receive Data, only one byte per interrupt may be received. DMA Receive Data will interrupt after the transfer counter decrements to zero.

Receive Command Sequence

This command will cause the FSC to assert command phase and receive a number of bytes, which will vary according to the group code field of the first byte. If the SCSI-2 bit is set in the Config 2 register, Group 2 commands will be recognized as 10-byte commands. If the SCSI-2 bit is cleared, Group 2 commands will be recognized as reserved commands. Groups 3 and 4 are always reserved. The FSC will request 6 bytes for reserved commands, 6 bytes for Group 6 vendor unique commands, and 10 bytes for Group 7 vendor unique commands.

Target Stop DMA

This command allows the host processor to stop a DMA data transfer command. The FSC must be in target state when this command falls to the bottom of the command FIFO or an illegal command interrupt will be generated. Target stop DMA may only be used when all of the following are true.

1. Either a Target Send Data or Target Receive Data command are currently executing.
2. The DMA controller has stopped.
3. The FSC is in steady state, that is:
   a. Send data–the FIFO is empty.
   b. Receive asynchronous data–the FIFO is full or the transfer counter is zero.
   c. Receive sync data–the transfer counter is zero or the synchronous offset max bit (read register 06, bit 3) is not set.

Upon receiving this command, the FSC will reset the DMA interface (release DREQ) then terminate the current command. It will not generate its interrupt until the rest of the completion criteria are met.

1. Send asynchronous data–completes immediately.
2. Send synchronous data–completes when the offset counter is zero.
3. Receive asynchronous data–completes immediately. There will be data in the FIFO which should be removed by the host processor.
4. Receive synchronous data– completes when all outstanding SCSI ACKs have been received. The offset counter is separate from the transfer counter. There will be data in the FIFO which should be removed by the host processor.

AC Electrical Characteristics

This document is intended for use with the SCSI controller embedded in the NCR89C100 Master I/O device. All relevant timing information can be found in the section titled "NCR89C100 Master I/O.

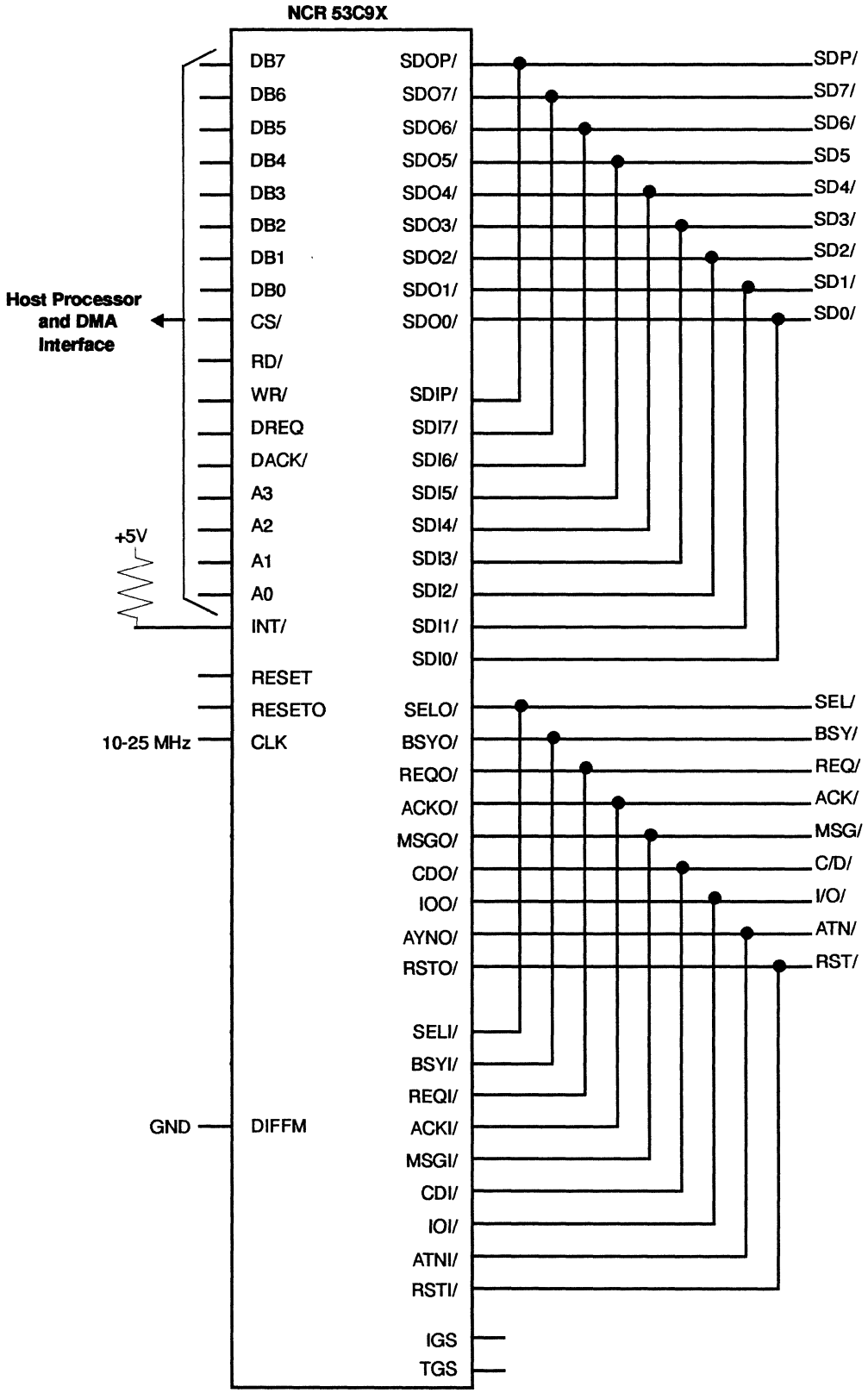


Figure 5-10 Single-Ended Mode Without External Drivers

NCR89C105 Chip Specification
Contents

List of Figures

List of Tables

NCR89C105
Chip Specification

Overview

The NCR89C105 device is designed for use in a low-cost single-processor system with an SBus interface. This chip is meant to replace the slave portion of the desktop I/O hardware with a highly integrated, low cost and low power part. The 89C105 integrates two dual serial controllers, a high-speed floppy controller, uniprocessor interrupt, reset, and counter/timer circuitry, power down control, and an external byte-wide expansion bus in a single 160 PQFP package.

Chip-Level Functional Block Diagram

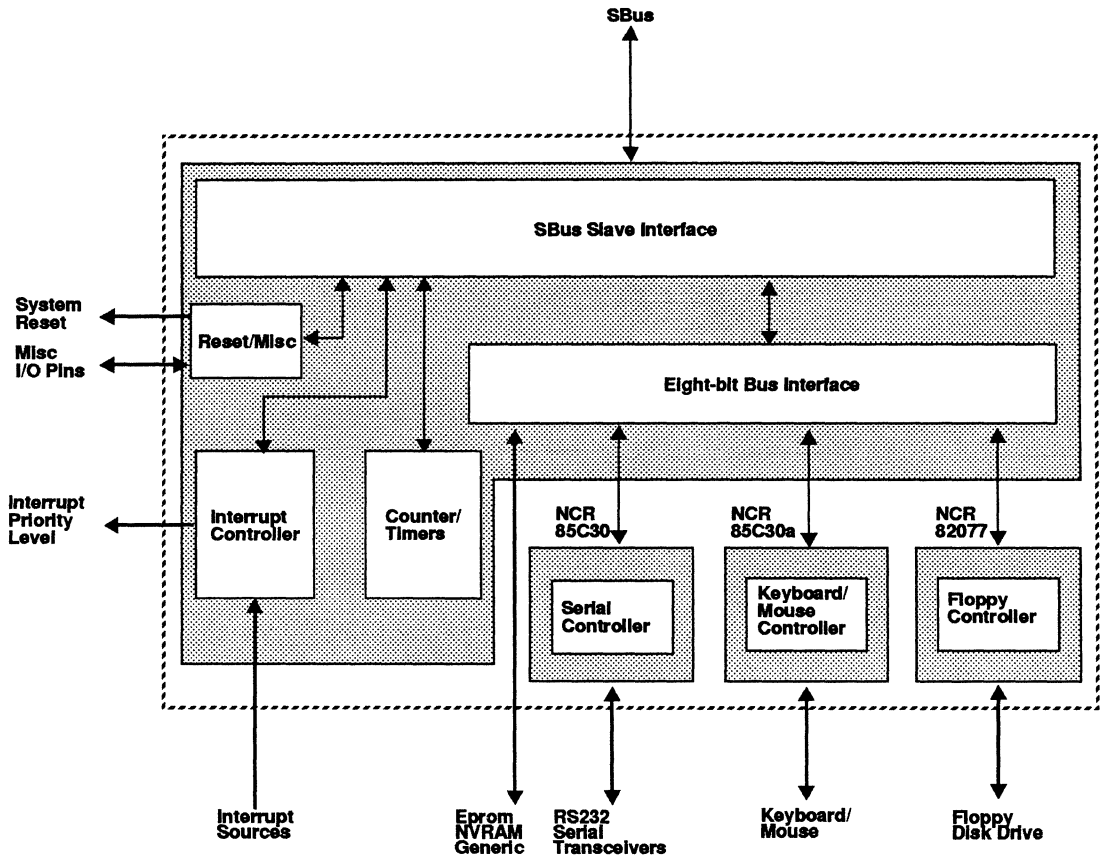


Figure 6-1 89C105 Block Diagram

Features

The following features are incorporated in the 89C105:

- Four NCR85C30 serial ports for keyboard/mouse and general purpose use, compatible with the AMD AM85C30 rev. C, without extended features. The TTYA/B serial ports are fully synchronous, while the keyboard/mouse ports are asynchronous only. All ports support data rates up to 38.4 Kb/s.
- NCR82077 floppy disk interface compatible with the Intel® 82077AA-1 single-chip floppy controller, supporting up to 1 Mbit/sec transfer rate.
- Expansion bus that supports the following byte-wide peripherals:
  - EPROM
  - TOD/NVRAM (Mostek 48T02 or 48T08)
  - Generic 8-bit device
- Auxiliary I/O registers (used for led, floppy, and system powerdown)
- Interrupt Controller for single-processor SBus system
- System reset control
- Counter/timers for single-processor SBus system
- JTAG internal and boundary scan for improved fault coverage and board testability

Intended Applications

The 89C105 is intended for uniprocessor SBus machines. It can work with either the Texas Instruments microSPARC processor or the Texas Instruments SuperSPARC processor and will also work in any SBus-based system.

Related Products

The 89C105 is designed to share a single SBus slot with the NCR89C100.

Pinout Information

This section includes the pinout map and tables which summarize the 89C105 pinout information in the following formats:

- Pinout by function
- Pinout by pin order on package
- JTAG boundary information

Pinout Map

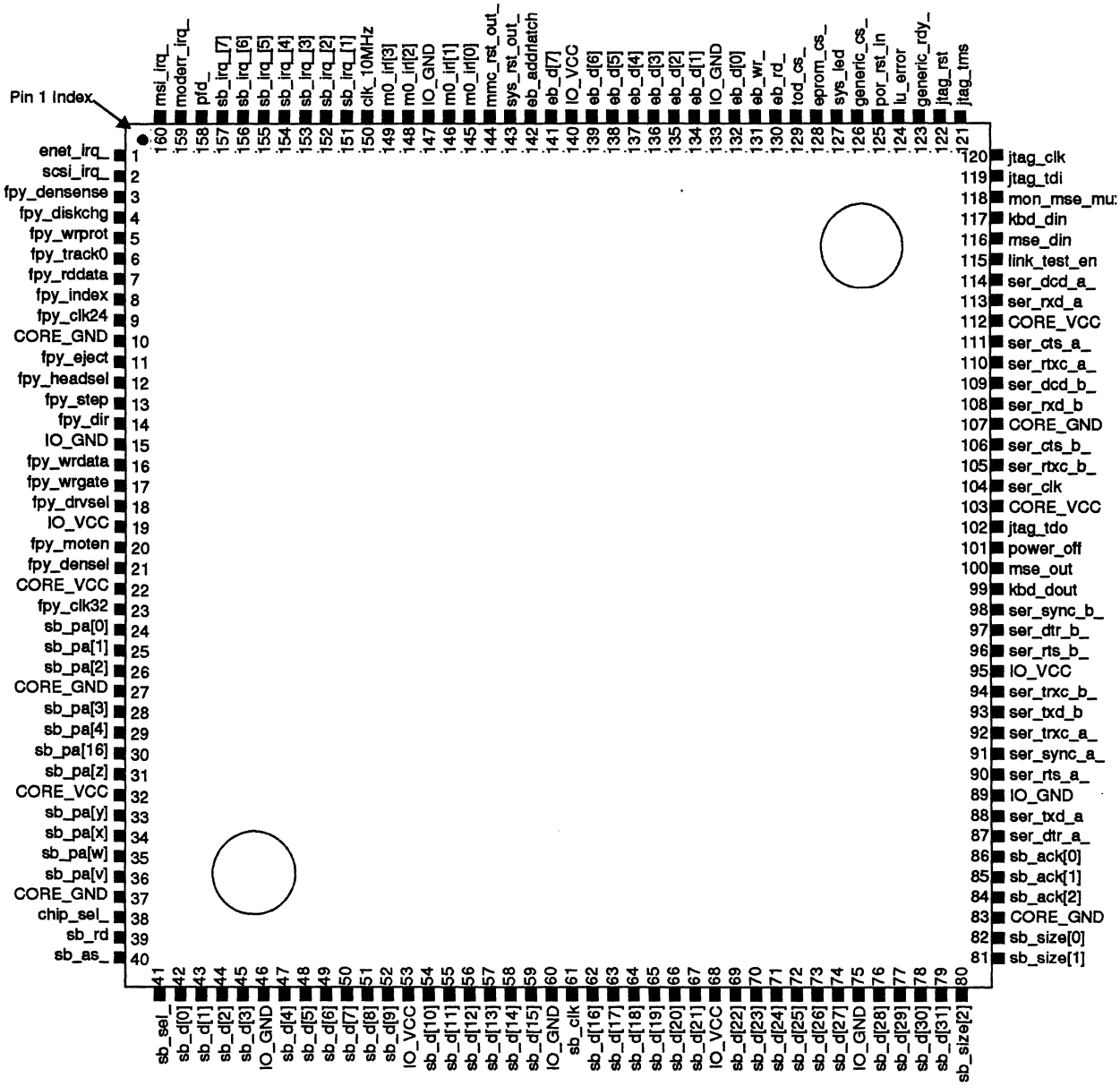


Figure 6-2 Pinout Map

Pinout Tables

Tables 6-4 and 6-5 present the 89C105 pinouts by function and by pin number sequence, respectively.

The pin type information in the Type column of Tables 6-4 and 6-5 is identified by fields which contain the mnemonic values shown in Tables 6-1 and 6-2.

Table 6-1 Valid Pin Mnemonics

Field	A	B	C	D	E	F	G	H	I
Value(s)	BS	IN	N	PD	2	L	U	25	T
	NCR	IO			4		D	100	H
		IOP			6				I
		ION			8				S16
		O			12				S18
		OT			16				S38
		IP							

The values represented by the mnemonics in each of these fields is as follows:

Table 6-2 Mnemonic Descriptions

Field	Mnemonic	Description
A	BS	Boundary Scan
	NCR	NCR type
B	IN	Input
	IO	Bidirectional
	IOP	Bidirectional with pull-up/pull-down
	ION	Bidirectional open drain
	O	Output
	OT	Tristate output
	IP	Input with pull-up/pull-down
C	N	Open drain
D	PD	IO pad
E	*integer*	Pad drain in mA as indicated
F	L	Slew rate limited output
G	U	Pull-up
	D	Pull-down
H	*integer*	Pull-up/pull-down value in uA as indicated

Table 6-2 Mnemonic Descriptions (Continued)

Field	Mnemonic	Description
I	T	TTL input receiver
	H	High drive TTL input receiver
	I	Inverting TTL input receiver
	S16	ds1216 Schmitt input receiver
	S18	ds1218 Schmitt input receiver
	S38	ds1238 Schmitt input receiver

For example, the pin type identification bsinpds18 means that the pin type is a boundary scan version of an input pad with a ds1218 Schmitt input receiver.
The Direction column in Tables 6-4 and 6-5 is used to identify the pin direction during system operation, using the mnemonics: shown in Table 6-3.

Table 6-3 Direction Mnemonic Descriptions

Mnemonic	Description
I	Input
O	Output
B	Bidirectional
T	Tristate
—	Not applicable

Note that the pad type given in the following tables may not correspond exactly to the functional direction listed for the pin (input, output, bidirectional, or tristate) due to either of the following reasons:

- The pin is used differently in a test mode; for instance, using an input as an output during test will require use of a bidirectional pad instead of an input.
- An equivalent output-only pad was not available. This applies specifically to the SBus outputs, which all use a custom 12 mA pad which was only available as a bidirectional pad.

Pinout by Function

Table 6-4 Pinout by Function

Name	Pin	Direction	Type	Description
SBus Interface: 54 pins				
sb_d[31]	79	B	BSIOPD12S18	SBus Data Bus (MSB)
sb_d[30]	78	B	BSIOPD12S18	SBus Data Bus
sb_d[29]	77	B	BSIOPD12S18	SBus Data Bus
sb_d[28]	76	B	BSIOPD12S18	SBus Data Bus
sb_d[27]	74	B	BSIOPD12S18	SBus Data Bus
sb_d[26]	73	B	BSIOPD12S18	SBus Data Bus
sb_d[25]	72	B	BSIOPD12S18	SBus Data Bus
sb_d[24]	71	B	BSIOPD12S18	SBus Data Bus
sb_d[23]	70	B	BSIOPD12S18	SBus Data Bus
sb_d[22]	69	B	BSIOPD12S18	SBus Data Bus
sb_d[21]	67	B	BSIOPD12S18	SBus Data Bus
sb_d[20]	66	B	BSIOPD12S18	SBus Data Bus
sb_d[19]	65	B	BSIOPD12S18	SBus Data Bus
sb_d[18]	64	B	BSIOPD12S18	SBus Data Bus
sb_d[17]	63	B	BSIOPD12S18	SBus Data Bus
sb_d[16]	62	B	BSIOPD12S18	SBus Data Bus
sb_d[15]	59	B	BSIOPD12S18	SBus Data Bus
sb_d[14]	58	B	BSIOPD12S18	SBus Data Bus
sb_d[13]	57	B	BSIOPD12S18	SBus Data Bus
sb_d[12]	56	B	BSIOPD12S18	SBus Data Bus
sb_d[11]	55	B	BSIOPD12S18	SBus Data Bus
sb_d[10]	54	B	BSIOPD12S18	SBus Data Bus
sb_d[9]	52	B	BSIOPD12S18	SBus Data Bus
sb_d[8]	51	B	BSIOPD12S18	SBus Data Bus
sb_d[7]	50	B	BSIOPD12S18	SBus Data Bus
sb_d[6]	49	B	BSIOPD12S18	SBus Data Bus
sb_d[5]	48	B	BSIOPD12S18	SBus Data Bus
sb_d[4]	47	B	BSIOPD12S18	SBus Data Bus
sb_d[3]	45	B	BSIOPD12S18	SBus Data Bus
sb_d[2]	44	B	BSIOPD12S18	SBus Data Bus
sb_d[1]	43	B	BSIOPD12S18	SBus Data Bus
sb_d[0]	42	B	BSIOPD12S18	SBus Data Bus (LSB)

Table 6-4 Pinout by Function (Continued)

Name	Pin	Direction	Type	Description
sb_ack[2]	84	T	BSIOPD12T	SBus Acknowledge
sb_ack[1]	85	T	BSIOPD12T	SBus Acknowledge
sb_ack[0]	86	T	BSIOPD12T	SBus Acknowledge
sb_clk	61	I	BSCLOCK	SBus Clock Input
sb_rd	39	I	BSINPDS18	SBus Read/Write
sb_sel_	41	I	BSINPDS18	SBus Select
sb_size[2]	80	I	BSINPDS18	SBus Transfer Size
sb_size[1]	81	I	BSINPDS18	SBus Transfer Size
sb_size[0]	82	I	BSINPDS18	SBus Transfer Size
sb_as_	40	I	BSINPDS18	SBus Address Strobe (addr is valid)
chip_sel_[1]	38	I	BSINPDS18	High order physical address bits (for slave decodes)
sb_pa[v][2]	36	I	BSINPDS18	High order physical address bits (for slave decodes)
sb_pa[w][2]	35	I	BSINPDS18	High order physical address bits (for slave decodes)
sb_pa[x][2]	34	I	BSINPDS18	High order physical address bits (for slave decodes)
sb_pa[y][2]	33	I	BSINPDS18	High order physical address bits (for slave decodes)
sb_pa[z][2]	31	I	BSINPDS18	High order physical address bits (for slave decodes)
sb_pa[16]	30	I	BSINPDS18	PA[16] (for system/user selection in int/tmr)
sb_pa[4]	29	I	BSINPDS18	Low order physical address bits
sb_pa[3]	28	I	BSIOPD4S18	Low order physical address bits
sb_pa[2]	26	I	BSIOPD4S18	Low order physical address bits
sb_pa[1]	25	I	BSINPDS18	Low order physical address bits
sb_pa[0]	24	I	BSINPDS18	Low order physical address bits
Floppy Interface: 17 pins				
fpy_clk32	23	I	BSINPDS38	32 MHz clock for floppy DDS

Table 6-4 Pinout by Function (Continued)

Name	Pin	Direction	Type	Description
fpy_clk24	9	I	BSINPDS38	24 MHz clock for floppy ASF
fpy_densense	3	I	BSINPDS18	Density sense input (auxio1 register bit)
fpy_diskchg	4	I	BSINPDS18	Disk change
fpy_wrprot	5	I	BSINPDS18	Write protect
fpy_track0	6	I	BSINPDS18	Track 0 indicator
fpy_rddata	7	I	BSINPDS18	Read data
fpy_index	8	I	BSINPDS18	Track index
fpy_eject	11	O	BSOTPD16L	Floppy eject (actually ME[3] of 82077 ASF)
fpy_headsel	12	O	BSOTPD16L	Head select
fpy_step	13	O	BSOTPD16L	Drive step pulse
fpy_dir	14	O	BSOTPD16L	Head step direction
fpy_wrdata	16	O	BSOTPD16L	Write data
fpy_wrgate	17	O	BSOTPD16L	Write enable
fpy_drvsel	18	O	BSOTPD16L	Floppy drive select (DS[0] of 82077 ASF)
fpy_moten	20	O	BSOTPD16L	Floppy motor enable (ME[0] of 82077 ASF)
fpy_densel	21	O	BSOTPD16L	Density select (ME[2] or DENSEL of 82077 ASF)
Serial Interface:19 pins				
ser_clk	104	I	BSINPDS38	19.66 MHz serial clock
ser_rtxc_a_	110	I	BSINPDS18	Receive/transmit clock A
ser_cts_a_	111	I	BSINPDS18	Clear to send A
ser_rxd_a	113	I	BSINPDS18	Receive data A
ser_dcd_a_	114	I	BSINPDS18	Data carrier detect A
ser_dtr_a_	87	O	BSOTPD4	Data terminal ready A
ser_txd_a	88	O	BSOTPD4	Transmit data A
ser_rts_a_	90	O	BSOTPD4	Request to send A
ser_sync_a_	91	B	BSIOPD4S18	Sync IO, A
ser_trxc_a_	92	B	BSIOPD4S18	Transmit clock A
ser_rtxc_b_	105	I	BSINPDS18	Receive/transmit clock B
ser_cts_b_	106	I	BSINPDS18	Clear to send B
ser_rxd_b	108	I	BSINPDS18	Receive data B

Table 6-4 Pinout by Function (Continued)

Name	Pin	Direction	Type	Description
ser_dcd_b_	109	I	BSINPDS18	Data carrier detect B
ser_dtr_b_	97	O	BSOTPD4	Data terminal ready B
ser_txd_b	93	O	BSOTPD4	Transmit data B
ser_rts_b_	96	O	BSOTPD4	Request to send B
ser_sync_b_	98	B	BSIOPD4S18	Sync IO, B
ser_trxc_b_	94	B	BSIOPD4S18	Transmit clock B
Keyboard/Mouse Interface: 4 pins				
kbd_din	117	I	BSINPDS18	Keyboard data in
kbd_dout	99	O	BSOTPD4	Keyboard data out
mse_din	116	I	BSINPDS18	Mouse data in
mse_out	100	O	BSIOPD4	Mouse data out
EBus Interface: 15 pins				
eb_d[7]	141	B	BSIOPD8S18	EBus data
eb_d[6]	139	B	BSIOPD8S18	EBus data
eb_d[5]	138	B	BSIOPD8S18	EBus data
eb_d[4]	137	B	BSIOPD8S18	EBus data
eb_d[3]	136	B	BSIOPD8S18	EBus data
eb_d[2]	135	B	BSIOPD8S18	EBus data
eb_d[1]	134	B	BSIOPD8S18	EBus data
eb_d[0]	132	B	BSIOPD8S18	EBus data
eb_addrlatch	142	O	BSOTPD4	EBus address latch
eb_rd_	130	O	BSIOPD8T	EBus read
eb_wr_	131	O	BSIOPD8T	Ebus write
tod_cs_	129	O	BSOTPD4	TOD chip select
eprom_cs_	128	O	BSOTPD4	EPROM chip select
generic_cs_	126	O	BSIOPD4T	Generic port chip select
generic_rdy_	123	I	BSIPPDU25S18	Generic port ready (25 uA pull-up)
Interrupt Signals: 16 pins				
m0_irl[3]	149	O	BSOTPD4	Module 0 encoded interrupt level
m0_irl[2]	148	O	BSOTPD4	Module 0 encoded interrupt level
m0_irl[1]	146	O	BSOTPD4	Module 0 encoded interrupt level
m0_irl[0]	145	O	BSOTPD4	Module 0 encoded interrupt level

Table 6-4 Pinout by Function (Continued)

Name	Pin	Direction	Type	Description
sb_irq_[7]	157	I	BSINPDS18	SBus interrupt requests
sb_irq_[6]	156	I	BSINPDS18	SBus interrupt requests
sb_irq_[5]	155	I	BSINPDS18	SBus interrupt requests
sb_irq_[4]	154	I	BSINPDS18	SBus interrupt requests
sb_irq_[3]	153	I	BSINPDS18	SBus interrupt requests
sb_irq_[2]	152	I	BSINPDS18	SBus interrupt requests
sb_irq_[1]	151	I	BSINPDS18	SBus interrupt requests
enet_irq_	1	I	BSINPDS18	Ethernet interrupt request
scsi_irq_	2	I	BSINPDS18	SCSI interrupt request
pfd_	158	I	BSINPDS18	Power fail detect (level 15 interrupt)
msi_irq_	160	I	BSINPDS18	MSI interrupt (Super-SPARC mode)
moderr_irq_	159	I	BSINPDS18	Processor level 15 interrupt (async error)
Reset Signals: 3 pins				
por_rst_in_	125	I	BSINPDS38	Powerup reset input
sys_rst_out_	143	O	BSIOPD12T	System (SBus) reset output
mmc_rst_out_	144	O	BSOTPD4	MMC reset output (SuperSPARC mode)
Miscellaneous System Signals: 6 pins				
clk_10mhz	150	I	BCCLOCKS38	10 MHz clock for counter/ timer block
mon_mse_mux	118	O	BSOTPD4T	Monitor/mouse mux select
power_off	101	O	BSIOPD4T	Power off output (to power supply)
link_test_en	115	O	BSIOPD4T	T7213 link test enable
sys_led	127	O	BSOTPD4	System LED output
iu_error_	124	I	BSIOPD4S18	Processor watchdog reset/Video interrupt (SuperSPARC mode)
Test: 5 pins				
jtag_tdo	102	O	NCROTPD4	JTAG test data output
jtag_tdi	119	I	NCRIPPD	JTAG test data input (100 uA pull-up)
jtag_clk	120	I	NCRINPD	JTAG clock
jtag_tms	121	I	NCRIPPD	JTAG test mode select (100 uA pull-up)

Table 6-4 Pinout by Function (Continued)

Name	Pin	Direction	Type	Description
jtag_rst	122	I	NCRIPPD	JTAG reset (100 uA pull-up)
Power, Ground: 21pins				
VDD (pad)	19	—	Power	Power Connection
VDD (pad)	53	—	Power	Power Connection
VDD (pad)	68	—	Power	Power Connection
VDD (pad)	95	—	Power	Power Connection
VDD (pad)	140	—	Power	Power Connection
VDD (core)	22	—	Power	Power Connection
VDD (core)	32	—	Power	Power Connection
VDD (core)	103	—	Power	Power Connection
VDD (core)	112	—	Power	Power Connection
VSS (pad)	15	—	Power	Ground Connection
VSS (pad)	46	—	Power	Ground Connection
VSS (pad)	60	—	Power	Ground Connection
VSS (pad)	75	—	Power	Ground Connection
VSS (pad)	89	—	Power	Ground Connection
VSS (pad)	133	—	Power	Ground Connection
VSS (pad)	147	—	Power	Ground Connection
VSS (core)	10	—	Power	Ground Connection
VSS (core)	27	—	Power	Ground Connection
VSS (core)	37	—	Power	Ground Connection
VSS (core)	83	—	Power	Ground Connection
VSS (core)	107	—	Power	Ground Connection

1. The chip_sel_pin is an additional qualifier (active low) to the sb_sel_ line. When sharing a single SBus select line with another device (such as the NCR89C100), one of the high order SBus physical address lines, such as PA[27], can be tied to the chip_sel_ pin to distinguish between the two devices.
2. In some system configurations, the high order physical address bits can be connected as follows: sb_a[v, w, x, y, z]=PA[24:20].

Pinout by Pin Number Sequence

Table 6-5 Pinout by Pin Number Sequence

Pin	Name	Direction	Type	Description
1	enet_irq_	I	BSINPDS18	Ethernet interrupt request
2	scsi_irq_	I	BSINPDS18	SCSI interrupt request
3	fpy_densense	I	BSINPDS18	Density sense input (auxio1 register bit)
4	fpy_diskchg	I	BSINPDS18	Disk change
5	fpy_wrprot	I	BSINPDS18	Write protect
6	fpy_track0	I	BSINPDS18	Track 0 indicator
7	fpy_rddata	I	BSINPDS38	Read data
8	fpy_index	I	BSINPDS18	Track index
9	fpy_clk24	I	BSINPDS38	24 MHz clock for floppy ASF
10	VSS (core)	—	Power	Ground Connection
11	fpy_eject	O	BSOTPD16L	Floppy eject (actually ME[3] of 82077 ASF)
12	fpy_headsel	O	BSOTPD16L	Head select
13	fpy_step	O	BSOTPD16L	Drive step pulse
14	fpy_dir	O	BSOTPD16L	Head step direction
15	VSS (pad)	—		Ground Connection
16	fpy_wrdata	O	BSOTPD16L	Write data
17	fpy_wrgate	O	BSOTPD16L	Write enable
18	fpy_drvsel	O	BSOTPD16L	Floppy drive select (DS[0] of 82077 ASF)
19	VDD (pad)	-		Power Connection
20	fpy_moten	O	BSOTPD16L	Floppy motor enable (ME[0] of 82077 ASF)
21	fpy_densel	O	BSOTPD16L	Density select (ME[2] or DENSEL of 82077 ASF)
22	VDD (core)	—		Power Connection
23	fpy_clk32	I	BSINPDS38	32 MHz clock for floppy DDS
24	sb_pa[0]	I	BSINPDS18	Low order physical address bits
25	sb_pa[1]	I	BSINPDS18	Low order physical address bits
26	sb_pa[2]	I	BSIOPD4S18	Low order physical address bits
27	VSS (core)	—		Ground Connection
28	sb_pa[3]	I	BSIOPD4S18	Low order physical address bits

Table 6-5 Pinout by Pin Number Sequence (Continued)

Pin	Name	Direction	Type	Description
29	sb_pa[4]	I	BSINPDS18	Low order physical address bits
30	sb_pa[16]	I	BSINPDS18	PA[16] (for system/user selection in int/tmr)
31	sb_pa[z][1]	I	BSINPDS18	High order physical address bits (for slave decodes)
32	VDD (core)	—		Power Connection
33	sb_pa[y][1]	I	BSINPDS18	High order physical address bits (for slave decodes)
34	sb_pa[x][1]	I	BSINPDS18	High order physical address bits (for slave decodes)
35	sb_pa[w][1]	I	BSINPDS18	High order physical address bits (for slave decodes)
36	sb_pa[v][1]	I	BSINPDS18	High order physical address bits (for slave decodes)
37	VSS (core)	—		Ground Connection
38	chip_sel_[2]	I	BSINPDS18	High order physical address bits (for slave decodes)
39	sb_rd	I	BSINPDS18	SBus Read/Write
40	sb_as_	I	BSINPDS18	SBus Address Strobe (addr is valid)
41	sb_sel_	I	BSINPDS18	SBus Select
42	sb_d[0]	B	BSIOPD12S18	SBus Data Bus (LSB)
43	sb_d[1]	B	BSIOPD12S18	SBus Data Bus
44	sb_d[2]	B	BSIOPD12S18	SBus Data Bus
45	sb_d[3]	B	BSIOPD12S18	SBus Data Bus
46	VSS (pad)	—		Ground Connection
47	sb_d[4]	B	BSIOPD12S18	SBus Data Bus
48	sb_d[5]	B	BSIOPD12S18	SBus Data Bus
49	sb_d[6]	B	BSIOPD12S18	SBus Data Bus
50	sb_d[7]	B	BSIOPD12S18	SBus Data Bus
51	sb_d[8]	B	BSIOPD12S18	SBus Data Bus
52	sb_d[9]	B	BSIOPD12S18	SBus Data Bus
53	VDD (pad)	—		Power Connection
54	sb_d[10]	B	BSIOPD12S18	SBus Data Bus

Table 6-5 Pinout by Pin Number Sequence (Continued)

Pin	Name	Direction	Type	Description
55	sb_d[11]	B	BSIOPD12S18	SBus Data Bus
56	sb_d[12]	B	BSIOPD12S18	SBus Data Bus
57	sb_d[13]	B	BSIOPD12S18	SBus Data Bus
58	sb_d[14]	B	BSIOPD12S18	SBus Data Bus
59	sb_d[15]	B	BSIOPD12S18	SBus Data Bus
60	VSS (pad)	—		Ground Connection
61	sb_clk	I	BSCLOCK	SBus Clock Input
62	sb_d[16]	B	BSIOPD12S18	SBus Data Bus
63	sb_d[17]	B	BSIOPD12S18	SBus Data Bus
64	sb_d[18]	B	BSIOPD12S18	SBus Data Bus
65	sb_d[19]	B	BSIOPD12S18	SBus Data Bus
66	sb_d[20]	B	BSIOPD12S18	SBus Data Bus
67	sb_d[21]	B	BSIOPD12S18	SBus Data Bus
68	VDD (pad)	—		Power Connection
69	sb_d[22]	B	BSIOPD12S18	SBus Data Bus
70	sb_d[23]	B	BSIOPD12S18	SBus Data Bus
71	sb_d[24]	B	BSIOPD12S18	SBus Data Bus
72	sb_d[25]	B	BSIOPD12S18	SBus Data Bus
73	sb_d[26]	B	BSIOPD12S18	SBus Data Bus
74	sb_d[27]	B	BSIOPD12S18	SBus Data Bus
75	VSS (pad)	—		Ground Connection
76	sb_d[28]	B	BSIOPD12S18	SBus Data Bus
77	sb_d[29]	B	BSIOPD12S18	SBus Data Bus
78	sb_d[30]	B	BSIOPD12S18	SBus Data Bus
79	sb_d[31]	B	BSIOPD12S18	SBus Data Bus (MSB)
80	sb_size[2]	I	BSINPDS18	SBus Transfer Size
81	sb_size[1]	I	BSINPDS18	SBus Transfer Size
82	sb_size[0]	I	BSINPDS18	SBus Transfer Size
83	VSS (core)	—	Power	Ground Connection
84	sb_ack[2]	T	BSIOPD12T	SBus Acknowledge
85	sb_ack[1]	T	BSIOPD12T	SBus Acknowledge
86	sb_ack[0]	T	BSIOPD12T	SBus Acknowledge
87	ser_dtr_a_	O	BSOTPD4	Data terminal ready A
88	ser_txd_a	O	BSOTPD4	Transmit data A

Table 6-5 Pinout by Pin Number Sequence (Continued)

Pin	Name	Direction	Type	Description
89	VSS (pad)	—	Power	Ground Connection
90	ser_rts_a_	O	BSOTPD4	Request to send A
91	ser_sync_a_	B	BSIOPD4S18	Sync IO, A
92	ser_trxc_a_	B	BSIOPD4S18	Transmit clock A
93	ser_txd_b	O	BSOTPD4	Transmit data B
94	ser_trxc_b_	B	BSIOPD4S18	Transmit clock B
95	VDD (pad)	—	Power	Power Connection
96	ser_rts_b_	O	BSOTPD4	Request to send B
97	ser_dtr_b_	O	BSOTPD4	Data terminal ready B
98	ser_sync_b_	B	BSIOPD4S18	Sync IO, B
99	kbd_dout	O	BSOTPD4	Keyboard data out
100	mse_out	O	BSOTPD4	Mouse data out
101	power_off	O	BSIOPD4T	Power off output (to power supply)
102	jtag_tdo	O	NCROTPD4	JTAG test data output
103	VDD (core)	—		Power Connection
104	ser_clk	I	BSINPDS38	19.66 MHz serial clock
105	ser_rtxc_b_	I	BSINPDS18	Receive/transmit clock B
106	ser_cts_b_	I	BSINPDS18	Clear to send B
107	VSS (core)	—		Ground Connection
108	ser_rxd_b	I	BSINPDS18	Receive data B
109	ser_dcd_b_	I	BSINPDS18	Data carrier detect B
110	ser_rtxc_a_	I	BSINPDS18	Receive/transmit clock A
111	ser_cts_a_	I	BSINPDS18	Clear to send A
112	VDD (core)	—	Power	Power Connection
113	ser_rxd_a	I	BSINPDS18	Receive data A
114	ser_dcd_a_	I	BSINPDS18	Data carrier detect A
115	link_test_en	O	BSIOPD4T	T7213 link test enable
116	mse_din	I	BSINPDS18	Mouse data in
117	kbd_din	I	BSINPDS18	Keyboard data in
118	mon_mse_mux	O	BSIOPD4T	Monitor/mouse mux select
119	jtag_tdi	I	NCRIPPD	JTAG test data input (100 uA pull-up)
120	jtag_clk	I	NCRINPD	JTAG clock

Table 6-5 Pinout by Pin Number Sequence (Continued)

Pin	Name	Direction	Type	Description
121	jtag_tms	I	NCRIPPD	JTAG test mode select (100 µA pull-up)
122	jtag_rst	I	NCRIPPD	JTAG reset (100 µA pull-up)
123	generic_rdy_	I	BSIPPDU25S18	Generic port ready (25 µA pull-up)
124	iu_error_	I	BSIOPD4S18	Processor watchdog reset/ Video interrupt (SuperSPARC mode)
125	por_rst_in_	I	BSINPDS38	Powerup reset input
126	generic_cs_	O	BSIOPD4T	Generic port chip select
127	sys_led	O	BSOTPD4	System LED output
128	eprom_cs_	O	BSOTPD4	EPROM chip select
129	tod_cs_	O	BSOTPD4	TOD chip select
130	eb_rd_	O	BSIOPD8T	EBus read
131	eb_wr_	O	BSIOPD8T	Ebus write
132	eb_d[0]	B	BSIOPD8S18	EBus data
133	VSS (pad)	—	Power	Ground Connection
134	eb_d[1]	B	BSIOPD8S18	EBus data
135	eb_d[2]	B	BSIOPD8S18	EBus data
136	eb_d[3]	B	BSIOPD8S18	EBus data
137	eb_d[4]	B	BSIOPD8S18	EBus data
138	eb_d[5]	B	BSIOPD8S18	EBus data
139	eb_d[6]	B	BSIOPD8S18	EBus data
140	VDD (pad)	—	Power	Power Connection
141	eb_d[7]	B	BSIOPD8S18	EBus data
142	eb_addrlatch	O	BSOTPD4	EBus address latch
143	sys_rst_out_	O	BSIOPD12T	System (SBus) reset output
144	mmc_rst_out_	O	BSOTPD4	MMC reset output (SuperSPARC mode)
145	m0_irl[0]	O	BSOTPD4	Module 0 encoded interrupt level
146	m0_irl[1]	O	BSOTPD4	Module 0 encoded interrupt level
147	VSS (pad)	—	Power	Ground Connection
148	m0_irl[2]	O	BSOTPD4	Module 0 encoded interrupt level
149	m0_irl[3]	O	BSOTPD4	Module 0 encoded interrupt level

Table 6-5 Pinout by Pin Number Sequence (Continued)

Pin	Name	Direction	Type	Description
150	clk_10mhz	I	BSCLOCKS38	10 MHz clock for counter/ timer block
151	sb_irq_[1]	I	BSINPDS18	SBus interrupt requests
152	sb_irq_[2]	I	BSINPDS18	SBus interrupt requests
153	sb_irq_[3]	I	BSINPDS18	SBus interrupt requests
154	sb_irq_[4]	I	BSINPDS18	SBus interrupt requests
155	sb_irq_[5]	I	BSINPDS18	SBus interrupt requests
156	sb_irq_[6]	I	BSINPDS18	SBus interrupt requests
157	sb_irq_[7]	I	BSINPDS18	SBus interrupt requests
158	pfd_	I	BSINPDS18	Power fail detect (level 15 interrupt)
159	moderr_irq_	I	BSINPDS18	Processor level 15 interrupt (async error)
160	msi_irq_	I	BSINPDS18	MSI interrupt (SuperSPARC mode)

1. In some system configurations, the high order physical address bits are connected as follows: sb_a[v,w,x,y,z] = PA[24:20].
2. The chip_sel_ pin is an additional qualifier (active low) to the sb_sel_line. Where the 89C100 and the 89C105 share a single SBus select line, PA[27] can be used to select between the two, with PA[27] = 0 selecting the 89C105.

JTAG Boundary Information

Table 6-66-6 gives a description of the boundary scan chain. The number listed in the Input, Output, and Enable columns represents the bit order of the scan chain. Bit 0 is the closest register to jtag_tdi.

Table 6-6 The 89C105 Boundary Chain Description

Pin	Name	Type	Input	Output	Enable
1	enet_irq_	INPUT	0	NA	NA
2	scsi_irq_	INPUT	1	NA	NA
3	fpy_densense	INPUT	2	NA	NA
4	fpy_diskchg	INPUT	3	NA	NA
5	fpy_wrprot	INPUT	4	NA	NA
6	fpy_track0	INPUT	6	NA	NA
7	fpy_rddata	INPUT	7	NA	NA
8	fpy_index	INPUT	8	NA	NA
9	fpy_clk24	INPUT	9	NA	NA
10	core_gnd	POWER	NA	NA	NA
11	fpy_eject	TRISTATE	NA	10	193
12	fpy_headsel	TRISTATE	NA	11	193
13	fpy_step	TRISTATE	NA	12	193
14	fpy_dir	TRISTATE	NA	13	193
15	io_gnd	POWER	NA	NA	NA
16	fpy_wrdata	TRISTATE	NA	14	193
17	fpy_wrgate	TRISTATE	NA	15	193
18	fpy_drvsel	TRISTATE	NA	16	193
19	io_vcc	POWER	NA	NA	NA
20	fpy_moten	TRISTATE	NA	17	193
21	fpy_densel	TRISTATE	NA	18	193
22	core_vcc	POWER	NA	NA	NA
23	fpy_clk32	INPUT	19	NA	NA
24	sb_pa[0]	INPUT	20	NA	NA
25	sb_pa[1]	INPUT	21	NA	NA
26	sb_pa[2]	BIDIR	22	23	5
27	core_gnd	POWER	NA	NA	NA
28	sb_pa[3]	BIDIR	24	25	5
29	sb_pa[4]	INPUT	26	NA	NA
30	sb_pa[16]	INPUT	27	NA	NA

Table 6-6 The 89C105 Boundary Chain Description (Continued)

Pin	Name	Type	Input	Output	Enable
31	sb_pa[z]	INPUT	28	NA	NA
32	core_vcc	POWER	NA	NA	NA
33	sb_pa[y]	INPUT	29	NA	NA
34	sb_pa[x]	INPUT	30	NA	NA
35	sb_pa[w]	INPUT	31	NA	NA
36	sb_pa[v]	INPUT	32	NA	NA
37	core_vss	POWER	NA	NA	NA
38	chip_sel_	INPUT	33	NA	NA
39	sb_rd	INPUT	34	NA	NA
40	sb_as_	INPUT	35	NA	NA
41	sb_sel_	INPUT	36	NA	NA
42	sb_d[0]	BIDIR	37	38	53
43	sb_d[1]	BIDIR	39	40	53
44	sb_d[2]	BIDIR	41	42	53
45	sb_d[3]	BIDIR	43	44	53
46	io_vss	POWER	NA	NA	NA
47	sb_d[4]	BIDIR	45	46	53
48	sb_d[5]	BIDIR	47	48	53
49	sb_d[6]	BIDIR	49	50	53
50	sb_d[7]	BIDIR	51	52	53
51	sb_d[8]	BIDIR	54	55	53
52	sb_d[9]	BIDIR	56	57	53
53	io_vdd	POWER	NA	NA	NA
54	sb_d[10]	BIDIR	58	59	53
55	sb_d[11]	BIDIR	60	61	53
56	sb_d[12]	BIDIR	62	63	53
57	sb_d[13]	BIDIR	64	65	53
58	sb_d[14]	BIDIR	66	67	53
59	sb_d[15]	BIDIR	68	69	53
60	io_vss	POWER	NA	NA	NA
61	sb_clk	INPUT	70	NA	NA
62	sb_d[16]	BIDIR	71	72	87
63	sb_d_17	BIDIR	73	74	87
64	sb_d_18	BIDIR	75	76	87

Table 6-6 The 89C105 Boundary Chain Description (Continued)

Pin	Name	Type	Input	Output	Enable
65	sb_d_19	BIDIR	77	78	87
66	sb_d_20	BIDIR	79	80	87
67	sb_d_21	BIDIR	81	82	87
68	io_vcc	POWER	NA	NA	NA
69	sb_d_22	BIDIR	83	84	87
70	sb_d_23	BIDIR	85	86	87
71	sb_d_24	BIDIR	88	89	87
72	sb_d_25	BIDIR	90	91	87
73	sb_d_26	BIDIR	92	93	87
74	sb_d_27	BIDIR	94	95	87
75	io_gnd	POWER	NA	NA	NA
76	sb_d_28	BIDIR	96	97	87
77	sb_d_29	BIDIR	98	99	87
78	sb_d_30	BIDIR	100	101	87
79	sb_d_31	BIDIR	102	103	87
80	sb_size_2	INPUT	104	NA	NA
81	sb_size_1	INPUT	105	NA	NA
82	sb_size_0	INPUT	106	NA	NA
83	core_gnd	POWER	NA	NA	NA
84	sb_ack_2_	BIDIR	107	108	109
85	sb_ack_1_	BIDIR	110	111	109
86	sb_ack_0_	BIDIR	113	114	112
87	ser_dtr_a_	TRISTATE	NA	115	193
88	ser_txd_a	TRISTATE	NA	116	193
89	io_vss	POWER	NA	NA	NA
90	ser_rts_a_	TRISTATE	NA	117	193
91	ser_sync_a_	BIDIR	119	120	118
92	ser_trxc_a_	BIDIR	122	123	121
93	ser_txd_b	TRISTATE	NA	124	193
94	ser_trxc_b_	BIDIR	126	127	125
95	io_vdd	POWER	NA	NA	NA
96	ser_rts_b_	TRISTATE	NA	128	193
97	ser_dtr_b_	TRISTATE	NA	129	164
98	ser_sync_b_	BIDIR	131	132	130

Table 6-6 The 89C105 Boundary Chain Description (Continued)

Pin	Name	Type	Input	Output	Enable
99	kbd_dout	TRISTATE	NA	133	164
100	mse_out	TRISTATE	NA	134	164
101	power_off	BIDIR	135	136	145
102	jtag_tdo	TDO	NA	NA	NA
103	core_vdd	POWER	NA	NA	NA
104	ser_clk	INPUT	137	NA	NA
105	ser_rtxc_b_	INPUT	138	NA	NA
106	ser_cts_b_	INPUT	139	NA	NA
107	core_vss	POWER	NA	NA	NA
108	ser_rxd_b	INPUT	140	NA	NA
109	ser_dcd_b_	INPUT	141	NA	NA
110	ser_rtxc_a_	INPUT	142	NA	NA
111	ser_cts_a_	INPUT	143	NA	NA
112	core_vdd	POWER	NA	NA	NA
113	ser_rxd_a	INPUT	144	NA	NA
114	ser_dcd_a_	INPUT	146	NA	NA
115	link_test_en	BIDIR	147	148	145
116	mse_din	INPUT	149	NA	NA
117	kbd_din	INPUT	150	NA	NA
118	mon_mse_mux	BIDIR	152	153	151
119	jtag_tdi	TDI	NA	NA	NA
120	jtag_clk	TCK	NA	NA	NA
121	jtag_tms	TMS	NA	NA	NA
122	jtag_rst	TRSTB	NA	NA	NA
123	generic_rdy_	INPUT	154	NA	NA
124	iu_error_	BIDIR	155	156	5
125	por_rst_in_	INPUT	157	NA	NA
126	generic_cs_	BIDIR	158	159	145
127	sys_led	TRISTATE	NA	160	164
128	eprom_cs_	TRISTATE	NA	161	164
129	tod_cs_	TRISTATE	NA	163	162
130	eb_rd_	BIDIR	165	166	164
131	eb_wr_	BIDIR	167	168	164
132	eb_d[0]	BIDIR	169	170	177

Table 6-6 The 89C105 Boundary Chain Description (Continued)

Pin	Name	Type	Input	Output	Enable
133	io_vss	POWER	NA	NA	NA
134	eb_d[1]	BIDIR	171	172	177
135	eb_d[2]	BIDIR	173	174	177
136	eb_d[3]	BIDIR	175	176	177
137	eb_d[4]	BIDIR	178	179	177
138	eb_d[5]	BIDIR	180	181	177
139	eb_d[6]	BIDIR	182	183	177
140	io_vdd	POWER	NA	NA	NA
141	eb_d[7]	BIDIR	184	185	177
142	eb_addlatch	TRISTATE	NA	187	186
143	sys_rst_out_	BIDIR	188	189	164
144	mmc_rst_out_	TRISTATE	NA	190	164
145	m0_irl[0]	TRISTATE	NA	191	164
146	m0_irl[1]	TRISTATE	NA	192	164
147	io_vss	POWER	NA	NA	NA
148	m0_irl[2]	TRISTATE	NA	194	164
149	m0_irl[3]	TRISTATE	NA	195	164
150	clk_10mhz	INPUT	196	NA	NA
151	sb_irq_[1]	INPUT	197	NA	NA
152	sb_irq_[2]	INPUT	198	NA	NA
153	sb_irq_[3]	INPUT	199	NA	NA
154	sb_irq_[4]	INPUT	200	NA	NA
155	sb_irq_[5]	INPUT	201	NA	NA
156	sb_irq_[6]	INPUT	202	NA	NA
157	sb_irq_[7]	INPUT	203	NA	NA
158	pfd_	INPUT	204	NA	NA
159	moderr_irq_	INPUT	205	NA	NA
160	msi_irq_	INPUT	206	NA	NA

Functional Operation

This section includes the following:

- Detailed chip block diagram
- Chip-level address map
- Functional chip description

Detailed Chip Block Diagram

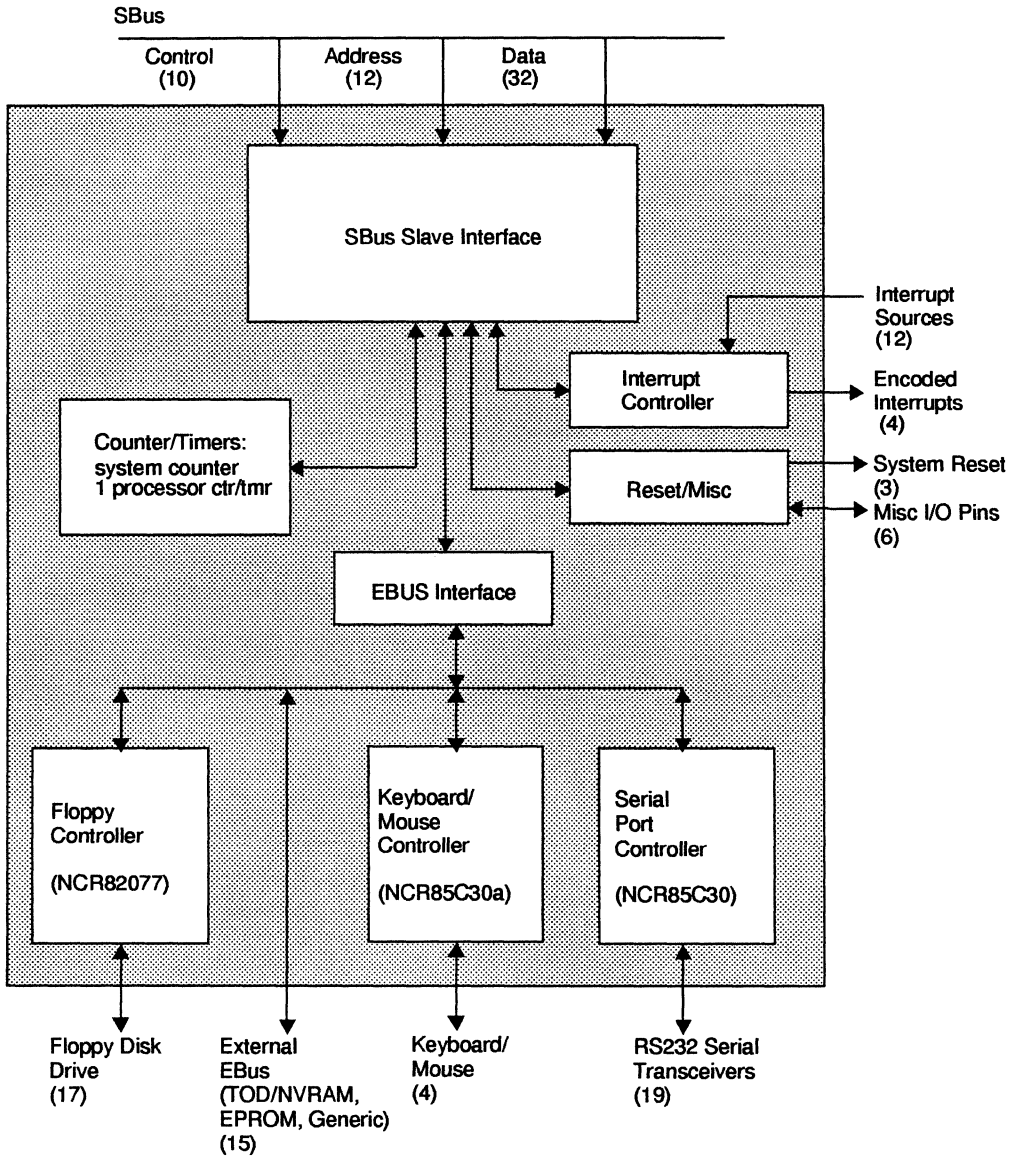


Figure 6-3 Chip block diagram

Chip-Level Address Map

The various devices and registers in the 89C105 support different size accesses. The address map indicates the allowed accesses at each address. B indicates that a byte access is allowed, H indicates an halfword access, W indicates a word access, and D indicates a double-word access.

Table 6-7 Chip-Level Address Map

PA[27:00]	Device	Access
000 0000 -> 0FF FFFF	Boot PROM	B,H,W
100 0000 -> 11F FFFF	**Keyboard, Mouse, and Serial Ports**	B
100 0000	Mouse Control Port	
100 0002	Mouse Data Port	
100 0004	Keyboard Control Port	
100 0006	Keyboard Data Port	
110 0000	TTYB Control Port	
110 0002	TTYB Data Port	
110 0004	TTYA Control Port	
110 0006	TTYA Data Port	
120 0000 -> 12F FFFF	**TOD/NVRAM**	B,H,W
130 0000 -> 13F FFFF	**General Purpose (Generic Port)**	B
140 0000 -> 14F FFFF	**Floppy Controller**	B
140 0002	Digital Output Register (DOR)	
140 0004	Main Status Register (MSR, Read Only)	
140 0004	Datarate Select Register (DSR, Write Only)	
140 0005	FIFO	
140 0006	Reserved (Test mode select)	
140 0007	Digital Input Register (DIR, Read Only)	
140 0007	Configuration Control Register (CCR, Write Only)	
150 0000 -> 170 0000	**Reserved**	
180 0000	**89C105 Configuration Register**	B
190 0000 -> 19F FFFF	**Auxiliary I/O Registers**	B
190 0000	Aux 1 Register (Miscellaneous System Functions)	
191 0000	Aux 2 Register (Software Powerdown Control)	

Table 6-7 Chip-Level Address Map (Continued)

PA[27:00]	Device	Access
1A0 0000	**Diagnostic Message Register**	B
1B0 0000	**Modem Register**	B
1C0 0000 -> 1CF FFFF	**Reserved**	
1D0 0000 -> 1DF FFFF	**Counter/Timers**	W,D
1D0 0000	Processor Counter Limit Register or User Timer MSW	
1D0 0004	Processor Counter Register or User Timer LSW	
1D0 0008	Processor Counter Limit Register (non-resetting port)	
1D0 000C	Processor Counter User Timer Start/Stop Register	
1D1 0000	System Limit Register (Level 10 Interrupt)	
1D1 0004	System Counter Register	
1D1 0008	System Limit Register (non-resetting port)	
1D1 000C	Reserved	
1D1 0010	Timer Configuration Register	
1E0 0000 -> 1EF FFFF	**Interrupt Controller**	W
1E0 0000	Processor Interrupt Pending Register	
1E0 0004	Processor Clear-Pending Pseudo-Register	
1E0 0008	Processor Set-Soft-Interrupt Pseudo-Register	
1E1 0000	System Interrupt Pending Register	
1E1 0004	Interrupt Target Mask Register	
1E1 0008	Interrupt Target Mask Clear Pseudo -Register	
1E1 000C	Interrupt Target Mask Set Pseudo-Register	
1E1 0010	Interrupt Target Register (Reads as 0, Write has no effect)	
1F0 0000	**System Control/Status Register**	W

Reads or writes of reserved addresses and accesses using a size that is not valid for the selected address range will result in an SBus error acknowledgment. Other accesses will result in either a byte or word acknowledgment, depending on the access.

The addresses in Table 6-7 and elsewhere in this specification assume the following address connections: SB_PA[v,w,x,y,z] = PA[24:20], and chip_sel_ = PA[27]. Other connections are possible, but the address map will be different for the different setups.

Functional Description

Overview

The 89C105 consists of five major functional units, plus test logic. Three of the functional units are application specific functions (ASF) designed by NCR to replicate the functionality of complete board-level chips: the 82077 floppy controller and the 85c30 serial ports (the 89C105 contains two 85C30 ASFs, one used for keyboard/mouse interface, and one for general-purpose tty serial ports). The other two functional blocks are the SBus interface/interrupt logic and counter/timers. The SBus logic interfaces between the external world and all functional blocks within the 89C105, as well as other devices accessed through the 89C105, such as the EPROM and NVRAM. The counter/timers have a 32/64-bit interface (32-bit data input, 64-bit data output to support two word bursts), while all of the ASFs and the other devices reside on an 8-bit bus expansion bus referred to as the EBus.

89C100 and 89C105 Interdependencies

When the 89C100 and the 89C105 are used together, the 89C105 receives three clocks from the 89C100 (fpy_clk24, fpy_clk32, and scc_clk_20). The 89C100 simply provides oscillator pads on its pins because of a pin limitation on the 89C105. The 89C100 does not use these clock signals internally. The 89C100 also sends its three interrupt signals to the 89C105 for processing, they are; sb_d_irq_, sb_e_irq_, and sb_p_irq_ for; SCSI, Ethernet, and parallel port interrupts, respectively. Refer to "NCR89C100 Master I/O" for a description of how the 89C100 generates interrupts.

Technology

The 89C105 is a standard cell design, based on the NCR VS700H technology (.95 μ drawn,.7 effective). It consists of 40,000 equivalent gates.

Start-Up Information

The 89C105 provides the system reset function (SBus reset, and, optionally, a separate Memory Controller reset for SuperSPARC-based systems). It takes a reset input indicating that the power supply voltage is valid, and produces reset outputs to initialize the system.

Chip Reset Information

On start-up, the 89C105 must have its reset input (POR_RST_IN_) asserted for at least 15 SBus clocks. Reset output is asserted for the entire time that the reset input is asserted, plus an additional reset time intended to guarantee that the various onboard oscillators have stabilized and initialization is complete. The reset signals are described below; see the System Status and System Control register section for a register-level description of the reset function and the timings associated with it.

Table 6-8 Chip Reset Signals Chart

Signal	Direction	Synopsis
por_rst_in_	Input, active low	Reset input–low until power supply stable
sys_rst_out_	Output, active low	System (SBus) reset.
mmc_rst_out_	Output, active low	Memory Controller reset

Processor Status Signals

Watchdog Resets and Module Error (Level-15) interrupts are communicated to the 89C105 (which serves as the system reset and interrupt master) via two low-active pins: IU_ERROR_ and MODERR_IRQ_. These signals are described in the System Status and System Control and Interrupt Control functional block descriptions.

NOTE: The IU_ERROR_ signal is only used by the microSPARC processor. In other systems, it should be tied high.

Buses

The 89C105 has two bus interfaces: the system SBus and a slow, eight-bit expansion bus (EBus) for devices such as EPROM, NVRAM, etc. The eight-bit bus is called the EBus.

SBus

The 89C105 uses the SBus as its system interface, and conforms to SBus Rev B.0. As allowed in the specification, it only supports a subset SBus functionality. The description of this subset follows. The chip-level address map lists valid access sizes for each address range, and this is described in more detail in each individual functional block's section.

Subset of SBus Features Supported

The 89C105 supports only slave SBus accesses. During a slave access, the 89C105 takes control of sb_ack[2:0] signals. Halfword or word accesses to the EPROM port will receive an 8-bit acknowledge, so the system SBus controller must support dynamic bus sizing. Burst accesses to the EPROM or any other address range other than the counter/timers will receive an error acknowledge. The counter/timers will accept two word burst accesses, but only to the User Timer Count Register, and only when configured to use the user timer mode of operation.

Table 6-9 represents all possible SBus responses. The 89C105's SBus interface can only generate those responses marked with a **. Any access to a reserved section of the 89C105's address space or any access that uses a size that is invalid for the address range accessed will result in an SBus error acknowledgment.

Table 6-9 sb_ack Mapping

sb_ack[2]	sb_ack[1]	sb_ack[0]	Definition
1	1	1	insert wait states **
1	1	0	error **
1	0	1	8-bit port ack **
1	0	0	rerun
0	1	1	32-bit port ack **
0	1	0	double-word ack
0	0	1	16-bit port ack
0	0	0	reserved

Eight-Bit Bus (EBus)

The EBus supports 8-bit "Intel-style" peripherals, with interface controls (chip select, read, write) that operate similarly to peripheral chips manufactured by the Intel Corporation (see the *Timing Diagrams* section for typical EBus cycle diagrams). Write accesses are single buffered in order to reduce SBus overhead. This means that the 89C105 will acknowledge SBus write accesses immediately, before completing the corresponding EBus cycle. Note that a second write immediately following the first has to wait for that first EBus cycle to complete (and the single write buffer to become available) before the 89C105 will acknowledge it and free the SBus. The Timing Diagrams section shows several examples of this write buffering.

EBus Timings

The EBus access times are customized for each device. The read/write pulse timings are given below (the read/write pulses are bracketed by chip select, which begins one SB_CLK before and ends one SB_CLK afterwards; see the Timing Diagram section for examples). The 89C105 contains hardware counters to ensure that the serial port minimum recovery times are met, up to the maximum SBus frequency of 25 MHz.

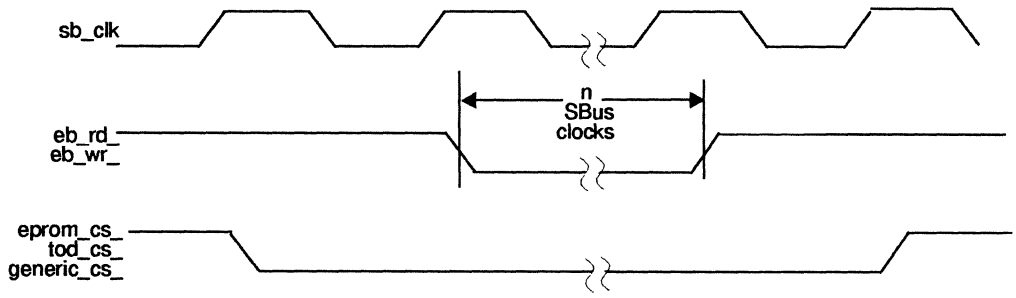


Figure 6-4 EBus Timings: Chip Selects and eb_rd_/eb_wr

Table 6-10 EBus Read/Write Timing

Device	SBus Clocks	16 MHz	20 MHz	25 MHz
Serial A/B	4 clocks	240 ns	200 ns	160 ns
Keyboard/Mouse	4 clocks	240 ns	200 ns	160 ns
Floppy	3 clocks	180 ns	150 ns	120 ns
NVRAM (wr)	5 clocks	300 ns	250 ns	200 ns
NVRAM (rd)	7 clocks	420 ns	350 ns	280 ns
EPROM	6 clocks	360 ns	300 ns	240 ns

EBus Address Map

Table 6-11 EBus Address Map

PA[27:00]	Device	Access
000 0000 -> 0FF FFFF	**Boot PROM**	B,H,W
100 0000 -> 11F FFFF	**Keyboard, Mouse, and Serial Ports**	B
100 0000	Mouse Control Port	
100 0002	Mouse Data Port	
100 0004	Keyboard Control Port	
100 0006	Keyboard Data Port	
110 0000	TTYB Control Port	
110 0002	TTYB Data Port	
110 0004	TTYA Control Port	
110 0006	TTYA Data Port	
120 0000 -> 12F FFFF	**TOD/NVRAM**	B,H,W
130 0000 -> 13F FFFF	**General Purpose (Generic port)**	B
140 0000 -> 14F FFFF	**Floppy Controller**	B
140 0002	Digital Output Register (DOR)	
140 0004	Main Status Register (MSR, Read Only)	
140 0004	Datarate Select Register (DSR, Write Only)	
140 0005	FIFO	
140 0006	Reserved (Test mode select)	
140 0007	Digital Input Register (DIR, Read Only)	
140 0007	Configuration Control Register (CCR, Write Only)	

Devices Residing on the EBus

The following devices reside on the EBus:

- EPROM
- TOD/NVRAM
- Floppy Controller
- Serial Controller A
- Serial Controller B
- Keyboard Controller
- Mouse Controller
- General Purpose Decode (Generic port)

Functional Blocks

This section includes block diagrams, functional descriptions, and block-level address maps for the following:

- Boot PROM
- TOD/NVRAM
- Floppy Controller
- Generic Port
- Serial Controller
- Keyboard/Mouse Controller
- System Status and System Control
- Interrupt Control
- Counter-Timers
- Chip Configuration Control
- Diagnostic Message Register
- Miscellaneous System Functions (Aux I/O Modem Registers)

Overview

The 89C105 consists of several major functional blocks: eight-bit devices (both external devices residing on the EBus and internal ASFs), an interrupt controller, a set of counter-timers, a system status and control block that generates system resets, and miscellaneous other system and configuration registers. Each of these blocks is described in detail below.

External Eight-Bit Devices

The following devices reside on the external EBus: EPROM, TOD/NVRAM. In addition, the 89C105 provides a general purpose decode (Generic Port) for other eight-bit devices.

Boot PROM

The Boot PROM contains start-up information that gets accessed immediately after any reset.

Boot PROM Address Map

The Boot PROM is located at the following address as shown in Table 6-12:

Table 6-12 Boot PROM Address Map

PA[27:00]	Device	R/W
000 0000 - 0FF FFFF	Boot PROM	R

An external chip select (eprom_cs_) is generated for transfers from this device. When an SBus read addressed to the EPROM space is detected, the chip select is asserted, and data is transferred from the PROM to the 8-bit expansion bus. The 89C105 passes this data through to the SBus and asserts sb_ack_ to end the transfer. The 89C105 will only assert a byte acknowledge, so it relies on the SBus controller for bus sizing for halfword or word transfers. If a burst transfer is attempted to this address range,

the 89C105 will return an error acknowledge.

TOD/NVRAM

The Time-of-Day clock/Non-Volatile RAM port is designed to support either a Thomson Mostek MK48T02 (2K NVRAM) or MK48T08 (8K NVRAM).

TOD/NVRAM Address Map

The Time-of-Day/Non-Volatile RAM have the address maps shown in Table 6-13 and Table 6-14:

Table 6-13 TOD/NVRAM (MK 48T02) Address Map

PA[27:00]	Device	R/W
120 0000 - 120 07F7	NVRAM (see software document)	R/W
120 07F8	TOD Control	R/W
120 07F9	Seconds (00-59)	R
120 07FA	Minutes (00-59)	R
120 07FB	Hour (00-23)	R
20 07FC	Day (01-07)	R
120 07FD	Date (00-31)	R
120 07FE	Month (01-12)	R
120 07FF	Year (00-99)	R

Table 6-14 TOD/NVRAM (MK48T08) Address Map

PA[27:00]	Device	R/W
120 0000 - 120 1FF7	NVRAM (see software document)	R/W
120 1FF8	TOD Control	R/W
120 1FF9	Seconds (00-59)	R
120 1FFA	Minutes (00-59)	R
120 1FFB	Hour (00-23)	R
120 1FFC	Day (01-07)	R
120 1FFD	Date (00-31)	R
120 1FFE	Month (01-12)	R
120 1FFF	Year (00-99)	R

A chip select pin (tod_cs) is used to select the chip so that data can be transferred to or from the 8-bit expansion bus under control of the EBUS data direction signals

(eb_wr_, eb_rd_). The 89C105 will only assert a byte acknowledge, so it relies on the SBus controller for bus sizing for halfword or word transfers. If a burst transfer is attempted to this address range, the 89C105 will return an error acknowledge.

Generic Port

The 89C105 can support additional Intel-style peripheral devices on the external EBus through the Generic Port. This port provides an additional chip select (which can be externally qualified with additional SBus address bits if desired). The length of the read/write cycles is programmable though the generic_rdy_ signal. The cycles will last two SBus clocks after the generic_rdy_ signal is sampled low. See the Functional Operation and Timing Diagrams sections for more details on the Generic Port timing.

SBus writes to the Generic Port are buffered, like all EBus writes (see the Functional Operation section for details). If a generic_rdy_ is not received after 15 SBus clocks, writes are terminated, and reads will return an error acknowledge (since the EBus is buffered, a valid acknowledge would already have been given for a write). See the Timing Diagrams section for an example of a Generic time-out.

Internal Eight-bit Devices

The 89C105 contains three internal eight-bit devices: a floppy disk controller, a serial communications controller, and a keyboard/mouse controller.

Floppy Controller

The NCR82077 floppy disk controller is compatible with the Intel 82077AA-1 single-chip floppy disk controller. For detailed information on the NCR82077, refer to the "NCR80277 Floppy Disk Controller Core" section of this manual. In the 89C105, this floppy controller can be used slightly differently than the standard (PC/AT) setup. Details are shown below.

Floppy Controller Address Map

The sub-addresses for the floppy controller are shown in Table 6-15.

Table 6-15 Floppy Controller Address Map

PA[27:00]	Device	R/W
140 0002	Digital Output Register (DOR)	R/W
140 0004	Main Status Register (MSR)	R
140 0004	Data Rate Select Register (DSR)	W
140 0005	Data FIFO	R/W
140 0006	RESERVED (test mode)	R
140 0007	Digital Input Register (DIR)	R
140 0007	Configuration Control Register (CCR)	W

Floppy Controller Use

In the 89C105, the 82077 floppy controller is used in a fairly unique way. All data is transferred using interrupt-driven programmed I/O instead of the more standard DMA setup. This requires that the Terminal Count bit be set under software control

(see the AuxIO register description for more details). In addition, the floppy disk interface is tailored to use a non-standard floppy drive that supports three density modes (720 KB, 1.2 MB, 1.44 MB) and has an auto-eject feature. To support this drive, some pins on the ASF, which are controlled by way of the DOR are assigned new functions. In addition, a bit in the Aux I/O Register is used to sense the floppy density. The INVERTb pin of the floppy controller is tied low internal to the 89C105. Because of this, all of the pins which connect to the floppy drive have low polarity.

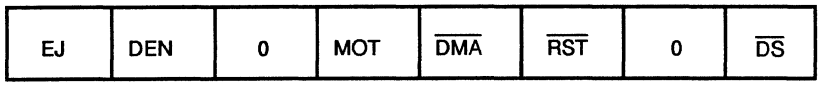


Figure 6-5 DOR Register Field Definitions

Field Definitions

0	DOR[5,1] are unused. They should be masked and ignored on read, and written as 0.
EJ	EJECT. To eject the floppy, write a 1, pause for at least 2us, and then write a 0. The drive must be selected and the motor on for this to work. (This applies to auto-eject floppy drives meeting the Sony MP-F17W-XX auto-eject interface only).
DEN	DENSITY SELECT. This controls the FPY_DENSEL pin (that pin is an inverted version of this register bit). For the Sony MP-F17W-P1 drive, this yields the following density selection:

Table 6-16 Floppy Density Select Operation (Sony MP-F17W-P1 only)

Disk	Density Select	Drive Status	Motor Speed
2DD	X	1.0 MB (720K formatted)	300 RPM
2HD	0	2.0 MB (1.44MB formatted)	300RPM
2HD	1	1.6MB (1.2MB formatted)	360 RPM

MOT	MOTOR ENABLE. Setting this to 1 turns on the floppy drive's motor.
DMA	$\overline{\text{DMAGATE}}$. This must be set to 1 after system reset to allow floppy interrupts to be seen by the rest of the system.
RST	$\overline{\text{RESET}}$. This must be set to 1 after system reset to bring the floppy ASF out of reset.
DS	$\overline{\text{DRIVE SELECT}}$. This must be set to 0 to select the floppy drive.

All bits in the DOR register are cleared to 0 by a system reset.

Floppy Drives Supported

The 89C105 floppy controller is compatible with any PC-style floppy drive that uses MFM encoding (typically 720K and 1.44M floppies). It is also compatible with the extra high density (2.88MB formatted) floppy drives that use the Perpendicular Mode recording format. In a system with a heavily loaded SBus, the interrupt latency in a workstation has been found to be too long in many cases to allow operation at the EHD data rates. This will vary depending on the system being designed.

Serial Controller

The serial port controller used for the general-purpose TTY serial ports is compatible with the AMD AM85C30, rev C Serial Communications Controller. All registers are 8-bit quantities.

The serial ports' input clock (PCLK) is 1/4 of the SERIAL_CLK input pin (the input is internally divided by 4). On typical systems, the SERIAL_CLK input should be driven with a 19.66 MHz clock, which means that PCLK runs at 4.9152 MHz.

Differences from AMD AM85C30 SCC

The NCR 85c30 ASF is functionally compatible with the AMD AM85C30 rev C, with the exception of the enhancements to WR7 and the 10x19 SDLC frame status FIFO. These functions, are not included in the 89C105 serial ports.

Serial Controller Address Map

The sub-addresses for the SCCs are as shown in Table 6-17:

Table 6-17 Serial Controller Address Map

PA[27:00]	Device	R/W
110 0000	TTYB Control Port	R/W
110 0002	TTYB Data Port	R/W
110 0004	TTYA Control Port	R/W
110 0006	TTYA Data Port	R/W

Keyboard/Mouse Controller

The serial port controller used for the keyboard/mouse ports is compatible with the AMD AM85C30, rev C Serial Communications Controller. All registers are 8-bit quantities.

The keyboard/mouse controller runs at the same PCLK rate as the serial controller.

Differences from AMD AM85C30 SCC

The 89C105 uses a reduced-function asynchronous only cell for the keyboard/mouse in order to conserve chip area. All synchronous logic has been removed, as well as some of the async modes not commonly used by drivers. Specific functions that were removed are: PLL, monosync/bisync/SDLC rx/tx circuitry, and any clocking mode (in WR4) except x16.

Keyboard/Mouse Controller Address Map

The sub-addresses for the keyboard/mouse controller are as shown in Table 6-18:

Table 6-18 Keyboard/Mouse Controller Address Map

PA[27:00]	Device	R/W
100 0000	Mouse Control Port	R/W
100 0002	Mouse Data Port	R/W
100 0004	Keyboard Control Port	R/W
100 0006	Keyboard Data Port	R/W

System Status and System Control

The System Status and System Control Register is at physical address 1F0 0000. It is used to generate "software" system resets and to indicate reset sources for diagnostic purposes.

System Status and System Control Register (Reset Register)

31 05 04 03 02 01 00

Figure 6-6 System Reset (Control) Register

Field Definitions:

SR	Software Reset (write-only).When set to 1, generates the equivalent of a power-on reset. This is self-clearing logic, so it will always read as 0.
RS	Reset Status (read/clear only). This bit is set to 1 after a software reset has been issued. It is cleared on a power-on reset.
(R)	Reserved. These bits read as 0; writing has no effect.
WD	Watchdog Reset (read/clear only). This bit is set to 1 when a Watchdog Reset request is received from the processor via the iu_error_ input, as discussed in the Reset section following. It is cleared by either a power-on or software reset, or by writing a 0 to this bit. Writing a 1 has no effect.

This circuit requires that the hardware reset input be asserted on power-up to properly initialize the system during power-up. The SYS_RST_OUT_ pin will be asserted as long as the POR_RST_IN_ reset input is asserted, plus an additional 200 milliseconds or so. The circuit will de-assert SYS_RST_OUT_ synchronous to the rising edge of SBus clock, as defined by the SBus specification.

Resets

There are two sources of reset recognized by the I/O reset controller: Power-on Reset (POR), and Software Reset (SWR). Either of these two sources will cause the 89C105

to assert its two reset outputs, SYS_RST_OUT_ (SBus reset), and MMC_RST_OUT_ (Memory Controller Reset--SuperSPARC systems only). Both the length of the reset pulse and the value left in the System Reset Register will differ depending on the source of the reset. The lengths of the reset pulses are as follows:

Table 6-19 Hardware and Software Resets

Reset Source	SBus Reset (SYS_RST_OUT_) Duration	Memory Controller Reset (MMC_RST_OUT_) Duration
POR_RST_IN_	> 200 milliseconds (starting after POR_RST_IN_ = 1)	> 200 milliseconds (starting after POR_RST_IN_ = 1)
Soft Reset	66 SBus clocks + 2 10 MHz clocks + 3.2e-5 seconds (approx)	4 SBus clocks

The software reset duration given above yields the following durations for SYS_RST_OUT_ (SBus reset) at each SBus frequency:

Table 6-20 SYS_RST_OUT_ Software Reset Duration

16 MHz	20 MHz	25 MHz
36.2 usec (602 SBus clocks)	35.5 usec (710 SBus clocks)	34.8 usec (871 SBus clocks)

These numbers are somewhat approximate due to synchronization between two asynchronous clocks that takes place in the process of generating the reset pulse. This can cause the numbers above to be off by several SBus clocks in one direction or the other.

The reset output SYS_RST_OUT_, controlled by the 89C105, is intended to put the entire system into a known state. The system processor as well as all I/O devices and state machines will be reset. It is not possible to reset part of the system and leave the rest untouched via either of these two resets.

In addition to the above sources, the processor may detect a Watchdog Reset if it experiences an error condition, which is trap with traps disabled. This condition is communicated to the 89C105 via the IU_ERROR_ input, and is latched in the System Reset Register (no other action is taken by the 89C105; the processor has its own watchdog reset circuitry that performs the partial watchdog reset).

When the processor recovers from a reset, it should determine the source of the reset. The hierarchy it should search for this determination is watchdog, SWR, then POR.

Table 6-21 Processor State after POR/SWR

Device, Bus, or Bit	State After POR/SWR
SBus	Reset
Interrupt Mask	All '1' (all sources masked)
Interrupt Target Register	0x0
Soft-Interrupt Bits	All '0'
System Error Bits	All '0'
Counter Timers	Initialized

Any reset source will cause the 89C105 to assert SYS_RST_OUT_ for a minimum of 512 SBus clocks.

Processor Status Pins

Watchdog Resets and Module Error (Level-15) interrupts are communicated to the 89C105 (which serves as the system reset and interrupt master) via two low-active pins: IU_ERROR_ and MODERR_IRQ_. These signals allow the processor to indicate the following conditions:

Table 6-22 Processor Status Codes

moderr_irq_	iu_error	Definition
1	1	Normal Operation
0	1	Module Error (Level-15) Int Request
1	0	Watchdog Reset Indicator
0	0	Both Module Error Interrupt and Watchdog

Interrupt Control

The 89C105 contains an interrupt controller designed for use in uniprocessor systems. It contains the system logic and a single set of processor-specific circuitry

Interrupt Control Block Diagram

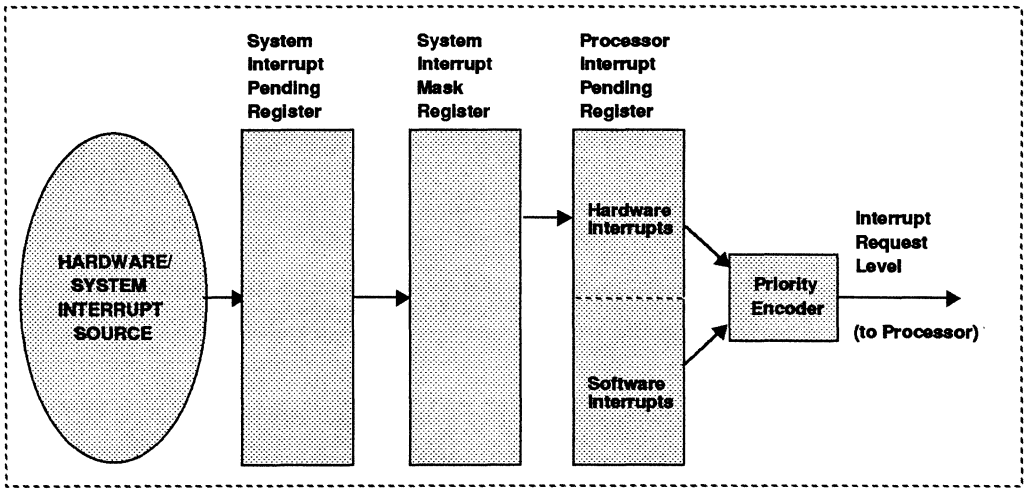


Figure 6-7 Interrupt Control Block Diagram

Interrupt Control Register Definitions

The Interrupt Control Registers are at physical address 1E0 0000. They are defined as follows:

Table 6-23 Interrupt Control Register Definitions

PA[27:00]	Register	R/W
1E0 0000	Processor Interrupt Pending	R
1E0 0004	Processor Clr_Pnd Pseudo-Reg	W
1E0 0008	Processor Set_Soft_Int Pseudo- Reg.	W
1E0 000C- 1E0 3FFF	RESERVED	N/A
1E1 0000	System Interrupt Pending Reg.	R
1E1 0004	Interrupt Target Mask Register	R
1E1 0008	Interrupt Target Mask Clear Pseudo-Reg.	W
1E1 000C	Interrupt Target Mask Set Pseudo-Reg.	W
1E1 0010-1EF FFFF	RESERVED	W

Reading and writing the System Interrupt Pending/Mask register allows the CPU to identify hardware interrupt sources, and to selectively mask those sources, as follows:

One must write/read zeroes to/from reserved bits.

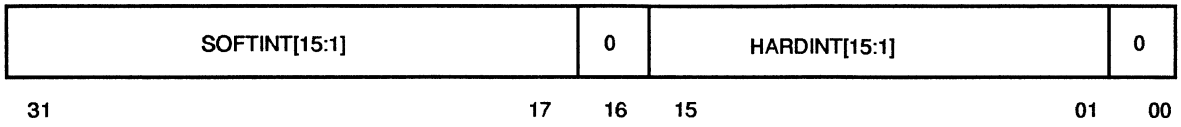


Figure 6-8 Processor Interrupt Pending Register (read only)

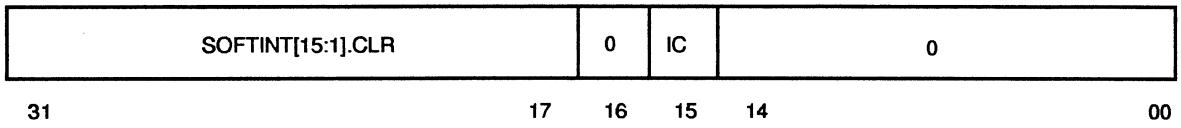


Figure 6-9 Processor Interrupt Clear-Pending Pseudoregister

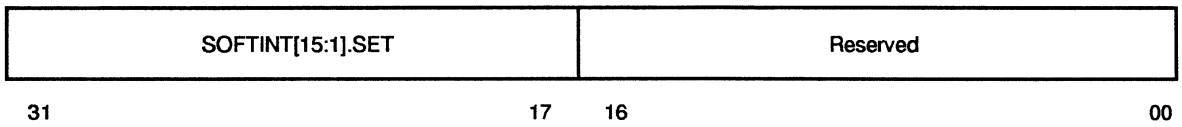


Figure 6-10 Processor Set-Soft-Int Pseudoregister

Field Definitions:

SOFTINT[15:1]	Software Interrupt.
HARDINT[15:1]	Hardware Interrupt.
IC	Interrupt Level 15 Clear.

Writing a 1 to any of the SOFTINT bits or the IC bit in the Interrupt Clear pseudoregister clears the associated interrupt.

The Software Interrupt Set register is used to generate software interrupts. Writes to a given bit in this register will cause the associated bit to be set in the Pending register, and the appropriate level interrupt request will be issued to the CPU.

All pending interrupts are CLEARED, and all mask bits are SET upon system reset.

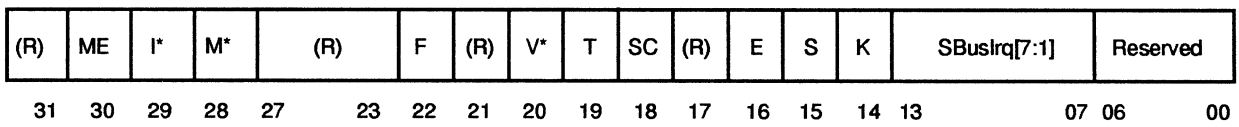


Figure 6-11 System Interrupt Pending Register

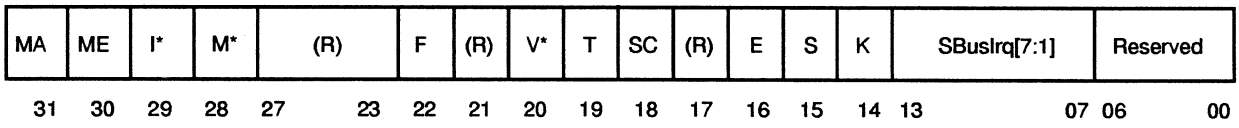


Figure 6-12 System Interrupt Target Mask Register (Read Only), and Mask Set and Mask Clear Registers (Write Only)

Field Definitions:

MA	Mask All interrupts. Writing a 1 disables all interrupts.
ME	Module Error (asynchronous fault)
I*	MSI (MBus-SBus Interface) interrupt
M*	EMC (ECC Memory Controller) interrupt
F	Floppy interrupt
T	Level 10 Counter/Timer
V*	Video interrupt
SC	SCSI interrupt
E	Ethernet interrupt
S	Serial Port interrupt
K	Keyboard/Mouse interrupt
SBus	Gives a direct indication of which SBus level interrupts are active.
(R)	Reserved- Read as 0s, writing has no effect.

* These bits are only active in SuperSPARC mode; when the 89C105 is in microSPARC mode (set via the configuration register), these System Interrupt Pending Register bits will always read as 0 and the corresponding System Interrupt Target Mask Register bits will have no effect (though writing the set/clear bits will still update the Mask register).

The Interrupt Target Mask Register occupies three addresses, one for reading the current state of the interrupt mask, and one each for setting and clearing mask bits. Writing a 1 to any defined bit field in the Mask Set register will disable that interrupt, and writing a 1 to the same field in the Mask Clear register will re-enable it. Interrupts are cleared by disabling and then re-enabling them. All pending interrupts are cleared, and all mask bits are set upon system reset.

The Mask All bit allows masking of Level 15 interrupts (considered non-maskable by the SPARC definition) allowing the boot firmware to establish a basic environment before receiving any such interrupts.

Interrupt Assignment and Priority

There are 15 levels of software-generated and/or externally generated interrupts supported by the 89C105 interrupt controller. Assignment and prioritization of these interrupts is performed by the interrupt logic. Interrupt assignments are as follows:

Table 6-24 Interrupt Level Assignments

Level	Sources
0	No Interrupts pending
1	SOFTINT.1
2	SOFTINT.2, SBus L1
3	SOFTINT.3, SBus L2, Parallel port
4	SOFTINT.4, SCSI
5	SOFTINT.5, SBus L3
6	SOFTINT.6, Ethernet
7	SOFTINT.7, SBus L4
8	SOFTINT.8, Video
9	SOFTINT.9, SBus L5
10	SOFTINT.10, System Counter/Timer
11	SOFTINT.11, SBus L6, Floppy
12	SOFTINT.12, Keyboard/Mouse, Serial Ports
13	SOFTINT.13, SBus L7
14	SOFTINT.14, Processor Counter/Timer
15	SOFTINT.15, Asynchronous HW Errors

Counter-Timers

The 89C105 features two programmable counter/timer channels, designed to provide a system timer and a single processor-specific set of timers. The System Counter is a 22-bit counter dedicated to the system timer function, and generates a level-10 interrupt upon time-out. The Processor Counter can be configured to behave as a 22-bit timer that generates a level-14 interrupt upon time-out, or to provide a real-time 54-bit counter for high resolution user performance analysis.

In the first mode, the timer is useful for OS kernel profiling. In the second mode, the timer can be loaded upon each entry into user mode, and saved on each exit. By mapping the counter as read-only for the user process, it provides "virtual" time, a measure of the context run time, which is useful for measuring application performance. It could also be loaded with a binary real time, which will then track precisely with the time-of-day.

Counter-Timers Block Diagram

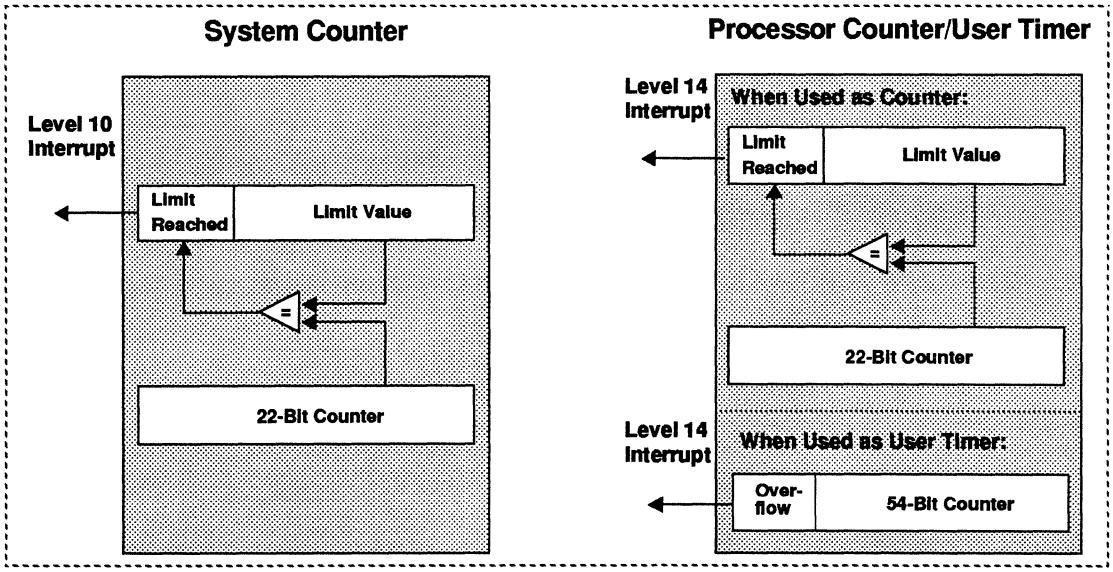


Figure 6-13 Counter-Timer Block Diagram

Counter-Timers Address Map

Counter/timer control registers are mapped as follows:

Table 6-25 Counter/Timer Address Map

PA[27:00]	Register	R/W
1D0 0000	Processor Counter Limit Register or User Timer MSW	R/W
1D0 0004	Processor Counter Register or User Timer LSW	R/W[1]
1D0 0008	Processor Counter Limit Register (non-resetting port)	W
1D0 000C	Processor Counter User Timer Start/Stop Register	R/W
1D1 0000	System Limit Register (Level 10 Interrupt)	R/W
1D1 0004	System Counter Register	R
1D1 0008	System Limit Register (non-resetting port)	W
1D1 000C	RESERVED	N/A
1D1 0010	Timer Configuration Register	R/W

1. Can be written as User Timer LSW, read-only as Counter Register.

Three addresses are associated with each counter: a count register, a limit register, and a pseudoregister that allows the limit to be loaded without resetting the count. The count and limit registers have the following format:

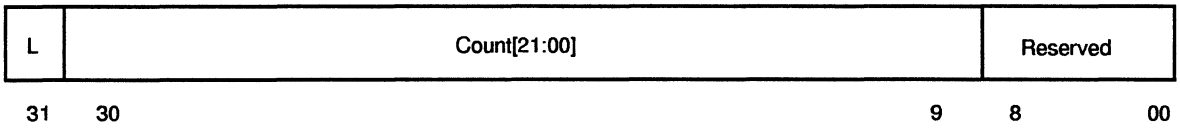


Figure 6-14 System Counter Register Field Definitions

Field Definitions:

L	Limit Reached
Reserved	Read as 0
Count	Current system count value

Each counter increments by one in bit position 9 every 500 nanoseconds. When a counter reaches the value in its corresponding limit register, it is reset to 500 ns (i.e. - 0x00000200), the limit-reached bit in both the counter and limit registers is set, and an interrupt is generated (if enabled) at Level 10 for the System Counter and level 14 for the Processor Counter.

The interrupt is cleared and the limit bits reset by reading the appropriate limit register. Reading the counter register does not change the state of the limit bit. Writing the limit register resets the corresponding counter to 500 nS (0x200).

The limit register can be loaded via the pseudoregister without resetting the count. If the count value is already higher than the new limit, the counter will proceed to count to its maximum value, then reset and count up to the new limit value before generating an interrupt. This allows alarm-clock, rather than time-tick, usage of the counter.

Setting a limit register to 0 causes the corresponding counter to free-run. Interrupts will be generated when the counter overflows, approximately every two seconds. All bits in all count and limit registers are cleared to 0 on reset.

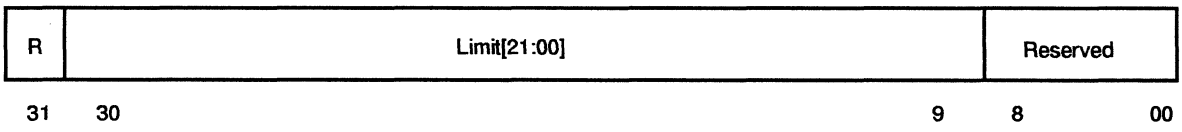


Figure 6-15 Counter/Timer Limit Register Field Definitions

Field Definitions:

R, Reserved	Reserved, read as 0.
Limit	Limit value to count to before setting interrupt and resetting counter.

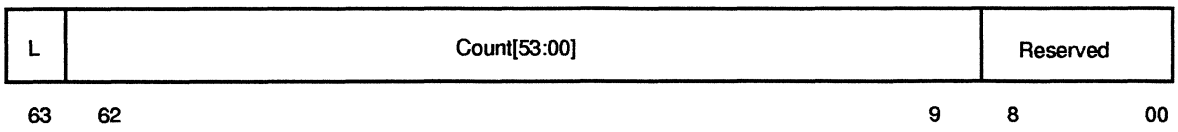


Figure 6-16 User Timer Count Register Field Definitions

Field Definitions:

L	Limit Reached
Reserved	Read as 0
Count	Current count value.

When the Processor Counter is configured to be a User Timer, it should be accessed only as a 64-bit word (to insure consistency between the LS- and MS- words). Although the counter is read/write, it is recommended that it be mapped read-only for user-mode access. The Limit bit is set any time the counter exceeds the maximum count value, and is cleared on any write to the register.

There is no interrupt associated with operation of the Processor Counter in User Timer Mode.

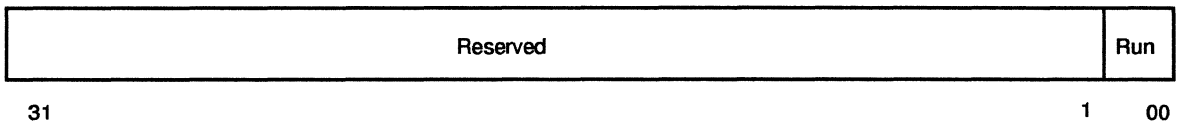


Figure 6-17 User Timer Start/Stop Register Field Definitions

Field Definitions:

Run	When set to 1, counting is enabled. When 0, frozen.
R, Reserved	Read as 0

The User Timer Start/Stop Register is provided to allow fast trap handlers to stop the User Timer blindly during time-critical code, without the necessity of reading and saving the count value. The timer must be restarted before reentering user state. A software flag must be maintained to indicate if the UT is in use, so that the fast trap handle can know that it must be restarted. This register has no effect if the Processor Counter is configured as a counter. All bits are cleared to 0 on reset.

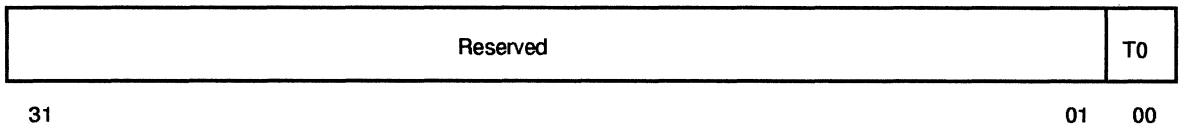


Figure 6-18 Counter/Timer Configuration Register Field Definitions

Field Definitions:

T0	When set to 1, the Processor Counter is configured as a User Timer.
R, Reserved	Read as 0.

Note that the System Counter cannot be configured as a User Timer. All bits are cleared to 0 on reset.

Chip Configuration Control

The 89C105 has several software-programmable options controlled by its configuration register. This register is located at address 180 000.

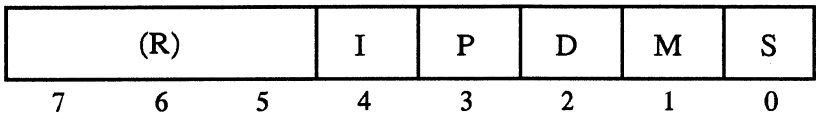


Figure 6-19 The 89C105 Configuration Register Field Definitions

Field Definitions:

(R)	These bits are unused. They read as 0; writing has no effect.
I	Modem Ring Indicate Interrupt Enable. When set to 1,the modem RI interrupt generation is activated (see also bit 1, and the description of the Modem Register). When cleared, no modem interrupt will be generated, regardless of the state of the M bit or the MSI_IRQ_ input.
P	Power Fail Detect Enable. When set to 1, a low on the PFD_ input will cause a module error (level 15) interrupt. The interrupt is visible (and clearable) in AuxIO register 2. When clear the PFD_ input is ignored.
D	Density Select Source (1 = 82077 density select, 0 = 82077 motor enable #2). This bit determines which signal drives the external density select pin (FPY_DENSEL). For the Teac tri-density (sw selectable) drive, this should be set to 0; for a standard PC drive (using an external 25-pin floppy cable) it should be set to 1, so that the 82077 can automatically control density selection between single and double density.

Field Definitions:

M — Modem Ring Select. This bit demuxes the MSI_IRQ_ input pin, to select whether it functions as a level 15 interrupt input from the MSI, or a modem ring indicator. When this bit is set to 1, a low on the MSI_IRQ_ input will cause a level 15 MSI interrupt. When it is cleared, a transition will cause a modem ring indicate interrupt (SBus level 5). Either a high or low transition can cause an interrupt in this mode, depending on the Edge Select bit in the Modem Register. The interrupt request is visible (and clearable) in the Modem Register. The unused input will be held in its inactive state.

S — SuperSPARC mode (1 = SuperSPARC, 0 = microSPARC). This bit determines the function of several muxed input pins (the 89C105 is extremely pin limited, so pins were not available to support all functions concurrently). The muxed pins are:

Table 6-26 microSPARC/SuperSPARC Muxed Pins

Pin	microSPARC Use	SuperSPARC Use
ser_rtxc_b	ser_rtxc_b_	emc_irq_
iu_error_	iu_error_	video_irq_

When in SuperSPARC mode, the microSPARC interrupts and signals will be forced to their inactive state; when in microSPARC mode, the SuperSPARC interrupts will be inactive (see the Interrupt Controller register for details of the SuperSPARC specific interrupts).

All of the 89C105 configuration bits are cleared to 0 by a system reset.

Diagnostic Messages

The Diagnostic Message Register is an 8-bit read/write register provided for diagnostic use. Accesses to this register have no effect, other than to change the value stored in it. The Diagnostic Message Register is non-volatile across resets (except power-up, where the register will come up in a random state).

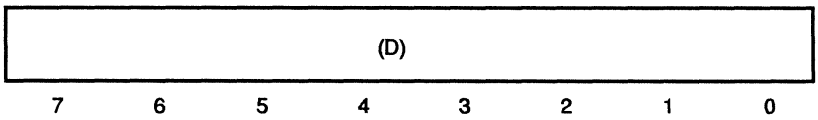


Figure 6-20 Diagnostic Message Register Bit Definitions

Field Definitions:

(D) — Diagnostic value. This value is read/writable, and will be preserved across resets.

All bits in the Diagnostic Message Register are unaffected by system reset.

Miscellaneous System Functions

The 89C105 contains two 8-bit Auxiliary I/O Registers: one dedicated to system power down control, and one used to support several hardware functions that don't fit well elsewhere. They are located at physical addresses 0x190 0000 (aux1) and 0x191 0000 (aux2).

LED/Floppy (Aux1) Register

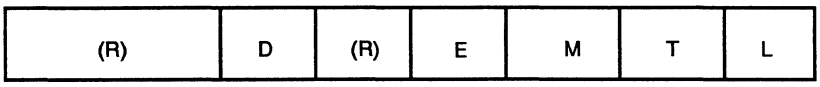


Figure 6-21 Auxiliary I/O Register 1 (0x190 0000) Field Definitions

Field Definitions:

(R)	AuxIO1[7:6,4] are unused. They should be masked out and their values discarded by software (they will always read as 0).
D	Floppy Density Sense (1=high density, read only). This bit directly reflects the state of the FPY_DENSENSE input pin.
E	Link Test Enable. This bit is directly reflected in the LINK_TEST_EN pin. It controls the AT&T 7213 LTE pin.
M	Monitor/Mouse Mux. This bit is directly reflected on the MON_MSE_MUX pin.
T	Terminal Count (1= TC). Writing a 1 will send a 4 SBus clock wide TC pulse to the 82077 floppy controller. This is self-clearing logic; it will always read as 0. Writing a 0 has no effect.
L	LED (1 = on, 0 = off.) This bit controls the system LED on the front panel.

All output bits (E, M, T, L) are cleared to 0 by a system reset. The input (D) is controlled by the corresponding chip pin (FPY_DENSENSE). The unused bits ([7:6, 4]) are unaffected by writes and will always read as 0.

Power Down Control (Aux2) Register

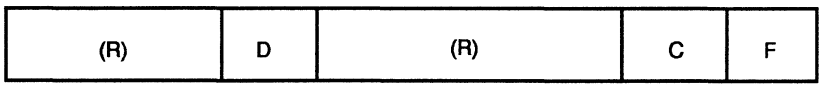


Figure 6-22 Auxiliary I/O Register 2 Field Definitions

Field Definitions:

(R)	AuxIO2[7:6,4:2] are unused. They should be masked out and their values discarded by software ([7:6, 4] will always read as 0; [3:2] are read and write-able but the values have no meaning or effect).
D	Power Failure Detect (1 = power fail). When the power fail detect signal from the power supply is asserted (low), this bit is set and a "module-error" interrupt will be generated (this is a level-15 interrupt). This bit is cleared by writing a 1 to bit 1 (of this register), or by disabling PFD in the Config register. It should be noted that the PFD* input is ignored if disabled in the Config Register.
C	Clear Power Fail Detect Int (1 = clear). This bit will clear the interrupt generated by PFD_ and the corresponding register bit (bit 5 above). Writing a 0 has no effect.
F	Power Off (1 = off). This bit is simply reflected in the power_off output pin. Setting it to 1 will turn the power supply off.

All AuxIO2 bits are cleared to 0 on system reset.

Modem Register

The 89C105 can support the RI (Ring Indicate) bit output of a modem directly when configured for Modem use (see the Configuration Register definition). This mode uses the MSI_IRQ_ input pin for modem RI sensing, so it is not available in SuperSPARC mode. When the Modem mode and the Modem interrupt are enabled in the Configuration Register, the 89C105 will generate an SBus level 5 interrupt on RI transitions.

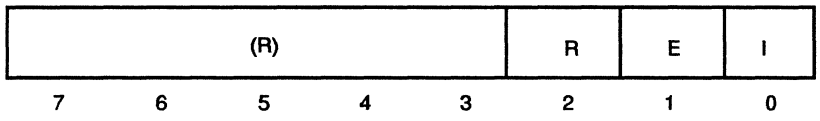


Figure 6-23 Modem Register Field Definitions

Field Definitions:

(R)	Modem bits [7:3] are unused, and will always read as 0. Writing has no effect.
R	RI pin. This pin directly reflects the state of the MSI_-IRQ_ pin (which is used for modem RI when in Modem mode). If the pin is low, then this bit will be 0.
E	Edge Select. This bit selects which edge of RI causes an interrupt. When cleared to 0, a 1->0 transition on the MSI_IRQ_ pin causes an interrupt. When set to 1, a 0->1 transition causes an interrupt. Toggling this bit after receiving the first edge of an RI pulse will allow one to get interrupts on both edges of RI.

Field Definitions:

I	Modem RI Interrupt. This bit is set to 1 if a modem RI interrupt is pending, and an SBus level 5 interrupt is set. Writing a 0 to this bit clears the interrupt.

Bits [1:0] (E,I) are cleared to 0 by a system reset. The input (R) is controlled by the corresponding chip pin (MSI_IRQ_). The unused bits ([7:3]) are unaffected by resets or writes and will always read as 0.

Test Block

This section describes the goals and implementation of the testability features implemented in the 89C105. These features have been incorporated to provide a structured test approach to both device fabrication testing and board-level testing and debug.

The 89C105 contains an IEEE JTAG 1149.1 compliant test controller and boundary scan architecture. All mandatory instructions are supported and this document contains the chip specific boundary scan information. The 89C105 also contains internal test logic and reserved instructions. The basic description of this logic is described below but not supported.

JTAG Scan Access

The goals for the 89C105 testability are to provide for high stuck at fault coverage at both the IC and board level. This is provided by the incorporation of an IEEE 1149.1 (JTAG) compatible TAP controller and boundary scan, which in conjunction with modular broadside access modes provides access to each of the major functional blocks on the I/O chips through either full scan or boundary scan (in the case of the ASFs). The ASFs are tested during device fabrication by a full broadside pin mode that provides direct access to all ports of each ASF from the device pins. This allows standardized functional test patterns to be applied directly to each ASF. At the board-level, the JTAG compatible boundary scan provides for complete access to PCB interconnect, including die to package bonding.

Block Access Modes

Diagnostic multiplexing between the pad ring and the internal macros is configurable into four different modes: Normal Mode, in which the device operates as required in the system; TBLK1 Mode, for scanned logic, in which all the ports to the scanned logic are accessible via scannable elements. In addition, the internal scan chain of the block is connected in series with the boundary scan chain, and the partition scan chain (if one is required) to form a complete scan path for access to all state and primary inputs of the block. TBLK2 and TBLK3, for ASF cores, in which each block is presented to the pins of the chip as if it where a stand-alone device.

Tristate Pin Function

All output pins of the device are tristate-able, controlled by elements in the boundary scan chain, to support manufacturing test. At power-up and in normal operation of the system this function is disabled by the TRSTB JTAG pin being held in the active low state.

Block Diagnostic Modes

TBLK1 (Internal Scan) Diagnostic Mode

The following figure illustrates the operation of the TBLK1 diagnostic mode. In this mode, the test logic is configured to connect every primary input to the Q-output of a scannable flip-flop and every primary output to the D-input of a scannable flip-flop. In addition, every flip-flop inside the block is configured into a single scan chain, known as the internal, or "iscan" chain.

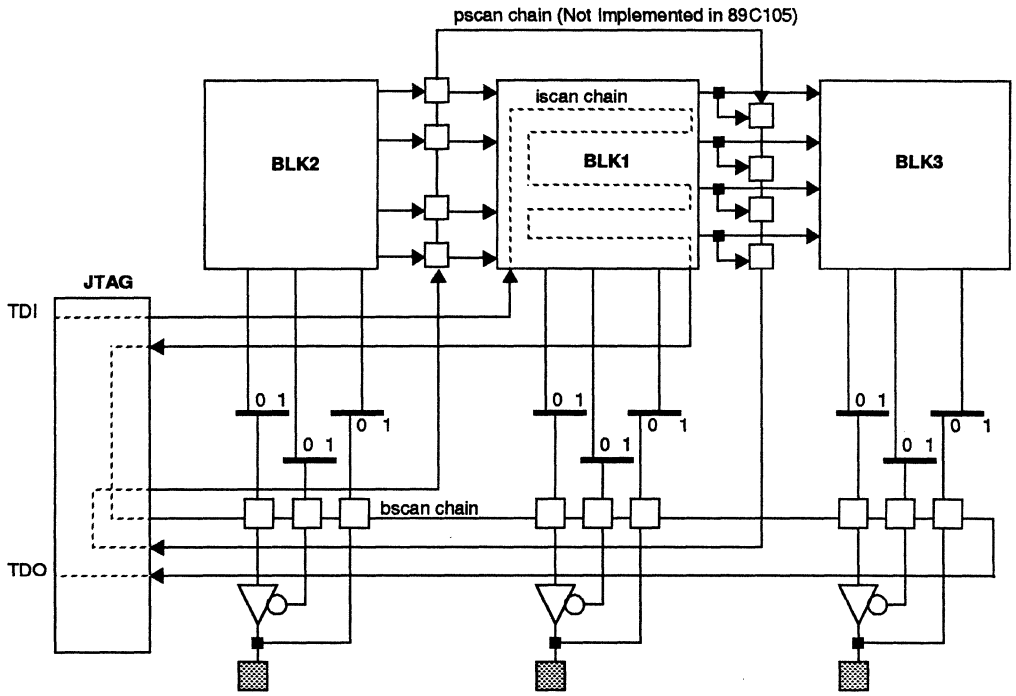


Figure 6-24 TBLK1 (Internal Scan) Diagnostic Mode

TBLK2/TBLK3 Diagnostic Mode

The following figure illustrates the operation of the TBLK2(TBLK3) diagnostic mode. In these modes the test logic is configured to connect internal inputs and outputs to BLK2 (BLK3) to pins normally assigned to BLK1 or BLK3(BLK2). Since these blocks are non-scannable, the only function of the JTAG controller in this mode is to configure the multiplexor logic into this mode. Hence the scan datapath is placed in BYPASS mode

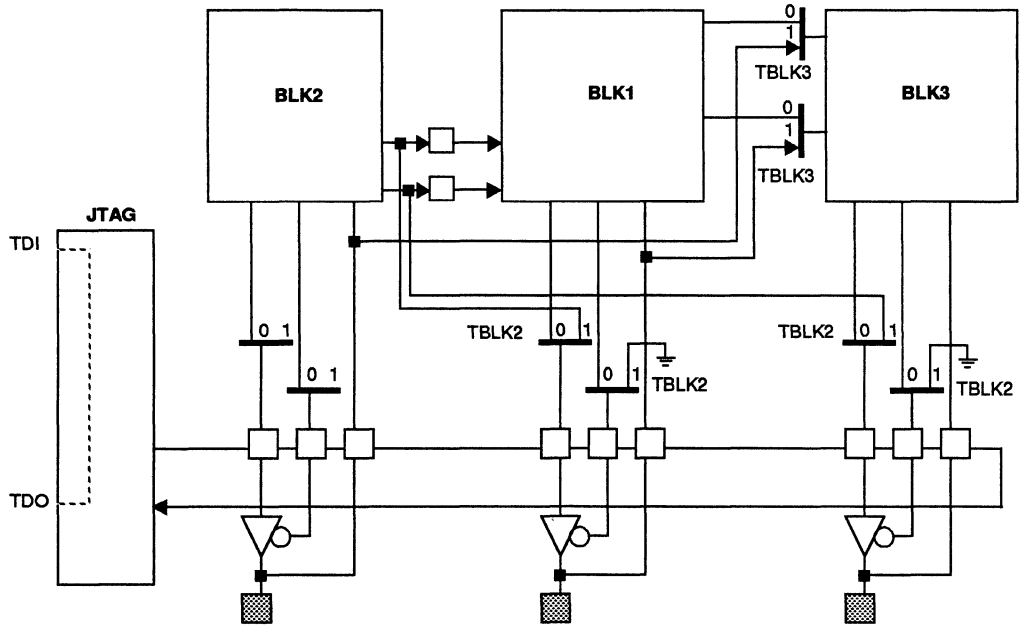


Figure 6-25 TBLK2/TBLK3 Diagnostic Mode

Outputs of the block that normally connect to BLK1 are multiplexed into the chip outputs of the other blocks, configured by the TBLK2(TBLK3) mode signal. Inputs to BLK2 (BLK3) are multiplexed with inputs from the other blocks. Figure 6-21 shows how the outputs of BLK2 and the inputs of BLK3 are configured.

Other JTAG test modes (TBLK2_BS and TBLK3_BS) are provided that operate identically except that the scan data path is configured to pass through the boundary chain. This allows application of the broadside test vectors to the blocks using the boundary chain to drive primary inputs and sample primary outputs in a pseudo-static manner, i.e. it does not directly support complex edge relationships between inputs. Instead these vectors must be "exploded" into multiple boundary scan vectors.

JTAG Controller

The JTAG controller contains the following elements:

- NCR Tap controller
- Scan Datapath including instruction register, bypass register, and ID register
- Clock control register and state machine

The following figure shows a simplified block diagram of the JTAG controller. It has been partitioned into two main functional areas: Scan Datapath and Scan Control Logic:

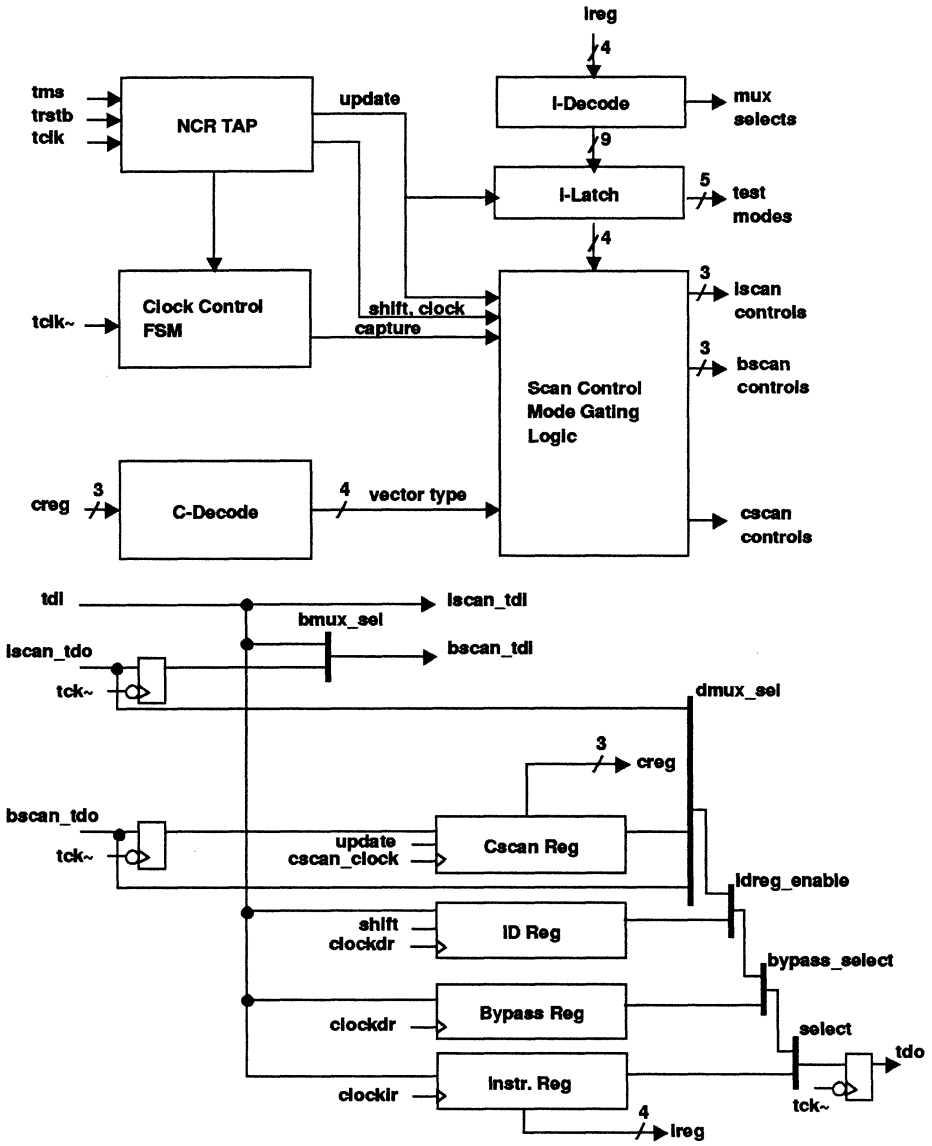


Figure 6-26 JTAG Controller Block Diagram

The NCR tap controller is an implementation of a reference 1149.1 tap state machine.[1] It is connected directly to the test access port on the 89C105 (TCK,TMS, TRSTB) and generates the basic scan controls (clock_dr, clock_ir, reset_l, select, shift_dr, shift_ir, update_dr, update_ir) which are used to control the scan architecture.

The NCR TAP implementation has been modified slightly to also make available the TAP state for use by supplemental state machines. The NCR state machine implements the reference state diagram described by the 1149.1 specification[2] and will not be reproduced here. The state coding is shown in the following table:

1. IEEE Std. 1149.1-1990 chapter 5.
2. IEEE Std. 1149.1-1990 page 5-1.

Table 6-27 State Assignments for NCR TAP Controller

Controller State	State[3:0]
Exit2-DR	0
Exit1-DR	1
Shift-DR	2
Pause-DR	3
Select-IR-Scan	4
Update-DR	5
Capture-DR	6
Select-DR-Scan	7
Exit2-IR	8
Exit1-IR	9
Shift-IR	A
Pause-IR	B
Run-Test/Idle	C
Update-IR	D
Capture-IR	E
Test-Logic-Reset	F

The instruction register for the 89C105 is a four bit register comprised of simple scannable elements. When the TAP state machine issues a reset signal this register is initialized to the IDCODE (1110) instruction. The parallel inputs of the instruction register are not used to load design-specific information and are tied-off to logic 0.

The 4-bit output of the instruction register is followed by an instruction decode stage which decodes up to 16 unique instructions. Not all of these are used by the 89C105 but are given mnemonics for completeness. The following table lists these mnemonics and the instruction value that corresponds to them:

Table 6-28 Decoded JTAG Instructions

Value	Mnemonic	Description
0000[1]	EXTEST	Boundary scan board interconnect test.
0001	SAMPLE	Boundary scan sample/preload.
0010	TBLK1	BLK1 ATPG scan test mode (Internal+-Boundary+Clock chains).
0011	TBLK2	BLK2 broadside test mode (Bypass).
0100	TBLK3	BLK3 broadside test mode (Bypass).
0101	SCANTOOL	BLK1 no-capture scan test mode (Internal chain).
0110	PSCAN	Reserved for partition scan (if implemented, otherwise Bypass).
0111	INTEST	Boundary scan capture of internal I/O.
1000	TBLK2_BS	BLK2 boundary scan test mode.
1001	TBLK3_BS	BLK3 boundary scan test mode.
1011	TPSCAN	Reserved for BLK1 tester partition scan mode (if implemented, otherwise Bypass). Other BLK1 pins controlled by broadside tester.
1100	BPSCAN	Reserved for BLK1 boundary partition scan mode (if implemented, otherwise Bypass). Other BLK1 pins controlled by boundary scan.
1101	ZMODE0	General purpose test mode.
1110	IDCODE	Device ID register.
1111[2]	BYPASS	Bypass mode.

1. Required instruction.
2. Required instruction.

The scan controls decoded by these instructions configure the scan datapath, the test multiplexors and control the clocks and pseudo clocks for the test mode in progress.

Instruction Decode

The I-Decode logic converts the 4-bit instruction register contents into decoded signals that control mux selection in the scan datapath and test mode configuration in the ASF blocks. Nine of these signals are latched by the I-latch to provide glitch free values on these signals which are updated during the IR-Update state.

Clock Control FSM

The clock control finite state machine monitors the state output from the NCR TAP controller and determines when it is necessary to insert the capture clock and/or pseudo clocks required to support ATPG stimulus application to BLK1. The clock gating is designed such that the boundary clock is guaranteed to be asserted at all elements of the boundary scan chain before it is applied to either the clock or pseudo clock (set/reset) inputs to the BLK1 internals.This requirement is present due to the

fact that ATPG vectors have an assumed order in which stimulus is applied to the circuit and state or primary outputs are captured. The clock control state machine in the 89C105 has been verified to support the requirements of TestScan ATPG from Cadence Design Systems, Inc. although it may function equally well with other ATPG systems.
The assumed sequence of operations required for ATPG pattern application is shown below.

1. Stimulate pins - boundary and pscan chains shift/update sequence.
2. Stimulate shift register/latches - internal scan chain shift in.
3. Measure pins - boundary and pscan chain capture sequence.
4. Pulse clocks/pseudo clocks - internal chain clock/set/reset.
5. Measure shift register/latches - internal scan chain shift out.

The Clock Control FSM has been designed to support this event ordering in a single continuous shift-update-capture-shift sequence. in TBLK1 mode the scan chains within the 89C105 are concatenated into a single chain containing internal, boundary and clock control scan chains. Hence after an initial shift-update sequence, the requirements of (1) and (2) have been met. The Capture-DR state is then used to measure the state of the primary outputs of BLK1 by issuing a clock to the boundary with the shift control not asserted. A delayed version of the clock (or update pulse in the case of the pseudo clocks) is then used apply clock, set or reset to the internal scan chain to implement the internal chain capture. This occurs only when indicated by the value of the "capture" output from the clock control state machine.

Clock Control Register

The other three bit positions in the clock control scan chain are transferred to the clock control register during a DR-update sequence, where they are decoded by the C-Decode logic to specify which vector class the following capture sequence belongs to out of the following categories:

1. Shift only, no capture.
2. Capture scan chain, (i.e. shift high during capture clock).
3. Normal clocked vector.
4. Set vector, no clock.
5. Reset vector, no clock.

Since the last two categories are only required when the logic under test contains asynchronous sets or resets, they are not required for the 89C105.

Mode Gating Logic

The decoded vector type information is combined with the clock control FSM information and the primary scan controls from the NCR TAP controller and instruction register decodes to generate control signals for each of the four scan chains with the 89C105: iscan, bscan, pscan (if present) or cscan. These signals are buffered and distributed throughout the device to the various chain elements. Since the 89C105 does not require a partition scan chain for its final implementation, these controls have been deleted.

Scan Datapath

The scan datapath within the JTAG Controller contains the chain configuration logic, implemented as a series of multiplexors; inter-chain flops to guarantee hold margins; the Cscan register; JTAG compliant ID, BYPASS and IR shift registers and the TDO output multiplexors and flop. This datapath, like the external scan chains and test logic is controlled by the scan control logic described above. The only variation from a more conventional IEEE 1149.1 implementation is the ability to configure the scan chain into various different modes based on the instruction type. The use of the hold flops, clocked by TCK~ is simply an implementation detail to reduced the effects of clock skew between the separate scan chains.

The 89C105's scan datapath does not include a partition scan chain, as in its final implementation this functionality has been incorporated into the internal, or "iscan" chain to facilitate physical implementation of the device. The elements of the embedded partition scan chain are therefore controlled by the same datapath controls as the existing iscan chain elements.

The table below gives the lengths of the various scan chains that comprise the 89C105's scan datapath.

Table 6-29 The 89C105 JTAG Chain Lengths

Chain Name	Number of Elements
BYPASS	1
I.D.	32
Instruction Register	4
Internal	549
Boundary	207
ATPG	759

Performance

The design as implemented in NCR's VS700H 0.95um (drawn) standard cell library has been verified to operate at a 5 MHz scan rate.

The JTAG controller occupies approximately 700 gates, and the scan overhead for the simple multiplexed flop scan element that it supports is estimated at about 10% from a gate count perspective, 5% in total area overhead.

Functional Timing Diagrams

This section contains cycle diagrams of the 89C105 SBus accesses.

EBUS Timing Diagrams

EPROM Read

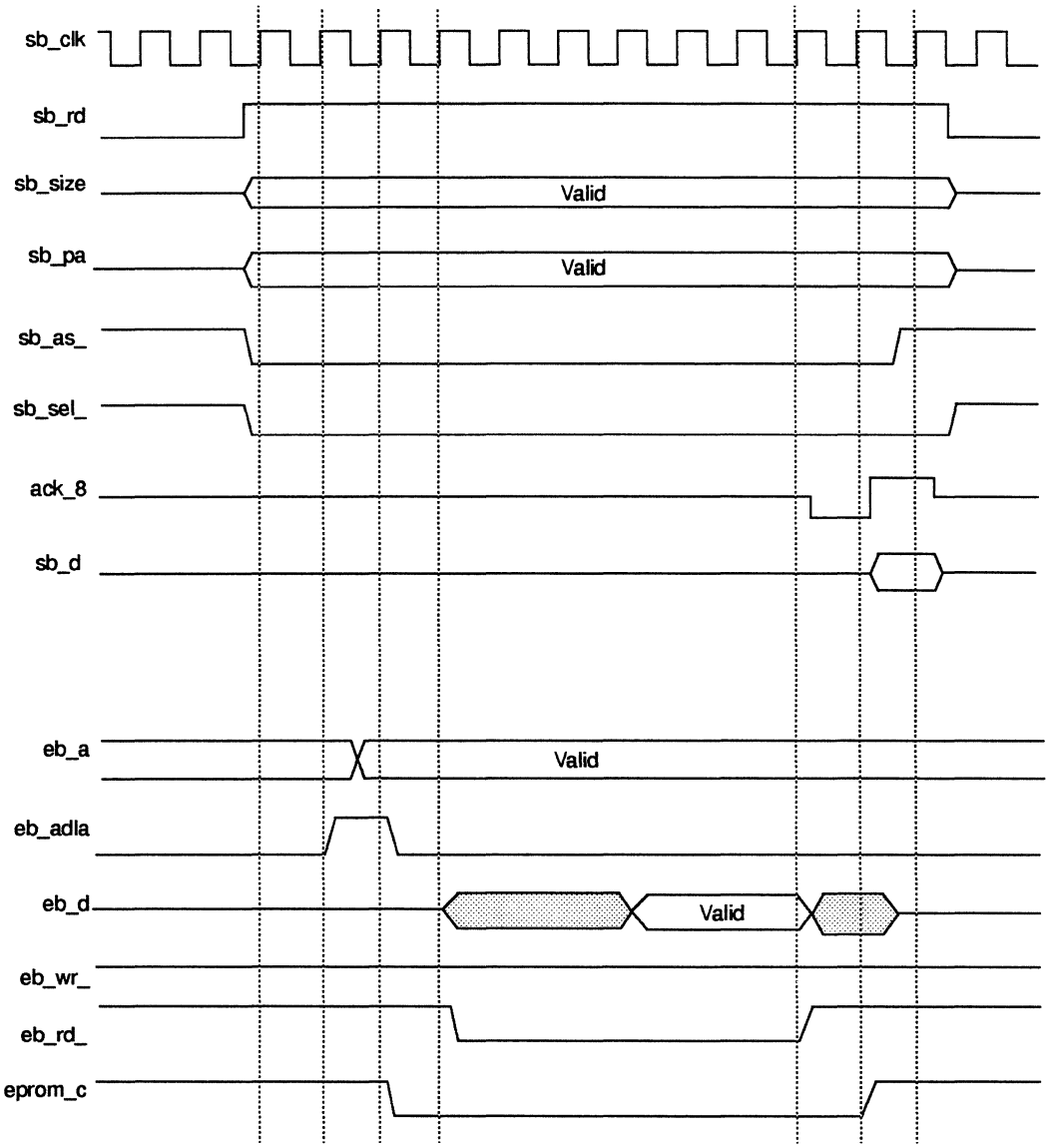


Figure 6-27 EPROM Read

TOD/NVRAM Read

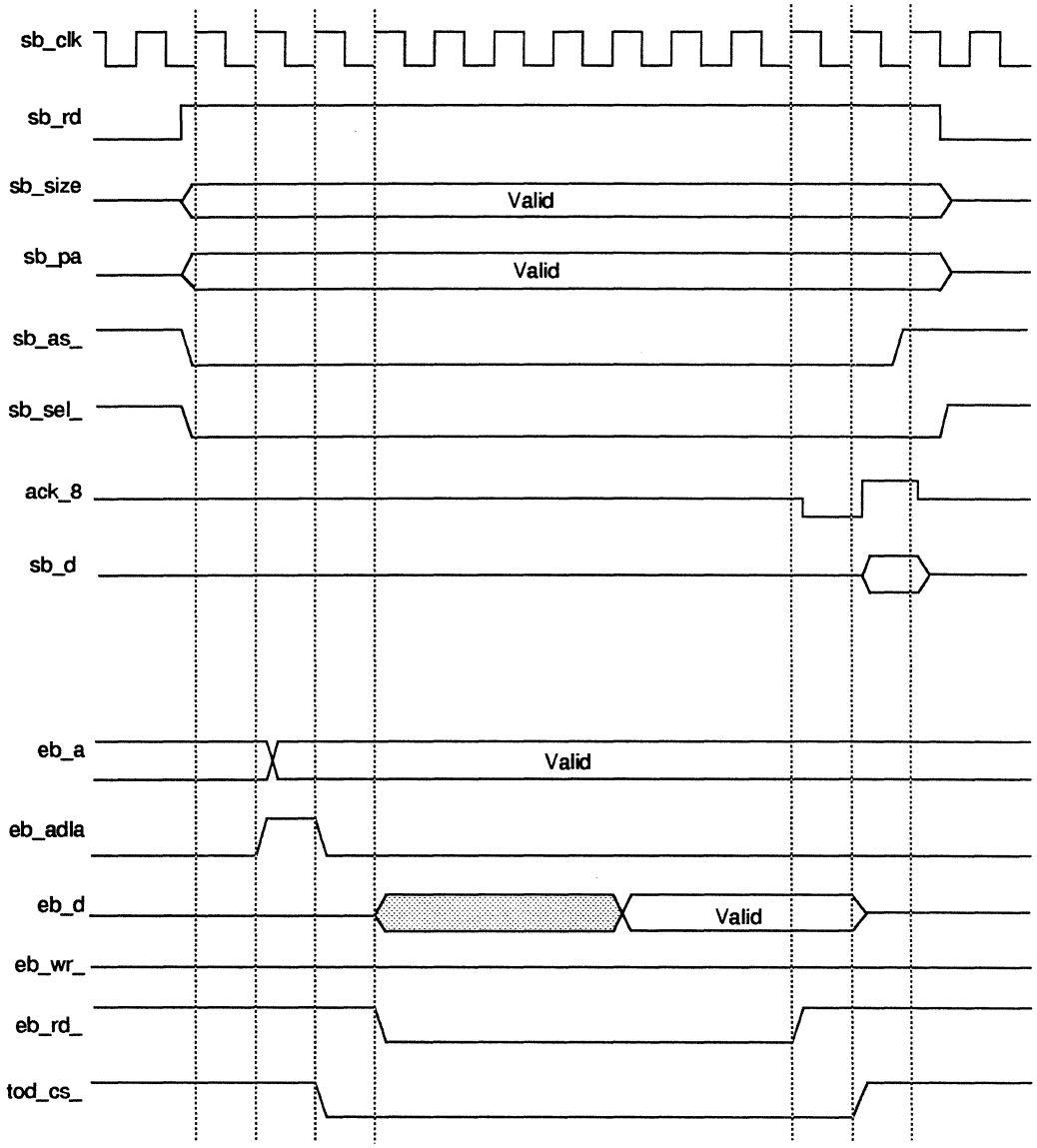


Figure 6-28 TOD/NVRAM Read

TOD/NVRAM Write, Showing Buffered EBus Cycle

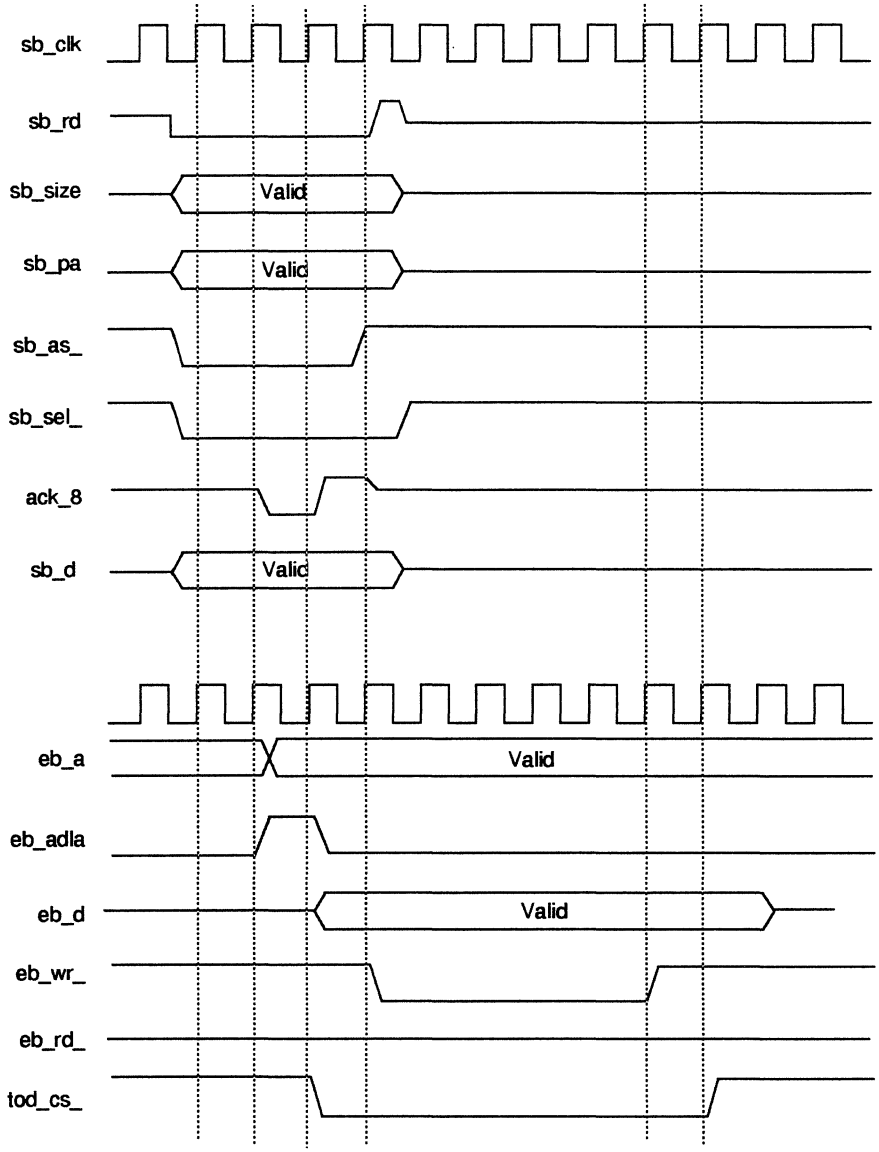


Figure 6-29 TOD/NVRAM Write, Showing Buffered EBus Cycle

TOD/NVRAM Write/Write, Showing Buffered EBus Cycle

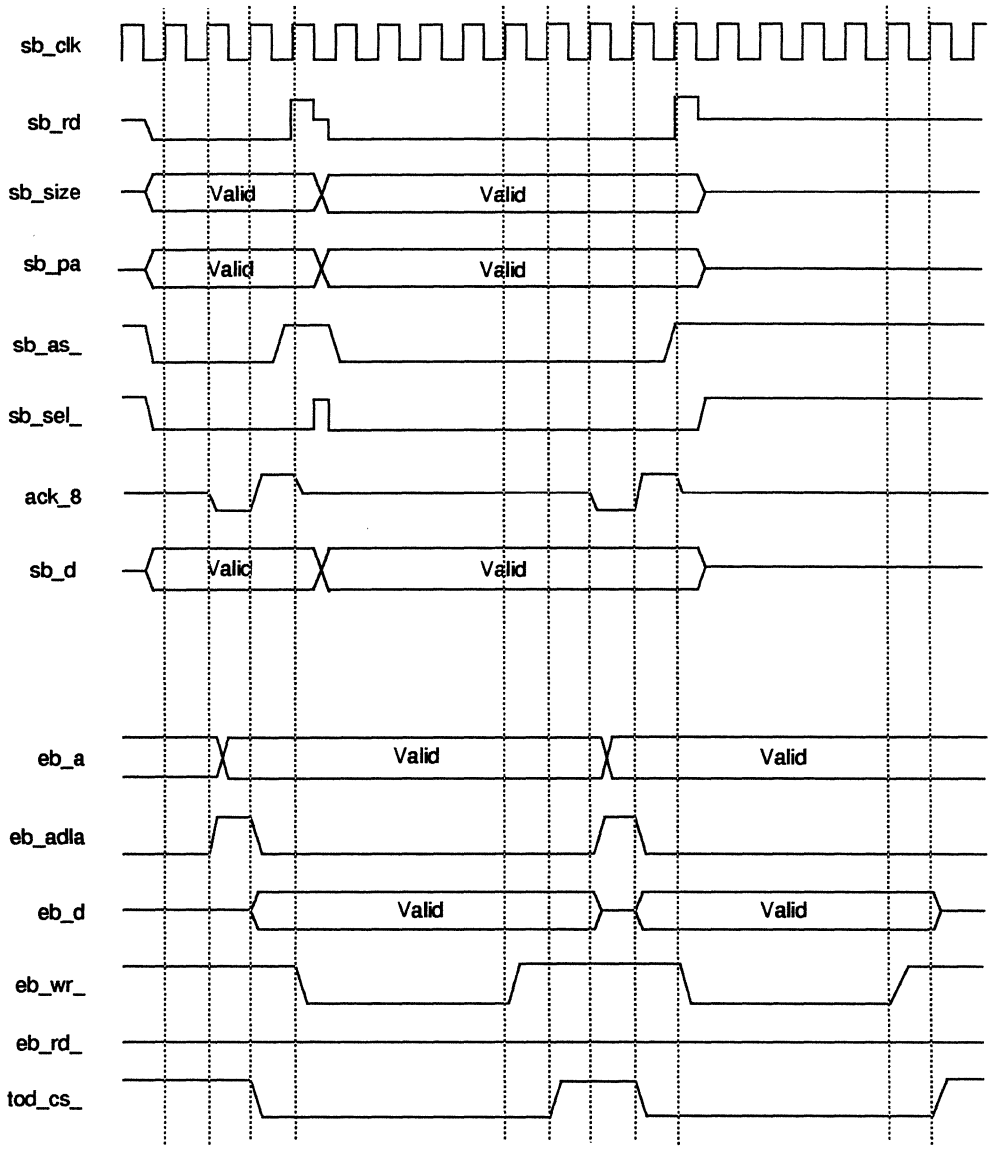


Figure 6-30 TOD/NVRAM Write/Write, Showing Buffered EBus Cycle

Generic Port Read

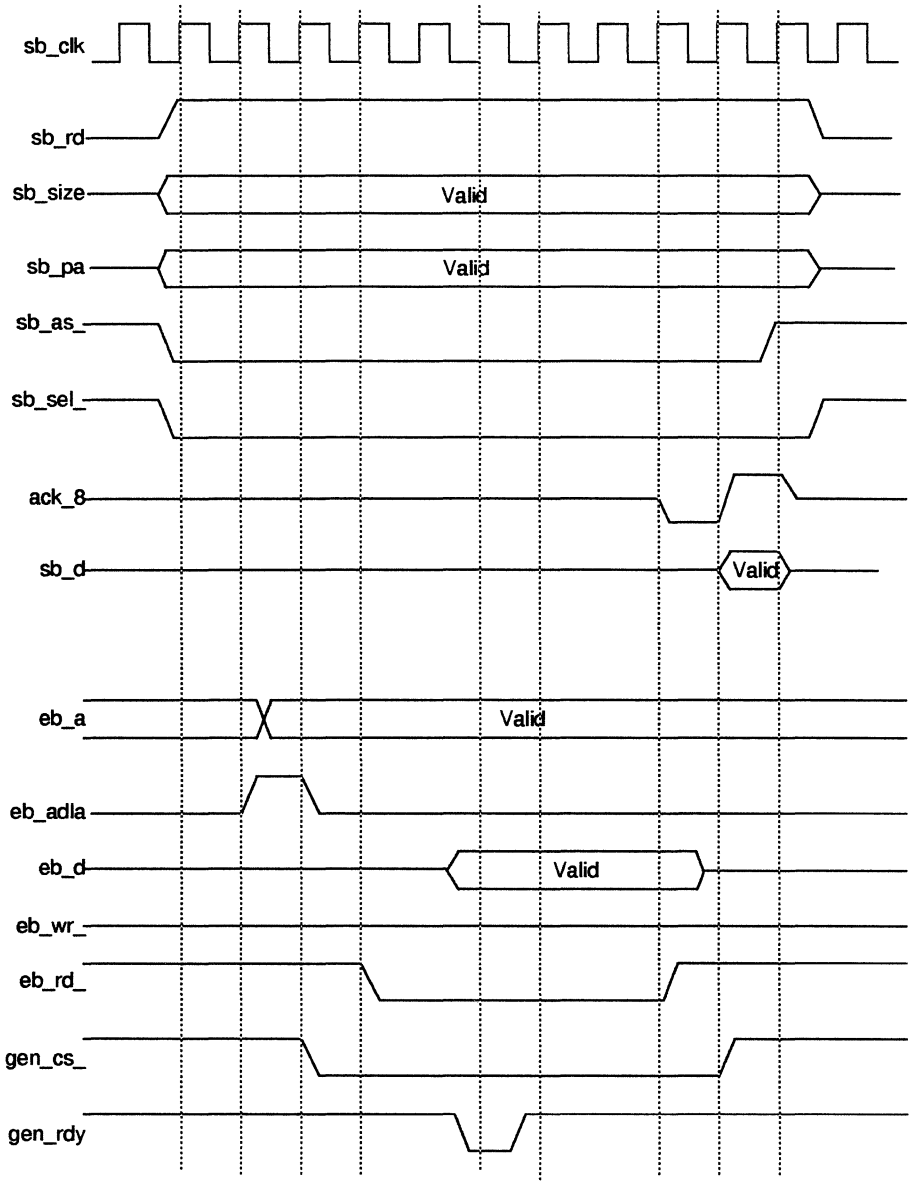


Figure 6-31 Generic Port Read

Generic Port Write

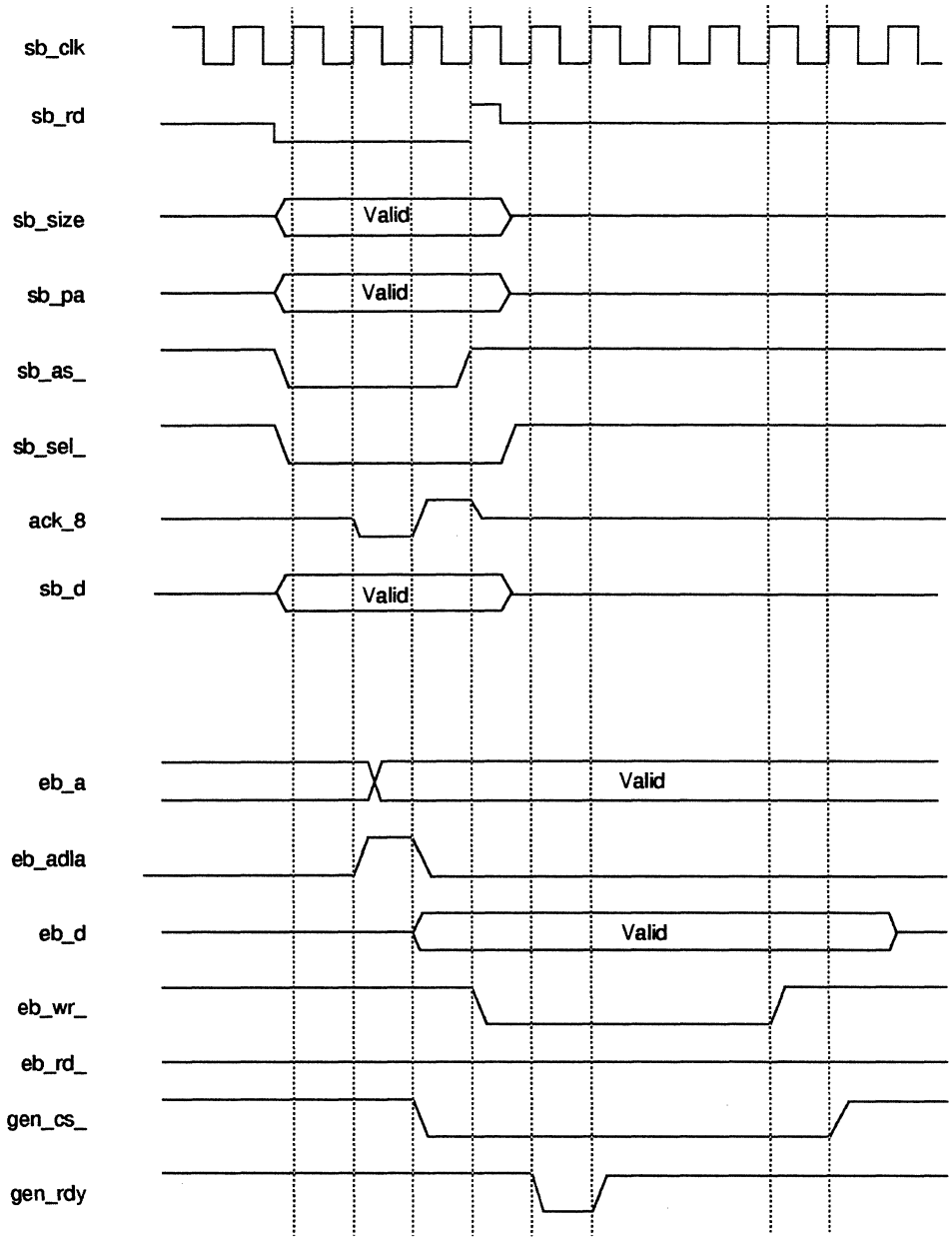


Figure 6-32 Generic Port Write

Generic Port Time-Out

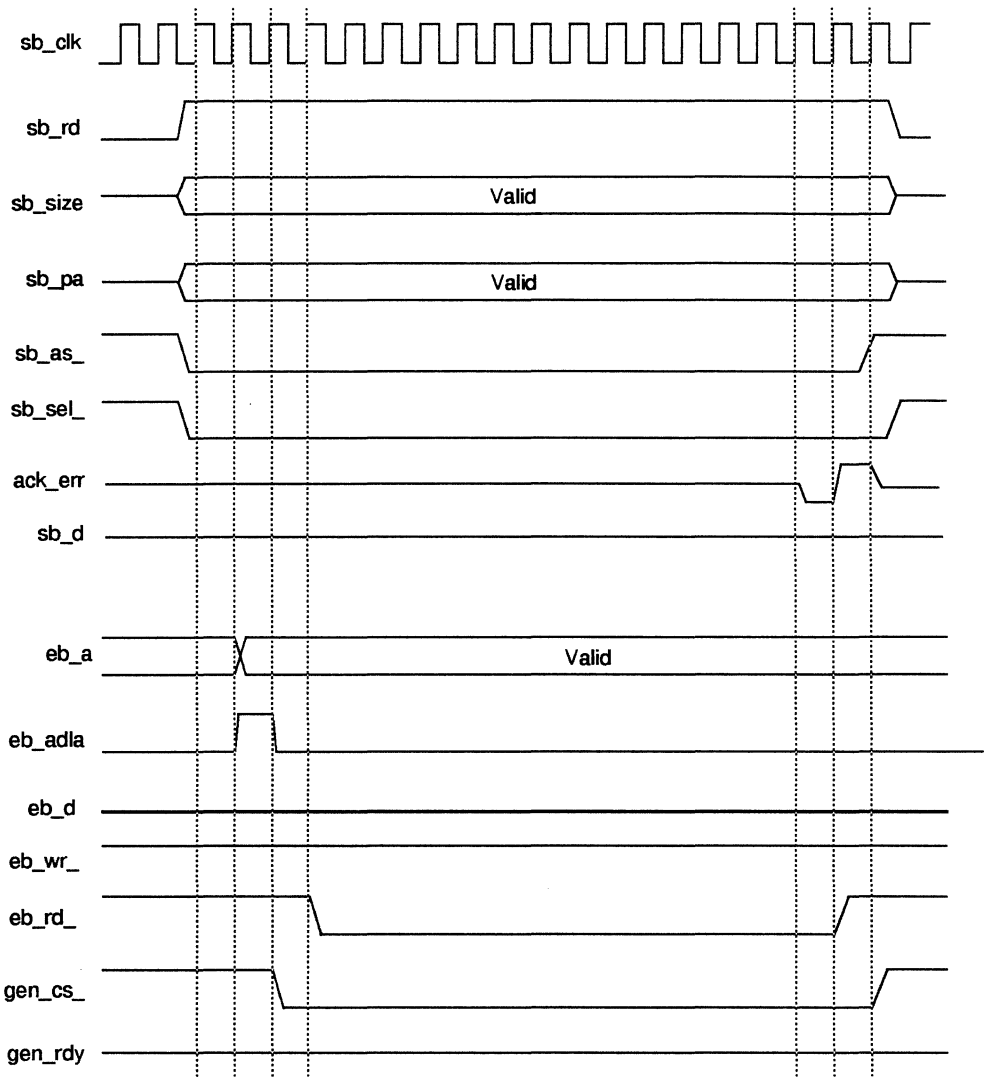


Figure 6-33 Generic Port Time-Out

Serial Ports/Keyboard/Mouse Write/Read Showing Hardware Holdoff

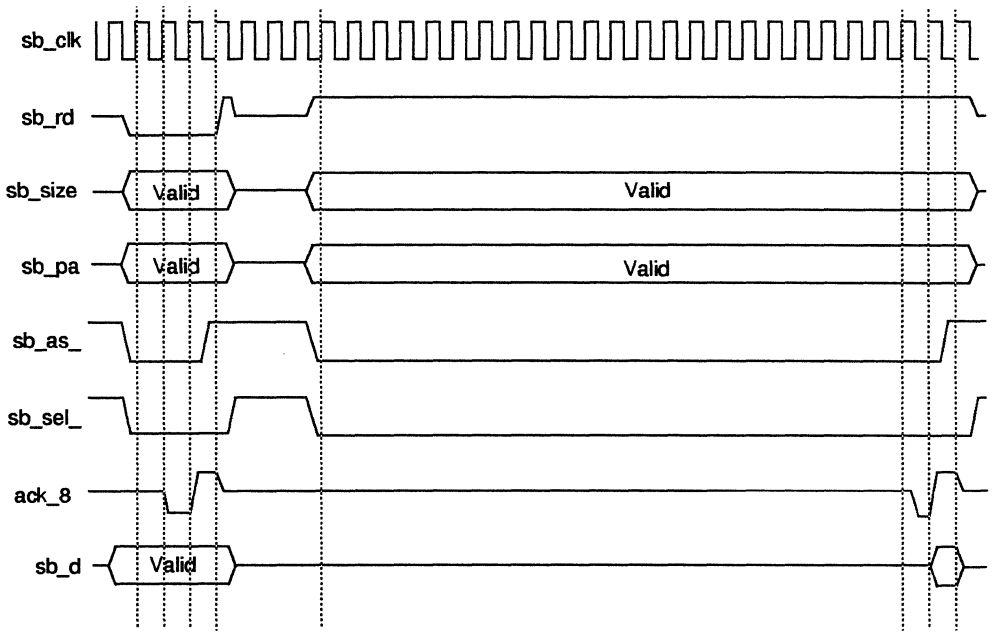


Figure 6-34 Serial Ports/Keyboard/Mouse Write/Read, Showing Hardware Holdoff

Floppy Read/Write

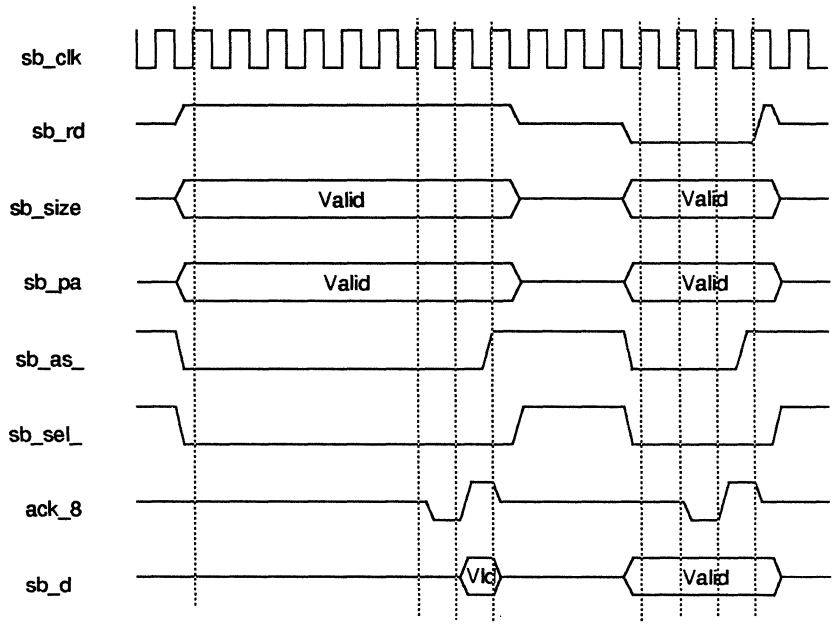


Figure 6-35 Floppy Read/Write

DMR/AUXIO/Configuration Register Read/Write

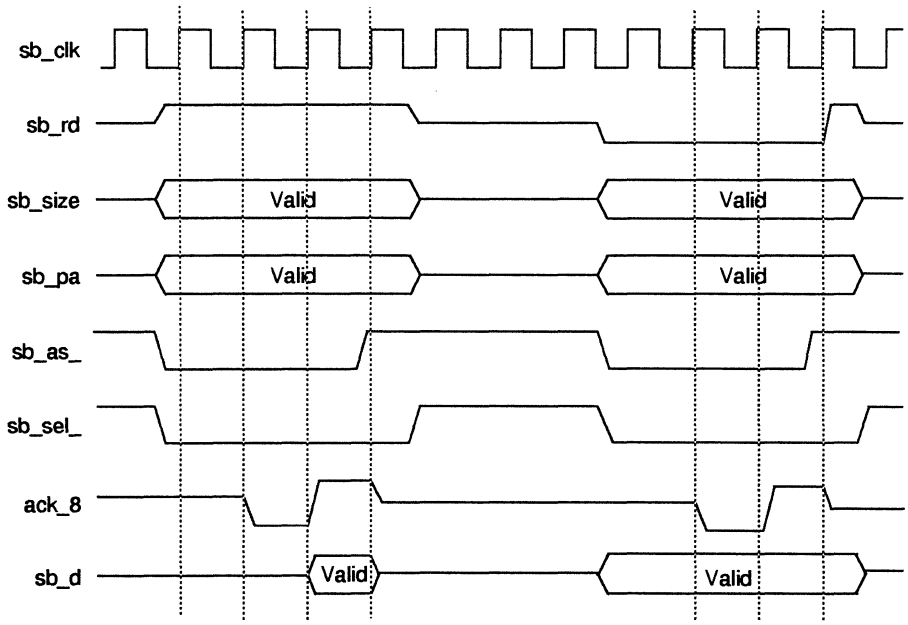


Figure 6-36 DMR/Aux IO/Configuration Register Read/Write

Counter/Timer Read

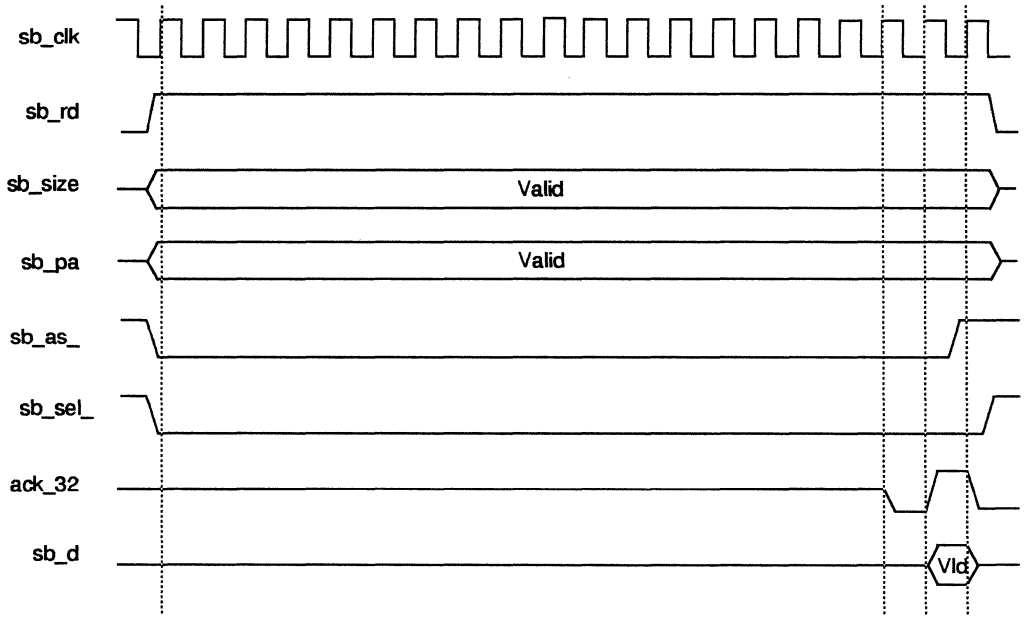


Figure 6-37 Counter/Timer Read

Counter/Timer Two-Word Burst Read

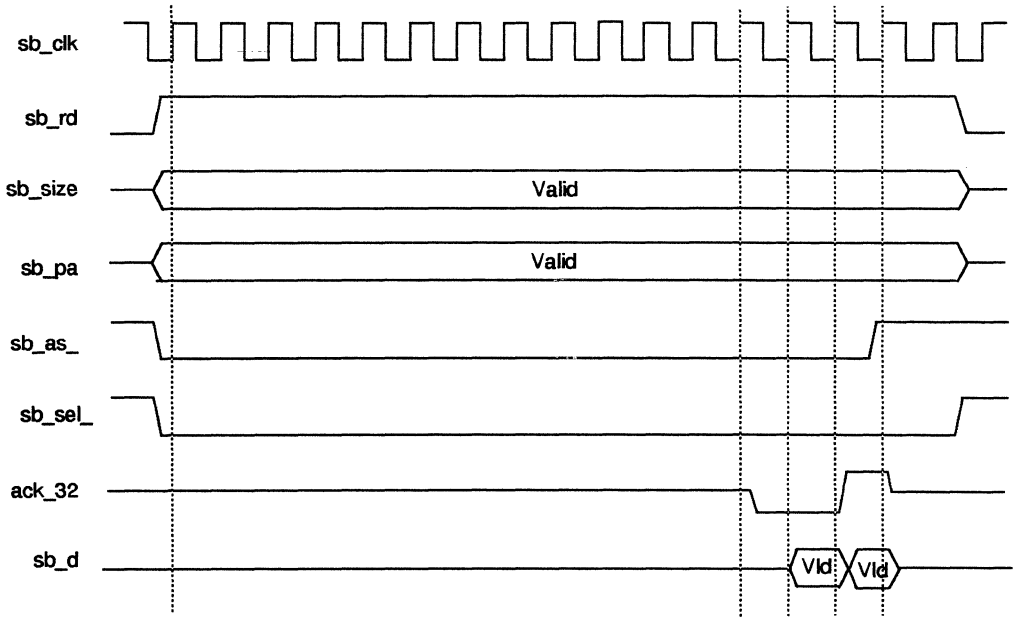


Figure 6-38 Counter/Timer Two-Word Burst Read

Counter/Timer Write

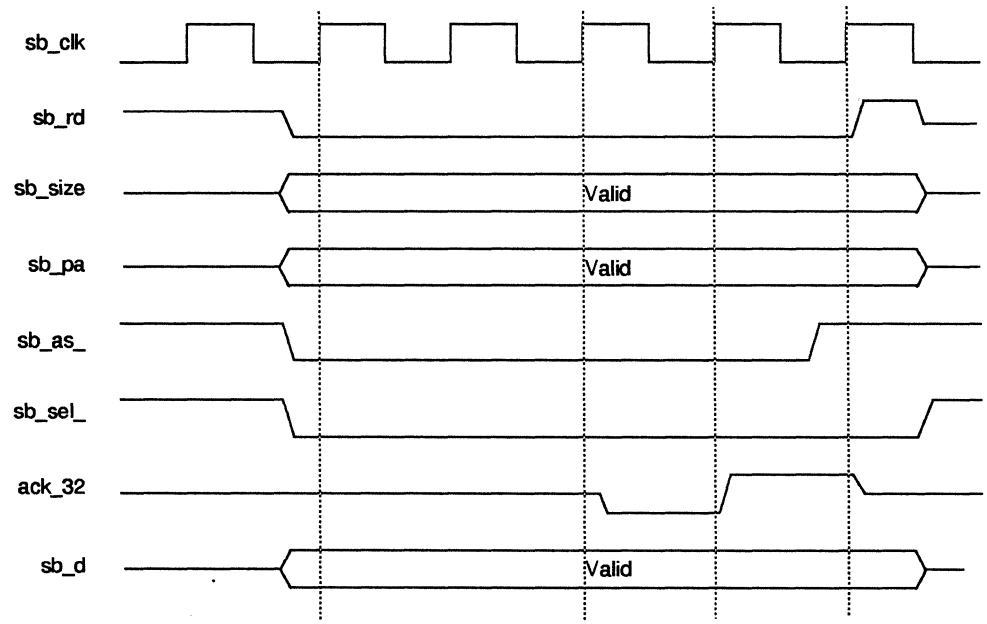


Figure 6-39 Counter/Timer Write

Counter/Timer Two-Word Burst Write

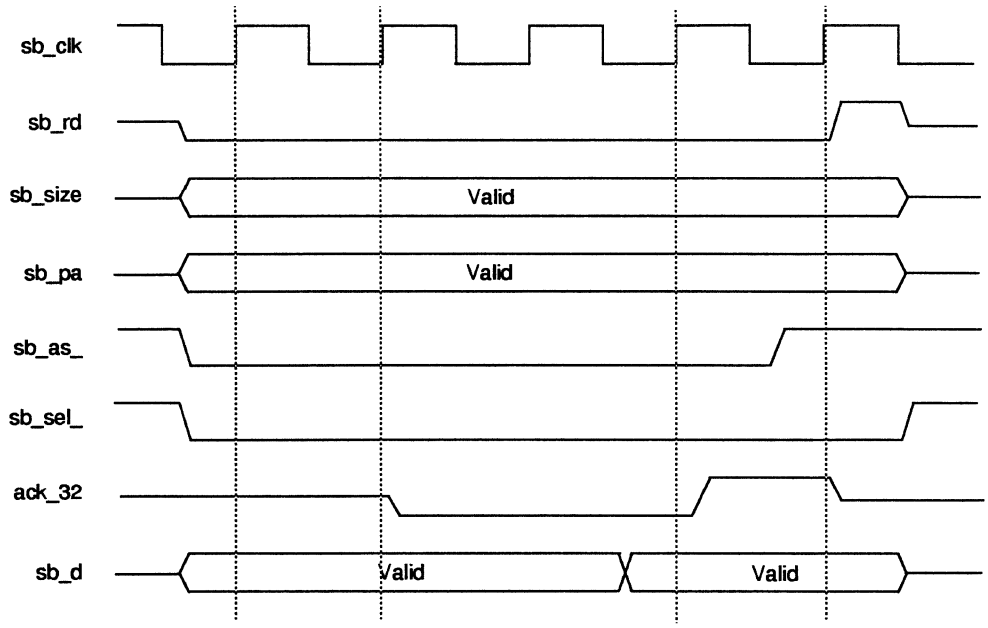


Figure 6-40 Counter/Timer Two-Word Burst Write

Interrupt Controller/SCSR Read/Write

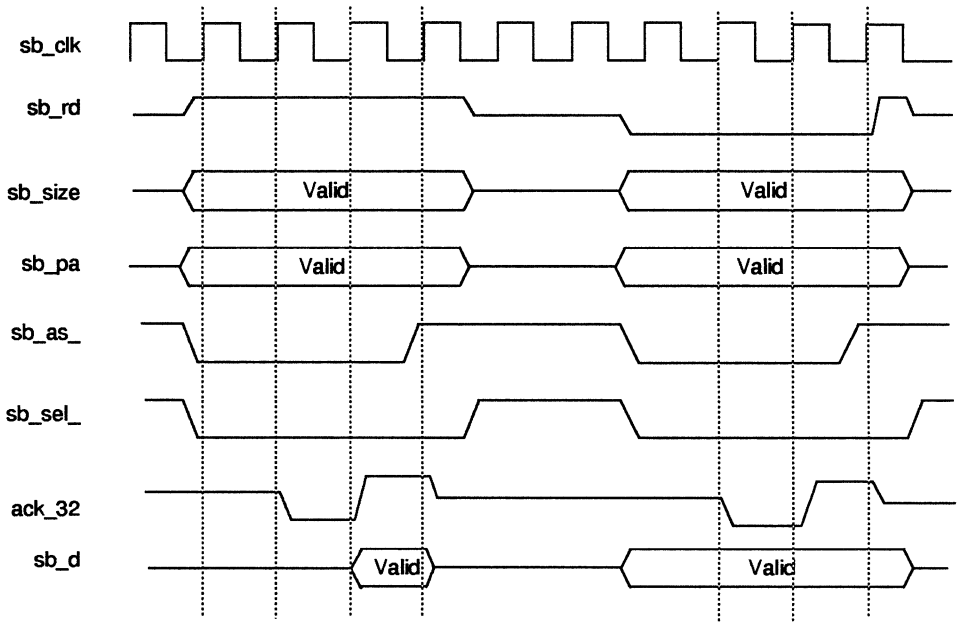


Figure 6-41 Interrupt Controller/SCSR Read/Write

Electrical Considerations

This section defines the following electrical specifications for the 89C105:

- Absolute and recommended operating conditions
- DC characteristics
- AC characteristics
- Power consumption

The 89C105 is implemented in the NCR VS700H process, and normal handling precautions required by all MOS devices must be observed. Regardless of the fact that all inputs and outputs are protected against ESD damage by internal protection structures, the device can be degraded or destroyed by exposure to high electrostatic fields.

Absolute Maximum Ratings

The following section details the absolute maximum ratings of the 89C105 chip. Operation of the device at values in excess of those listed here will result in degradation or destruction of the device and should be avoided. This table does not imply that functional operation at conditions above those listed in the "Recommended Operating Conditions" is possible. This is a stress rating and operation of a device at or above this rating may cause failure or affect reliability.

Table 6-30 Absolute Maximum Operating Conditions

Name	Symbol	Min	Max	Units
Supply Voltage	V_{DD}	-0.5	+7	Volts
Input, Output Voltage	V_{IM}	-0.5	V_{DD} + 0.5	Volts
Current Drain VDD and VSS pins	I_I		100	mA
Lead Temperature (less than 10 second soldering)	T_l		250	°C
Operating Temperature	T_J	0	70	°C
Storage Temperature	T_A	-55	150	°C

Recommended Operating Conditions

The following section details the recommended DC operating conditions for the 89C105 chip:

Table 6-31 Recommended Operating Conditions

Name	Symbol	Min	Nom	Max	Units
Supply Voltage	V_{dd}	4.75	5.0	5.25	Volts
Operating Free-Air Temperature	T_a	0	25	70	°C

DC Characteristics

This table specifies the DC characteristics of the 89C100 chip over the range of the recommended operating conditions.

Table 6-32 DC Characteristics

Name	Symbol	Min	Nom	Max	Units
TTL Input Receiver					
High Level Input Voltage	V_{ih}	2.0			Volts
Logic Low Input Voltage	V_{il}			0.8	Volts
DS1216 Schmitt Input Receiver					
High Level Input Voltage	V_{ih}	1.9	1.6		Volts
Logic Low Input Voltage	Vil		1.2	0.9	Volts
DS1218 Schmitt Input Receiver					
High Level Input Voltage	V_{ih}	2.1	1.8		Volts
Logic Low Input Voltage	V_{il}		1.2	0.9	Volts
DS1238 Schmitt Input Receiver					
High Level Input Voltage	V_{ih}	4.1	3.8		Volts
Logic Low Input Voltage	V_{il}		1.2	0.9	Volts
Minimum high-level source current, V_{oh} = 2.4 V					
2 mA buffer	I_{oh}	2.0			mA
4mA buffer		4.0			
8mA buffer		8.0			
16 mA buffer		16.0			
24 mA buffer		24.0			
48mA buffer		n/a			
Minimum low-level sink current, V_{ol} = 0.4 V					
2 mA buffer	I_{ol}	2.0			mA
4mA buffer		4.0			
8mA buffer		8.0			
16 mA buffer		16.0			
24 mA buffer		24.0			
48mA buffer		48.0			
SCSIPAD (V_{ol} = 0.5 V)		48.0			
SCSIPADF (V_{ol} = 0.5 V)		48.0			
Input Leakage Current	I_{in}			±10	µA

Table 6-32 DC Characteristics

Name	Symbol	Min	Nom	Max	Units
Tristate Output Leakage Current	I_{oz}			±10	µA
High Level Output Voltage	V_{oh}	4.4	4.5		Volts
Low Level Output Voltage	V_{ol}		0.0	0.1	Volts
Input Capacitance	Ci		6		pF
Output Capacitance	Co		6		pF
Bidirectional Pin Capacitance	Cb		6		pF

AC Characteristics

The following table lists the 89C105 AC characteristics.

Table 6-33 The 89C105 AC Characteristics

Number	Description	Conditions	Min	Max	Units
SBus Timing					
F_{sb}	SBus Clock Frequency		16.7	25	
1	Clock Period		40	60	ns
2	Clock High		17		ns
3	Clock Low		17		ns
4	Hold wrt CLK Rising			0	ns
5	Setup to CLK Rising		15		ns
6	Setup to CLK Rising		15		ns
7	Hold wrt CLK Rising	See Note 1		1	ns
8	CLK Rising to Output Valid	100 pF load		22.5	ns
9	CLK Rising to Output Invalid	100 pF load		22.5	ns
10	CLK Rising to Output Valid	100 pF load		22.5	ns
11	CLK Rising to Output Invalid	100 pF load	2.5	25	ns
12	CLK Rising to Output Low	100 pF load		22.5	ns
13	CLK Rising to Output High	100 pF load		22.5	ns
EBus Timing					
14	Setup to CLK Rising		15		ns
15	Hold wrt CLK Rising			2	ns
16	CLK Rising to Output Valid	100 pF load		22.5	ns
17	CLK Rising to Output Invalid	100 pF load		22.5	ns
18	CLK Rising to Output Valid	100 pF load		22.5	ns

Table 6-33 The 89C105 AC Characteristics

Number	Description	Conditions	Min	Max	Units
Miscellaneous Timing					
19	CLK Rising to Output Valid	100 pF load		22.5	ns
20	10 MHz Clock Period	See Note 2	50		ns
Floppy Controller Timing					
21	fpy_clk32 Clock Period	See Note 3	31.3	31.3	ns
22	fpy_clk24 Clock High Time		16.7	25	ns
23	fpy_clk24 Clock Low Time		16.7	25	ns
24	fpy_clk24 Clock Period	See Note 3	41.7	41.7	ns
25	Internal Clock Period (t_{ci})	See Note 4	62.5	250	ns
26	fpy_dir Change to fpy_step Setup Time		1.0		µs
27	fpy_step Pulse Width		7	8	µs
28	fpy_step Rate	See Note 5	1	15	ms
29	fpy_index Pulse Width		4		tci
30	fpy_rddata Pulse Width		50		ns
31	fpy_rddata Data Rate	See Note 6	250K	1M	bit/s
32	fpy_wrgate to fpy_wrdata Setup Time		250		ns
33	fpy_wrdata Pulse Width	See Note 7	125	500	ns
Serial/Keyboard/Mouse Controllers Timing					
34	ser_clk LOW Width	See Note 8	50		ns
35	ser_clk HIGH Width		50		ns
36	ser_clk Cycle Time (Tpc)		122		ns
37	ser_rtxc_ Width		150		ns
38	ser_rtxc_ Cycle Time		4		Tpc
39	ser_trxc_ Width		150		ns
40	ser_trxc_ Cycle Time		4		Tpc
41	ser_dcd_ or ser_cts_ Width		200		ns
42	ser_sync_ Width		200		ns
43	ser_rxd to ser_rtxc_ Setup Time		0		ns
44	ser_rxd to ser_rtxc_ Hold Time		150		ns
45	ser_sync_ to ser_rtxc_ Setup Time		–200		ns
46	ser_sync_ to ser_rtxc_ Hold Time		5		Tpc
47	ser_trxc_ to ser_txd Delay			200	ns
48	ser_txd to ser_trxc_ Delay			200	ns

NOTE 1: This is the only violation of SBus Specification B.0. No known implementation to date provides less than 1.0 ns hold time on these signals.

NOTE 2: This clock MUST run at 10 MHz (100ns period) for correct software operation.

NOTE 3: The NCR82077 core uses a digital data separator to read the data off of the disk. Any variation from the nominal input clock frequencies will shift the capture range of the cell.

NOTE 4: Internal clock period is a function of the selected data rate.

Data Rate	Frequency	Period
1 Mbs	16 MHz	62.5 ns
500 Kbs	8 MHz	125 ns
300 Kbs	4.8 MHz	208 ns
250 Kbs	4 MHz	250 ns

NOTE 5: fpy_step Rate time is selected by a SPECIFY command. Failure to issue a specify command before issuing a Recalibrate or Seek command or implied seek will cause unpredictable results.

NOTE 6: fpy_rddata data rate is determined by the floppy tape drive or tape drive.

The values are:

- 1 Mbs – 1.0 µs minimum
- 500 Kbs – 2.0 µs minimum
- 300 Kbs – 3.3 µs minimum
- 250 Kbs – 4.0 µs minimum

NOTE 7: fpy_wrdata pulse width is based on the selected data rate. The values are 2 X t_{CI}:

- 1 Mbs - 2 X 62.5 = 125 ns
- 500 Kbs - 2 X 125 = 250 ns
- 300 Kbs - 2 X 208 = 416 ns
- 250 Kbs - 2 X 250 = 500 ns

NOTE 8: The input ser_clk is used to clock the serial controllers and the keyboard/mouse controller. The timing numbers in this section apply to both of these controllers.

AC Timing Diagrams

SBus/EBus Input Signals

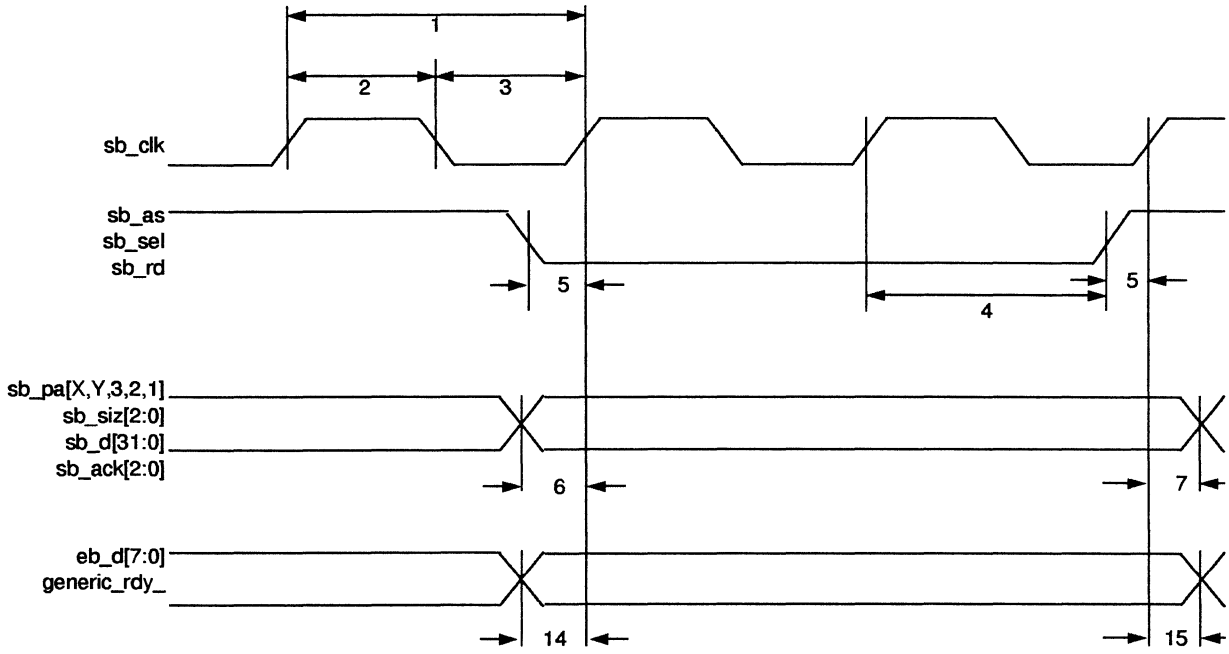


Figure 6-42 SBus/EBus Input Signals

Counter/Timer Input Clock

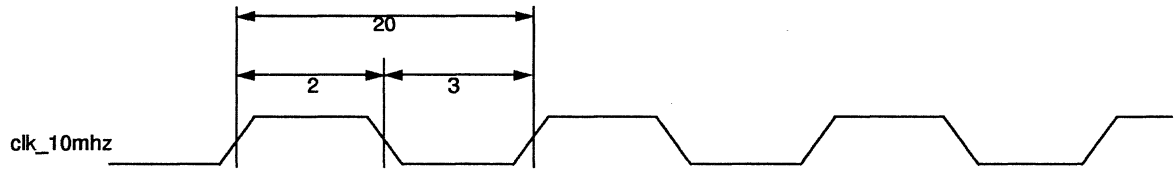


Figure 6-43 Counter/Timer Input Clock

SBus/EBus Output Signals

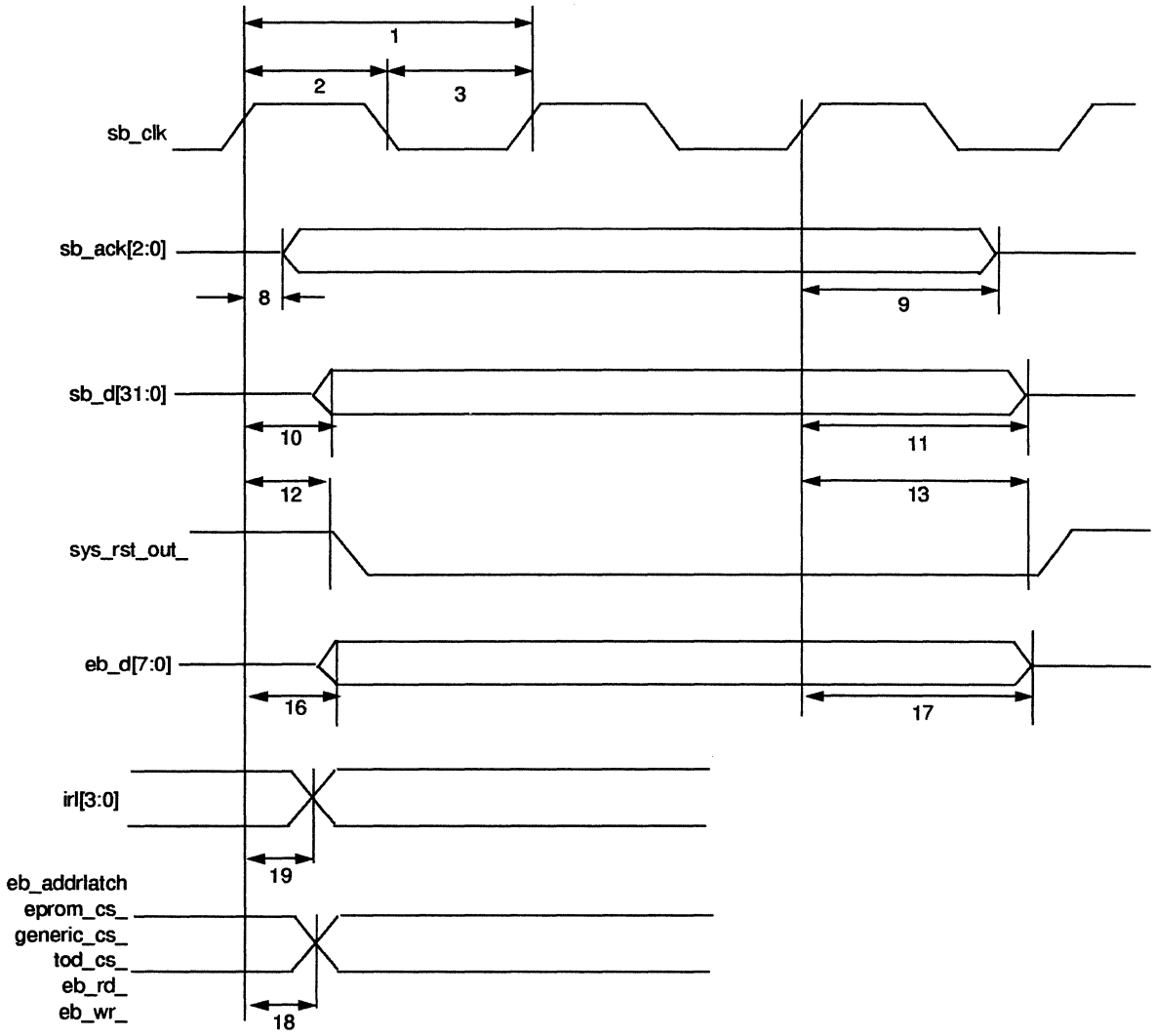


Figure 6-44 SBus/EBus Output Signals

Floppy Controller Clock Inputs

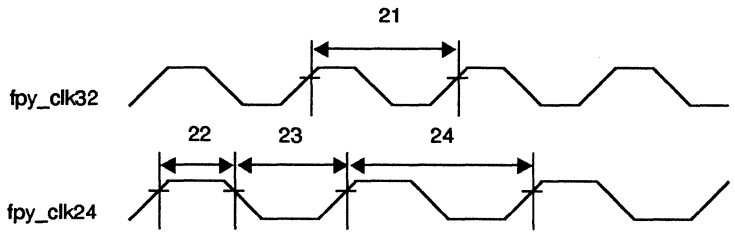


Figure 6-45 Floppy Controller Clock Inputs

Floppy Drive Interface Timing

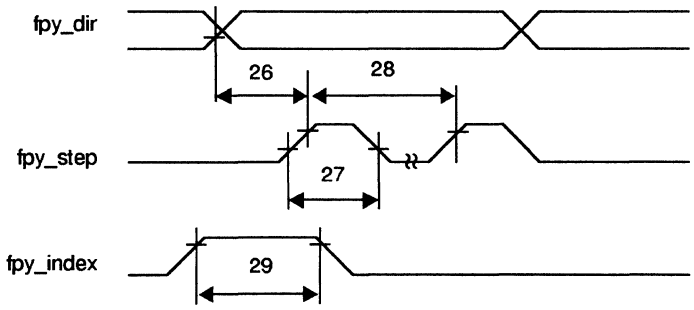


Figure 6-46 Floppy Drive Interface Timing

Floppy Read Data

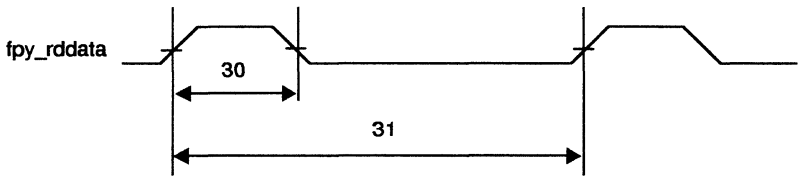


Figure 6-47 Floppy Read Data

Floppy Write Data

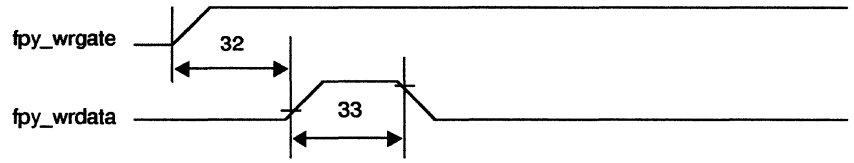


Figure 6-48 Floppy Write Data

Serial/Keyboard/Mouse Clock Input

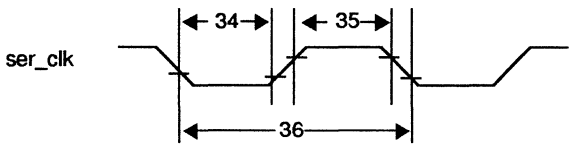


Figure 6-49 Serial/Keyboard/Mouse Clock Input

Serial Pulse Widths

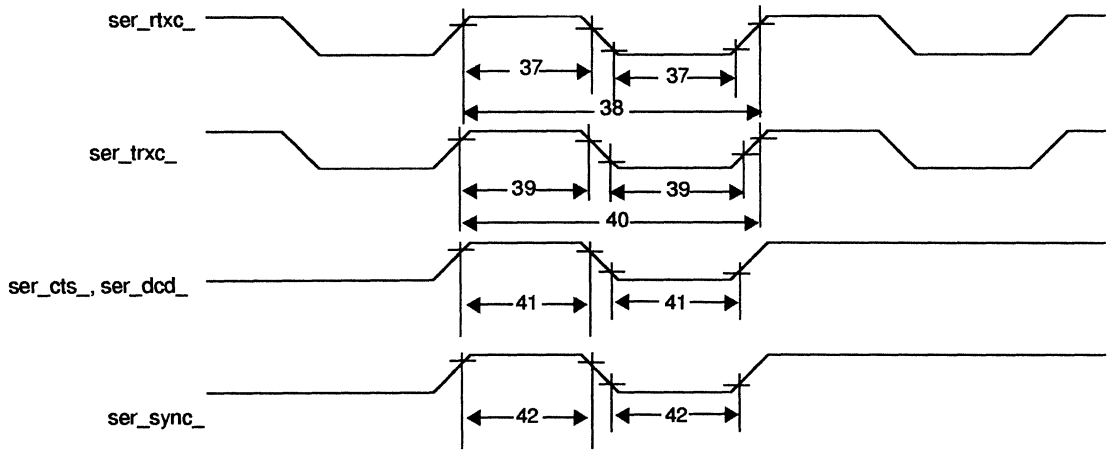


Figure 6-50 Serial Pulse Widths

Serial Data Timing

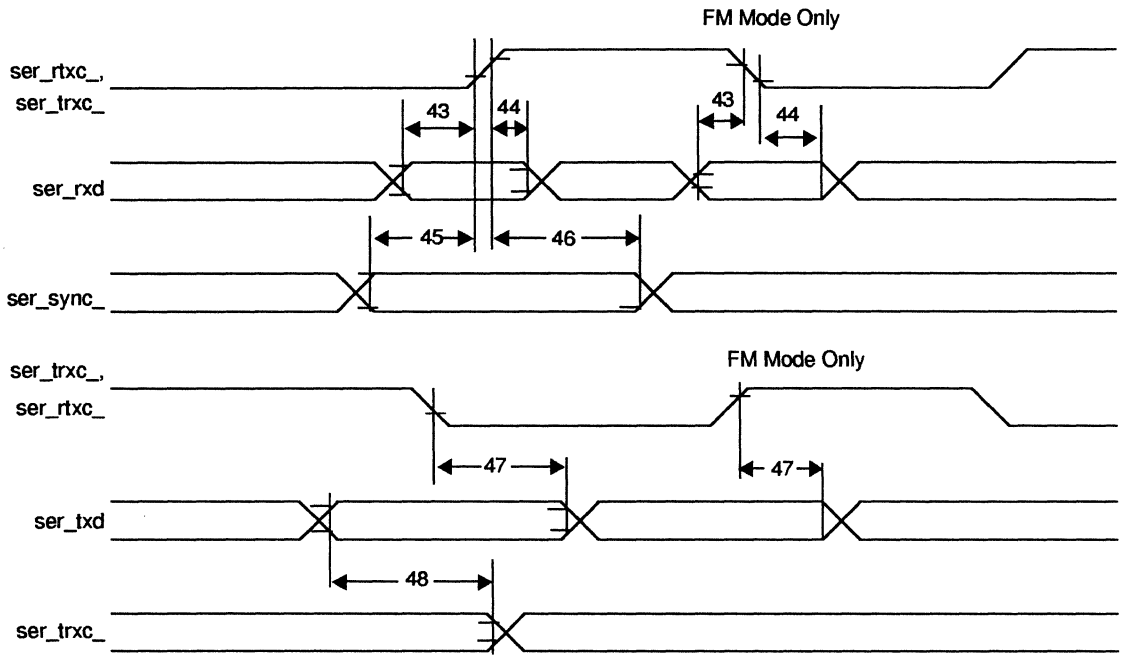


Figure 6-51 Serial Data Timing

Power Consumption

The 89C105 power consumption depends on SBus clock frequency, SBus and other output loading, and the workload. Power measurements are given for maximum loading (every ASF/functional block running in a manner to maximize power consumption) of the device. These measurements were taken over the entire operating ranges of voltage and temperature. The typical number reflects the average power consumed by the device under these conditions and the maximum number reflects the high limit of power that the part may consume in operation. Note that in normal system operation, the power consumption should fall at or below the typical number given here.

Table 6-34 Power Consumption

SBus Freq	Typical Power	Maximum Power	Units
25 MHz	280	560	mW

The 89C105 is packaged in a 160-pin PQFP package, and uses a custom copper leadframe to enhance thermal performance. The package thermal parameters are:

Table 6-35 Package Thermal Parameters (Still Air)

Θja Maximum	Units
24	°C/W

The 89C105 AC characteristics are given at 70°C junction temperature. By using the package characteristics and the power consumption numbers, one can get a rough idea of the allowable operating environments. Operation at junction temperatures in excess of 70°C is not recommended for performance reasons (the critical timing paths will not meet 25 MHz SBus specifications above this temperature). Operation at junction temperatures above 125°C is not recommended at any time, as it will cause reliability problems.

Packaging Information

The 89C105 chip is a standard cell design, based on the NCR VS700H technology (0.95 micron drawn, 0.7 effective). The chip contains 40,042 gates (18,870 cells), with the following breakdown:

Function	Gates
Serial Ports A/B	11,296
Keyboard/Mouse	4,545
Floppy ASF	11,916
SBus/Interrupt/Ctr-timers	6,991
Test Logic/Pad pre-drivers	5,294
TOTAL:	40,042

Packaging Identification

The 89C105 is packaged in a 160-pin Plastic Quad Flat Pack (PQFP), with the following marking (the last line will be filled in with wafer lot and date code information):

Mechanical Packaging Specification

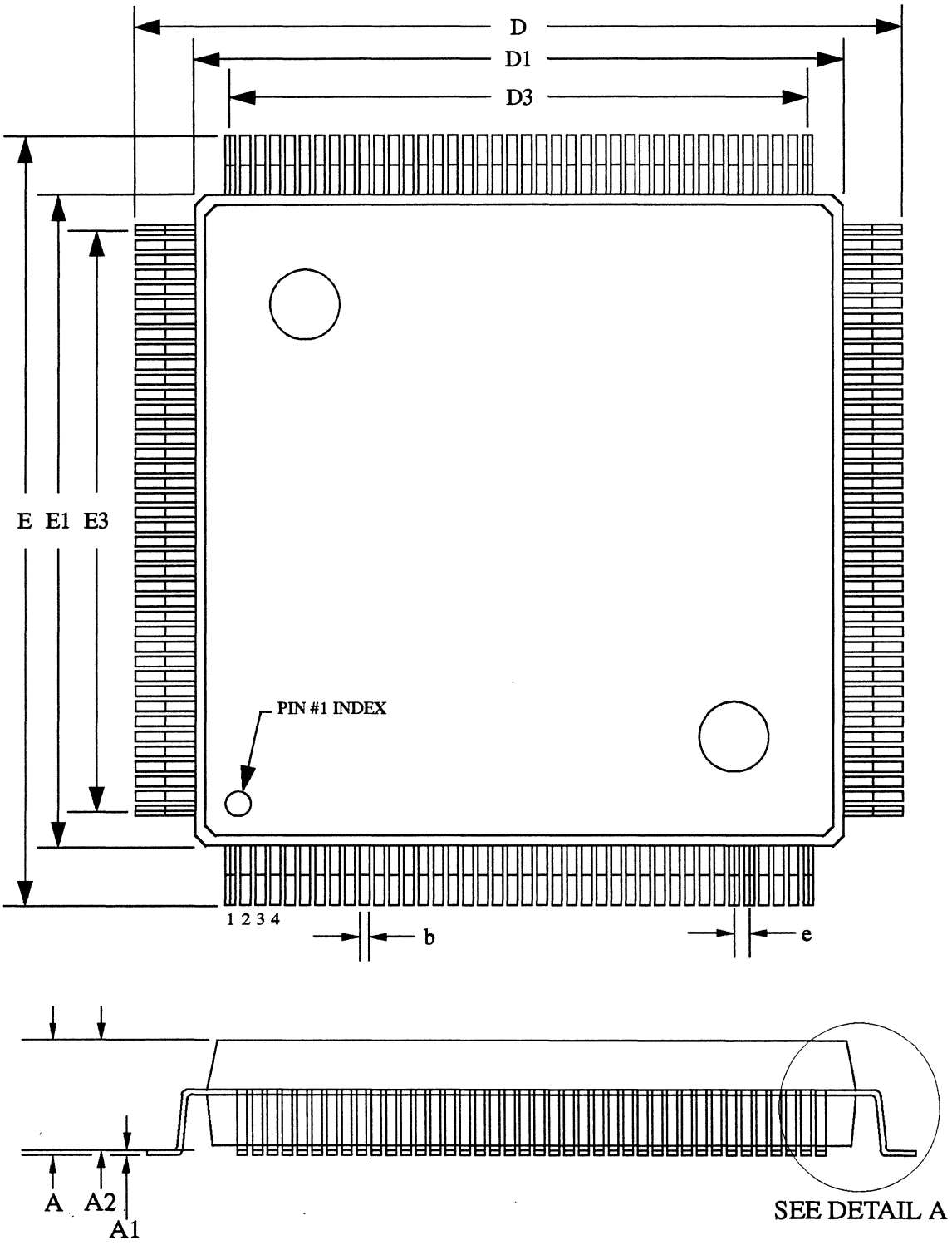


Figure 6-52 Mechanical Packaging Specification

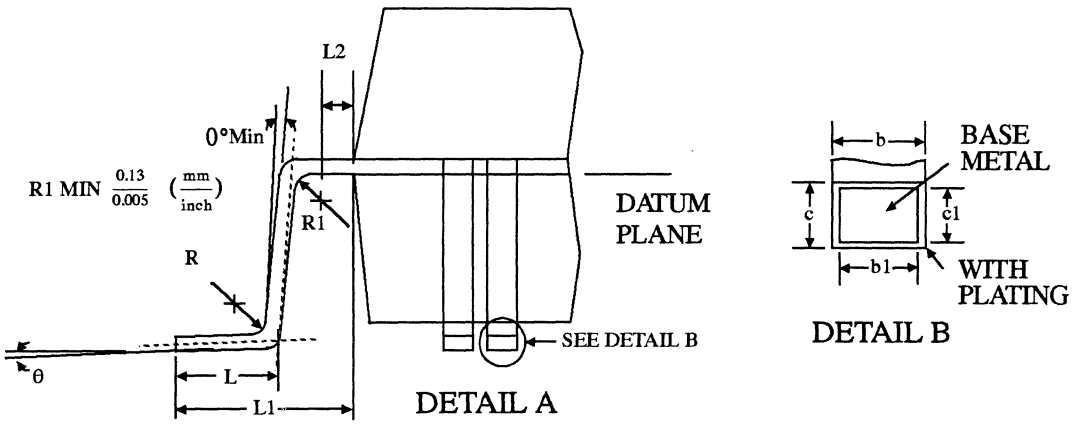


Figure 6-53 Mechanical Packaging Specifications (Detail A and B)

Table 6-36 Package Measurements (mm)

Sym	Min.	Nom.	Max.
A	—	—	4.07
A1	0.25	—	—
A2	3.17	3.42	3.67
D	31.65	31.90	32.15
D1	27.90	28.00	28.10
D3		25.35 Ref.	
E	31.65	31.90	32.15
E1	27.90	28.00	28.10
E3		25.35 Ref.	
L	0.65	0.80	0.95
L1		1.95 Ref.	
L2	0.40	—	—
e		0.65 BSC.	
b	0.22	—	0.38
b1	0.22	0.30	0.33
c	0.13	—	0.23
c1	0.13	—	0.17
R	0.13	—	0.40
θ	0°	—	7°

NCR89C105 System Considerations

Muxed Pins

The 89C105 is designed to interface directly to the Texas Instruments microSPARC processor, as well as serve as a substitute for the SEC chip for uniprocessor SuperSparc systems. In order to support both modes within the pin limitations of a 160-pin package, several pins were reused in different ways for the two processors. In addition, the modem support and density select function allow several pins to be used in multiple modes. A complete list of pins that can be used in multiple modes is shown in Table 6-37.

Table 6-37 Muxed pins

Pin #	Pin Name	Direction	Control	Synopsis
21	fpy_densel	output	Config[2]	Either the DENSEL or ME[2] output of the 82077 macrocell.
105	ser_rtxc_b_	input	Config[0]	Either ser_rtxc_b_ or emc_irq_ (Memory Controller interrupt).
124	iu_error_	input	Config[0]	Either iu_error_ or video_irq_.
160	msi_irq_	input	Config[1]	Either modem_ri or msi_irq_.

Interrupt Latency

The 89C105 uses interrupt-driven programmed I/O to support the internal macrocells. This imposes some fairly stringent interrupt latency requirements on the system processor and operating system, which in turn limits the performance of the devices. For example, the maximum data rate supported by the serial ports under SunOS in asynchronous mode is 38.4Kb/s, even though the hardware is capable of much higher data rates. For the floppy controller, the data rate supported by SunOS is limited to 500Kb/s (720KB/1.44MB floppies), even though the hardware can support 1Mb/s transfer rates (2.88MB floppies). Interrupt latency requirements for these data rates are included below. Note that the latency distribution is important; a system can have a low average latency but have enough periods of extremely high latency to disrupt PIO device operation. The devices can withstand some amount of latencies above the numbers given below, but they will see data overruns and underruns, which will reduce performance. At some point, the performance is degraded to an extent that makes operation impossible--for instance, when the number of overruns seen by the floppy controller averages more than 1/sector.

Table 6-38 Interrupt Latency Requirements

Macrocell	Latency	Background
82077 floppy controller	142 µs	This macrocell has a 16B FIFO, with the effective threshold set to 9B by the SunOS driver. This means that the latency tolerance at 500 Kb/s before overrunning or underrunning is 9Bx16 µs/B-2 µs (overhead) = 142 µs.
85c30 serial controller	208 µs	This macrocell has a 1B transmit FIFO. At 38.4Kb/s, this gives a latency tolerance of 8b/38400 = 208 µs.
85c30 keyboard/ mouse	208 µs	Same as above. The keyboard/mouse controller is typically run at only 2400 b/s, however, which yields a latency tolerance of 3.33 ms.

Many factors impact interrupt latency, some of the more significant of which are listed below.

- Integer performance. For a given OS/interrupt handler, the faster the processor, the shorter the latency.
- System load. System load has a large impact on latency, particularly loads that generate many interrupts, such as I/O processing. The interrupt levels for the PIO devices have been chosen to be at or near the highest possible priority to minimize the impact that other less-critical interrupting devices will have on PIO performance. Most interrupt handlers have some amount of critical code, however, so even low priority interrupts can significantly increase PIO latency.
- Critical code. Several OS resources require mutual exclusion control, which is typically implemented (in a uniprocessor) by disabling interrupts for a block of critical code. The number and length of these critical code sections depends on the system load.
- Message passing. Sun-4M® multiprocessing systems pass messages via software interrupts. These interrupts can be at higher levels than the PIO interrupts, and they generally also entail sections of critical code. Most of this code has been compiled out of uniprocessor-only versions of SunOS.

Unused Functional Blocks

For systems that don't use one or more of 89C105's functions, this section describes the steps that must be taken to disable the unused devices.

Unused Floppy Controller

The floppy macrocell uses an asynchronous reset, so a system that does not need the floppy function need not even supply a floppy clock. All inputs should be pulled to a known value, and the floppy interrupt bit should be masked in the System Interrupt Mask Register.

Unused Serial Ports or Keyboard/Mouse

The serial ports use a synchronous reset, so they need a clock to reach a benign state. Any clock up to about 30-32 MHz may be tied to the serial_clk input (such as the SBus clock). The serial controller and/or keyboard/mouse controller interrupt bit should be masked in the System Interrupt Mask Register, and all inputs should be pulled to a known value.

Unused Interrupt Controller

The interrupt controller must be used to access any of the internal macrocells, since they are all interrupt driven. However, if 89C105 is not providing the system interrupt control function, then the unused interrupt inputs should be tied high, all interrupt sources other than any internal devices that are being used should be masked in the System Interrupt Mask Register, and the IRL[3:0] lines should be decoded to provide discrete open-drain interrupts (if desired).

Unused Counter/Timers

If 89C105 is not providing the system counter/timer function, then the two counter interrupts should be masked in the System Interrupt Mask Register. In this case, the clk_10mhz input need not be exactly 10 MHz, but a clock still needs to be applied for 89C105 to exit reset. The clock may be up to 20 MHz. NOTE: The length of the SYS_RST_OUT_ pulse (and 89C105's own internal reset, which is the same length) will change if a clock frequency other than 10 MHz is used.

Unused Reset Controller

If 89C105 does not generate the system reset, then either the system reset or the power valid signal may be tied to POR_RST_IN_. 89C105 will not respond to accesses until the SYS_RST_OUT_ pulse goes inactive, however. This is over 200 milliseconds with CLK_10MHZ at 10 MHz, and half that at 20 MHz.

NCR85C30 Serial Communications Controller
Contents

NCR85C30 Serial Communications Controller

Features

- Application Specific Function (ASF) compatible with the standard NMOS 8530 SCC
- Standard cell design allows easy integration
- Two full-duplex channels
- Highly programmable serial communications controller allows flexible operations
- Supports SDLC, HDLC, IBM BISYNC, ASYNC and other protocols
- Contains baud rate generators, digital phase-locked loops, CRC generators/checkers and other support circuitry
- Simulation and test patterns provided
- 44-pin PLCC kit part
- 8 MHz operation

Product Description

Designed with NCR's VS700H submicron CMOS cell-based technology, the 85C30 Serial Communications Controller operates as a dual-channel, multiprotocol data communications peripheral. It can function in either serial-to-parallel or parallel-to-serial mode. It can be configured to meet a wide variety of serial communication applications, including full-duplex and half-duplex Bus architectures, Token Passing Ring (SDLC Loop mode), or Star configurations.

The 85C30 has access to 14 Write and 7 Read registers per channel. Internal functions reduce the need for external logic, and allow easy integration with other high-level digital and analog functions.

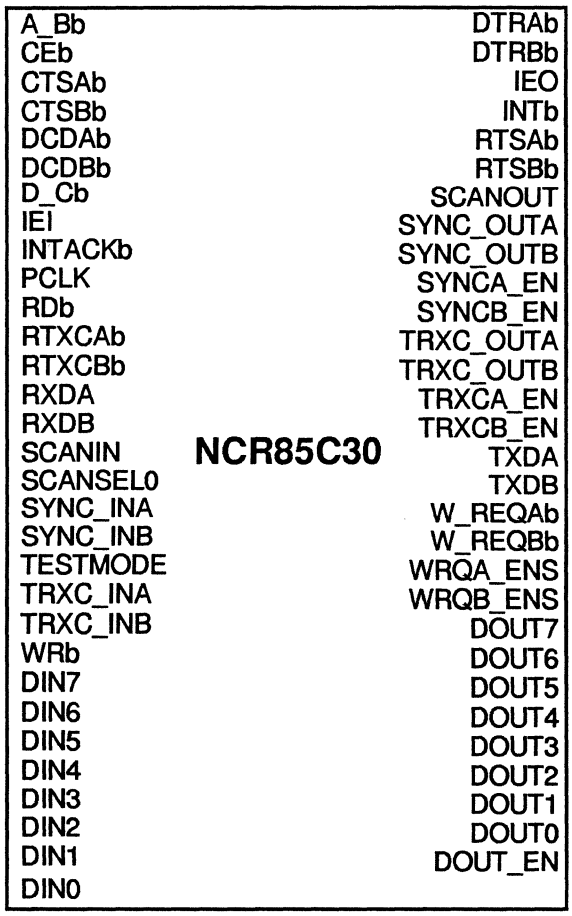


Figure 7-1 Symbol

Pin Name	Type	Description
A_Bb	I	Channel Select. A high level on this signal selects Channel A for read or write operations. A low level selects Channel B.
CEb	I	Cell Enable. A low level on CEb selects the cell for read or write operation.
CTSAb CTSBb	I I	Clear to Send. In Auto Enable mode, these signals will enable their respective transmitters when activated low. Otherwise, these signals can be used as general purpose inputs.
DCDAb DCDBb	I I	Data Carrier Detect. In Auto Enable mode, these signals will enable their respective receivers when activated low. Otherwise, these signals can be used as general purpose inputs.
D_Cb	I	Data/Control Select. A high on D_Cb indictes a data transfer operaton during a read or write. A low indicates a control operation.
IEI	I	Interrupt Enable In. IEI is used with the IEO signal to form an interrupt daisy chain. A high on IEI indicates that no higher priority device has an interrupt request or an interrupt under service.
INTACKb	I	Interrupt Acknowledge. A low on INTACKb indicates that the succeeding read will be part of an interrupt acknowledge cycle. If IEI is high, the interrupt vector will be placed on the data bus.
PCLK	I	Master Clock. Master clock signal which allows synchronous operation of the control logic.
RDb	I	Read. A low RDb signal performs a read if the cell is selected. Internal data will be driven onto the data bus. A low on RDb and WRb simultaneously performs a reset.
RTXCAb RTXCBb	I I	Receive/Transmit Clocks. These signals can be programmed to be the source clock for the receiver, transmitter, baud rate generator, or the digital phase lock loop in their respective channels.
RXDA RXDB	I I	Receive Data. These signals receive the serial input stream for their respective channels.
SCANIN	I	Scan Input. When TESTMODE is high, this signal supplies the input to serial partial scan chains to aid in fault testing.
SCANSEL0	I	Scan Select. When TESTMODE is high, this signal selects between two internal partial scan chains.

Pin Name	Type	Description
SYNC_INA SYNC_INB	I I	Sync Input. In async mode, these signals control the status of the SYNC/HUNT bit in register 0. In External Sync Mode, these signals are used for character synchronization.
TESTMODE	I	Test Mode Enable. A high on this signal allows the operation of the internal partial scan chains for test purposes. A high on TESTMODE with a low on RDb and WRb performs a test mode reset which initializes all flip-flops to a known state. This signal must be forced low for normal operation.
TRXC_INA TRXC_INB	I I	Transmit/Receive Clock Input. These signals can supply the clock input for the receiver or transmitter if programmed appropriately for their respective channels.
WRb	I	Write. A low on this signal indicates a write operation. A low on the WRb and RDb simultaneously performs a reset.
DIN0–DIN7	I	Data Inputs. These signals are the write data inputs for commands or data transfer.
DOUT0– DOUT7	O	Data Outputs. These signals are the data outputs which provide the results of a read operation. These signals can be used with the DIN and DOUT_EN signal to form a bidirectional bus.
DOUT_EN	O	Data Output Enable. A high on this signal indicates that a valid read operation is active. The DOUT_EN signal can be used to control tristate drivers to form a bidirectional data bus.
DTRAb DTRBb	O O	Data Terminal Ready/DMA Request. These signals follow the inverse of the DTR bit in Write Register 5 if programmed for the DTR function in Register 14. This signal can also act as a DMA request if so programmed in Register 14.
IEO	O	Interrupt Enable Output. This signal is used with IEI to form an interrupt daisy chain. This signal is high if IEI is high and the controller is servicing an interrupt or has an interrupt pending.
INTb	O	Interrupt. A low on INTb indicates an interrupt is pending.
RTSAb RTSBb	O O	Request to Send. When in Auto Enable and Async Mode, this signal will go high after the transmitter is empty. Otherwise this signal follows the inverse of the RTS bit in Register 5.
SCANOUT	O	Scan Output. This signal scans out the result of the internal partial scan chain.

Pin Name	Type	Description
SYNC_OUTA SYNC_OUTB	O O	Sync Output. These signals are active low when the proper sync character or flag has been received in the Synchronous or SDLC modes, respectively.
SYNCA_EN SYNCB_EN	O O	Sync Output Enable. These signals can be used with the SYNC_IN and SYNC_OUT pins to create a bidirectional SYNC pin. A high on this signal indicates an output.
TRXC_OUTA TRXC_OUTB	O O	Transmit/Receive Clock Output. These signals can be programmed to output the transmitter clock, the baud rate generator output, the digital phase lock loop output, or the oscillator output.
TRXCA_EN TRXCB_EN	O O	TRXC Output Enable. These signals can be used with the TRXC_IN and TRXC_OUT signals to form a bidirectional TRXC pin. A high on this signal indicates an output.
TXDA TXDB	O O	Transmit Data. These signals are the transmitter serial output data for their respecive channels.
W_REQAb W_REQBb	O O	Wait/Request. These signals can be used as a Wait signal to the CPU or as a DMA request per programming of Write Register 1.
WRQA_ENS WRQB_ENS	O O	Wait/Request Enable Strength. These signals can be used with the W_REQb signals to recreate the varied strength of the high level on the W_REQb pin. A high indicates a driving signal, a low indicates a floating signal.

Input Signal	Capacitance
A_B	.054 pF
$\overline{CE}$	.054 pF
$\overline{CTSA}$	.054 pF
$\overline{CTSB}$	.054 pF
$\overline{DCDA}$	.054 pF
$\overline{DCDB}$	.054 pF
$\overline{D\_C}$	.054 pF
IEI	.054 pF
$\overline{INTACK}$	.054 pF
PCLK	.054 pF
$\overline{RD}$	.054 pF
$\overline{RTXCA}$	.054 pF
$\overline{RTXCB}$	.054 pF
RXDA	.054 pF
RXDB	.054 pF
SCANIN	.054 pF
SCANSELO	.054 pF
SYNC_INA	.054 pF
SYNC_INB	.054 pF
TESTMODE	.054 pF
TRXC_INA	.054 pF
TRXC_INB	.054 pF
$\overline{WR}$	.054 pF
DIN0–DIN7	.054 pF

Output Signal	Drive Capability
DOUT0–DOUT7	HBUF
DOUT_EN	HBUF
$\overline{DTRA}$	HBUF
$\overline{DTRB}$	HBUF
IEO	HBUF
$\overline{INT}$	HBUF
$\overline{RTSA}$	HBUF
$\overline{RTSB}$	HBUF
SCANOUT	HBUF
SYNC_OUTA	HBUF
SYNC_OUTB	HBUF
SYNCA_EN	HBUF
SYNCB_EN	HBUF
TRXC_OUTA	HBUF
TRXC_OUTB	HBUF
TRXCA_EN	HBUF
TRXCB_EN	HBUF
TXDA	HBUF
TXDB	HBUF
$\overline{W\_REQA}$	HBUF
$\overline{W\_REQB}$	HBUF
WRQA_ENS	HBUF
WRQB_ENS	HBUF

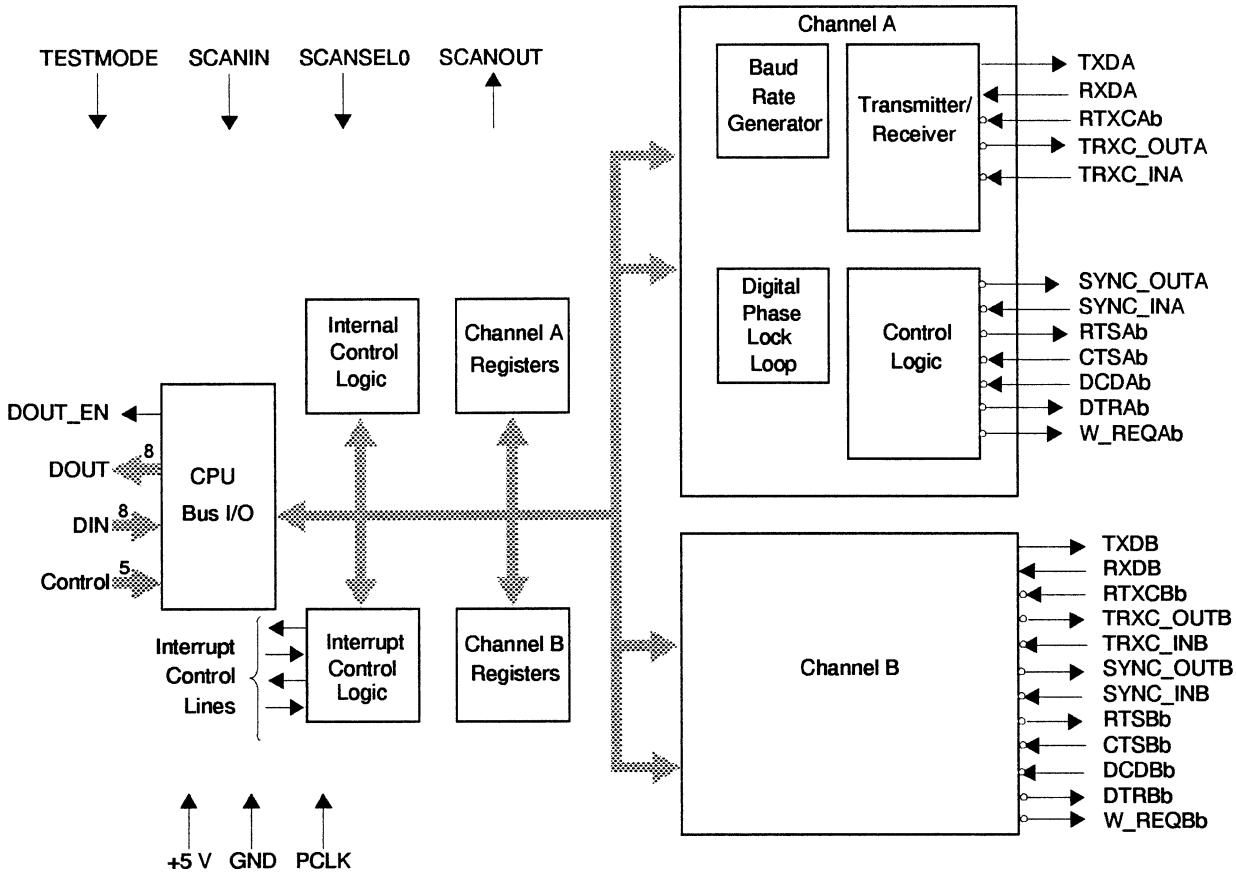


Figure 7-2 NCR85C30 Serial Communications Controller Block Diagram

Architectural Description

This section briefly describes major functional blocks of the 85C30 Serial Communications Controller (SCC) design, referencing the functional block diagram. For further detail, consult an 8530 technical manual.

CPU Bus I/O

This functional block comprises the bus interface logic. Major features include the following: control of read/write commands, interrupt acknowledge, bus functions, and DMA request and wait logic.

The signals A_Bb, D_Cb, and INTACKb are used to control the type of bus transaction. A_Bb is used for channel selection; a high level on this signal selects Channel A for read or write operations, whereas a low level selects Channel B. D_Cb is used for data/control selections; a high level on D_Cb causes a data transfer operation during a read or write, but a low level causes a control operation. INTACKb is used for interrupt acknowledgment. If IEI is high, a low on INTACKb causes the succeeding read to issue from RR2 (interrupt vector) as part of an interrupt acknowledge cycle.

Addressing in the SCC is accomplished by a two-step sequence in which the SCC registers are accessed utilizing a register pointer to perform the actual addressing whenever D_Cb is low. If D_Cb is high, reads or writes bypass the pointer bits and allow direct access of data registers. Writing to WR0 sets the pointer for a particular register and then that register is accessed on the next read or write cycle having D_Cb low.

Interrupt Control Logic

This functional block has two major features: a "daisy chaining" mechanism governed by IEI and IEO pins, and prioritized interrupt vectors (Receive, Transmit, and External/Status in that order). Channel A interrupts have higher priority than Channel B interrupts. Designers can tie multiple devices together, chaining the IEI of one device to the IEO of another. Multiple device interrupts can be acknowledged as a group with only the highest priority device being serviced. Interrupt vectors can be placed on the data bus in order to enhance interrupt response time. Channel A interrupt vectors can include status, whereas Channel B vectors always include status.

Registers

The 85C30 registers consist of Channel A and Channel B registers, as well as two common ones - registers 2 and 9.

See the section entitled *Register Description* for further information.

Baud Rate Generator (BRG)

This function block has a programmable clock output, usable by the transmitter/receiver, that can also drive the DPLL. Its multiple clock sources include PCLK, TRXC_INA, TRXC_INB, RTXCAb, or RTXCBb. The BRG's time constant is programmed utilizing WR12 and WR13, applying the following equation:

$$\text{Time Constant} = \frac{\text{Clock Frequency}}{2 \times (\text{Clock Mode}) \times (\text{Baud Rate})} - 2$$

Digital Phase Lock Loop (DPLL)

This functional block accomplishes the following two features: it allows clock recovery from a serial data stream and it can be used with FM-encoded or NRZI-encoded data.

The required driving clock rates are 32 times the data rate for NRZI encoding mode, versus 16 times the data rate for FM encoding mode.

The DPLL has the following components: an edge detector, 5-bit counter, and two output decoders (one for the receive clock and one for the transmit clock). Either the BRG output or the RTXCAb/RTXCBb pin can be used as the DPLL clock source input. The Enter Search Mode command is used to enable the DPLL by unlocking the counter and enabling the edge detector. Using edge detection and making counting cycle adjustments correlated to bit boundary transitions, the DPLL maintains proper phase relationships (counteracting jitter) during clock recovery. There is a 90-degree lag of the transmit clock after the receive clock in FM mode. In NRZI mode the transmit and receive clocks are in phase.

Transmitter

The transmitter converts parallel data to serial data for transmission, handling common asynchronous and synchronous protocols. Character lengths from five to eight bits per character can be specified.

The major features of this functional block include the following: an 8-bit shift register, parity generator, CRC generator, zero-bit insertion logic for SDLC/HDLC protocols, format encoder, and clock generator.

The shift register accepts data from various registers under transmitter control to perform parallel-to-serial conversion and proper protocol sequencing. Input to the shift register includes Register 8 for data, Registers 7 and 6 for SYNC characters, and Register 5 for character length control.

Parity generation on data can be enabled by writing WR4. Even or odd parity can be selected with parity generated only on the enabled data bits. Parity is transmitted after the last bit of a data character.

The CRC generator can accommodate both the CRC-16 ($X^{16} + X^{15} + X^2 + 1$) and CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) polynomials and can be used only with the synchronous modes. In SDLC mode, CRC is not calculated on the inserted zero bits, and automatic inversion of the CRC is performed as required by the protocol.

In SDLC mode, zero-bit insertion is performed as required by the protocol. When more than five consecutive ones occur in the data stream (data and CRC), a zero is automatically inserted after the fifth consecutive one. This inserted zero will be automatically deleted by the receiver for transparent operation.

The format encoder allows FM0, FM1, NRZI, or NRZ data encoding.
The control logic is capable of handling the following five modes: ASYNC, MONOSYNC, BISYNC, SDLC/HDLC, and SDLC Loop (a superclass of SDLC where communicating devices can be in a ring configuration). The ASYNC mode is capable of handling 1, 1.5, or 2 stop bits per character; MONOSYNC is capable of handling 6-bit or 8-bit sync characters; BISYNC is capable of handling 12-bit or 16-bit sync characters; and SDLC/HDLC can send aborts or flags on underrun.

The clock generator feature allows for ASYNC mode divide by 16, 32, or 64 clock, as well as 1X mode.

Receiver

The receiver functional block converts received serial data to parallel data. It handles the following common protocols: ASYNC, MONOSYNC, BISYNC, SDLC/HDLC, and SDLC Loop. Character lengths from five to eight bits per character can be specified.

The major features of this functional block include the following: data and sync character shift registers, a parity checker, a CRC checker, an 8-bit CRC delay register, a 3-byte-deep receive data FIFO, zero-bit deletion logic for SDLC, and clock generator.

The shift registers include an 8-bit data shift register and sync shift register to allow recognition of sync characters in MONOSYNC, BISYNC, and SDLC/HDLC modes.

The SCC receiver can be programmed to cause a Special Condition interrupt upon parity error detection, FIFO overrun, CRC/framing error, or EOF and thereby lock the receive data FIFO until reset by the CPU. Interrupts can be optionally set for the first character or all characters received. Parity can be excluded as a special condition.

The CRC checker validates data integrity in synchronous modes. It can use two different CRC polynomials for error checking, both CRC-16 and CRC-CCITT. For MONOSYNC and BISYNC modes, an 8-bit delay register allows sufficient decision time for character exclusion from the CRC calculation.

The receive FIFO, zero-bit deletion logic, format decoder, and clock generator have provisions of note to the designer. The 3-byte-deep receive FIFO includes data with adjoining error bits for parity, framing error, EOF, overrun, and CRC. The SCC has zero-bit deletion logic for SDLC mode. A format decoder allows for decoding of FM0, FM1, NRZI or NRZ encoded data. The clock generator feature allows for ASYNC divide-by-16, -32, or -64 clock, as well as 1X mode.

Register Description

The following sections describe the NCR 85C30 SCC registers. Each register is detailed in terms of bit configuration, the active states (see below) of each bit, and their definitions.

Read Register	Description
RR0	Transmit/Receive Buffer Status and External Status
RR1	Receive Condition Status/Residue Codes.
RR2	Interrupt Vector (Modified in B Channel)
RR3	Interrupt Pending (Channel A only)
RR8*	Receive Buffer
RR10	Loop/Clock Status
RR12	Lower Byte of Time Constant
RR13	Upper Byte of Time Constant
RR15	External Status Interrupt Enable
*Can also be selected by driving D_Cb high during access.	

Write Registers	Description
WR0	Command Register
WR1	TX/RX Interrupt and Data Transfer Mode Definition
WR2	Interrupt Vector
WR3	Receive Parameters and Control
WR4	TX/RX Miscellaneous Parameters and Modes
WR5	Transmit Parameter and Controls
WR6	Sync Character or SDLC Address Field
WR7	Sync Character or SDLC Flag
WR8*	Transmit Buffer
WR9	Master Interrupt Control
WR10	Miscellaneous Transmitter/Receiver Control Bits
WR11	Clock Mode Control
WR12	Lower Byte of Baud Rate Generator Time Constant
WR13	Upper Byte of Baud Rate Generator Time Constant
WR14	Miscellaneous Control Bits
WR15	External Status/Interrupt Control
*Can also be selected by driving D_Cb high during access.	

Table 7-1 Write Register 0 – Bit Functions

Signal	WR0 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Register 0 Select						0	0	0
Register 1 Select						0	0	1
Register 2 Select						0	1	0
Register 3 Select						0	1	1
Register 4 Select						1	0	0
Register 5 Select						1	0	1
Register 6 Select						1	1	0
Register 7 Select						1	1	1
Register 8 Select			0	0	1	0	0	0
Register 9 Select			0	0	1	0	0	1
Register 10 Select			0	0	1	0	1	0
Register 11 Select			0	0	1	0	1	1
Register 12 Select			0	0	1	1	0	0
Register 13 Select			0	0	1	1	0	1
Register 14 Select			0	0	1	1	1	0
Register 15 Select			0	0	1	1	1	1
Null Code			0	0	0			
Reset External/Status Interrupts			0	1	0			
Send Abort (SDLC Mode)			0	1	1			
Enable Int on Next RX Character			1	0	0			
Reset TX Interrupt Pending			1	0	1			
Error Reset			1	1	0			
Reset Highest IUS			1	1	1			
Null Code	0	0						
Reset RX CRC Checker	0	1						
Reset TX CRC Generator	1	0						
Reset TX Underrun/EOM Latch	1	1						

Table 7-2 Write Register 1 - Bit Functions

Signal	WR1 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
External Interrupt Enable								●
Transmit Interrupt Enable							●	
Parity is Special Condition						●		
Receive Interrupt Disable				0	0			
Receive Interrupt on First Character or Special Condition				0	1			
Interrupt on all Receive Characters or Special Condition				1	0			
Receive Interrupt on Special Condition Only				1	1			
Wait/DMA Request on Receive/ Transmit			●					
Wait/DMA Request Function		●						
Wait/DMA Request Enable	●							

Table 7-3 Write Register 2 - Interrupt Vector

Signal	WR2 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Interrupt Vector	V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0

Table 7-4 Write Register 3 - Bit Functions

Signal	WR3 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Receiver Enable								●
Sync Character Load Inhibit							●	
Address Search Mode (SDLC)						●		
Receiver CRC Enable					●			

Table 7-4 Write Register 3 – Bit Functions

Signal	WR3 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Enter Hunt Mode				●				
Auto Enables			●					
RX 5 Bits/Character	0	0						
RX 7 Bits/Character	0	1						
RX 6 Bits/Character	1	0						
RX 8 Bits/Character	1	1						

Table 7-5 Write Register 4 – Bit Functions

Signal	WR4 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Parity Enable								●
Parity Even/$\overline{\text{Odd}}$							●	
Sync Modes Enable					0	0		
1-Stop Bit/Character					0	1		
1.5-Stop Bits/Character					1	0		
2-Stop Bits/Character					1	1		
8-Bit Sync Character			0	0				
16-Bit Sync Character			0	1				
SDLC Mode			1	0				
External Sync Mode			1	1				
X1 Clock Mode	0	0						
X16 Clock Mode	0	1						
X32 Clock Mode	1	0						
X64 Clock Mode	1	1						

Table 7-6 Write Register 5 - Bit Functions

Signal	WR5 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Transmitter CRC Enable								●
RTS							●	
$\overline{\text{SDLC}}$/CRC-16						●		
Transmitter Enable					●			
Send Break				●				
Transmitter 5 Bits (or less)/ Character		0	0					
Transmitter 7 Bits/Character		0	1					
Transmitter 6 Bits/Character		1	0					
Transmitter 8 Bits/Character		1	1					
DTR	●							

Table 7-7 Write Register 6 - Protocols

Protocol	WR6 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Monosync, 8 Bits	$SYNC_7$	$SYNC_6$	$SYNC_5$	$SYNC_4$	$SYNC_3$	$SYNC_2$	$SYNC_1$	$SYNC_0$
Monosync, 6 Bits	$SYNC_1$	$SYNC_0$	$SYNC_5$	$SYNC_4$	$SYNC_3$	$SYNC_2$	$SYNC_1$	$SYNC_0$
Bisync, 16 Bits	$SYNC_7$	$SYNC_6$	$SYNC_5$	$SYNC_4$	$SYNC_3$	$SYNC_2$	$SYNC_1$	$SYNC_0$
Bisync, 12 Bits	$SYNC_3$	$SYNC_2$	$SYNC_1$	$SYNC_0$	1	1	1	1
SDLC	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
SDLC (Address Range)	ADR7	ADR6	ADR5	ADR4	×	×	×	5

Table 7-8 Write Register 7 – Protocols

Protocol	WR7 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Monosync 8 Bits	$SYNC_7$	$SYNC_6$	$SYNC_5$	$SYNC_4$	$SYNC_3$	$SYNC_2$	$SYNC_1$	$SYNC_0$
Monosync 6 Bits	$SYNC_5$	$SYNC_4$	$SYNC_3$	$SYNC_2$	$SYNC_1$	$SYNC_0$	×	×
Bisync,16 Bits	$SYNC_{15}$	$SYNC_{14}$	$SYNC_{13}$	$SYNC_{12}$	$SYNC_{11}$	$SYNC_{10}$	$SYNC_9$	$SYNC_8$
Bisync, 12 Bits	$SYNC_{11}$	$SYNC_{10}$	$SYNC_9$	$SYNC_8$	$SYNC_7$	$SYNC_6$	$SYNC_5$	$SYNC_4$
SDLC	0	1	1	1	1	1	1	0

Table 7-9 Write Register 8 – 8-Bit Data Register

Signal	WR8 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Data	●	●	●	●	●	●	●	●

Table 7-10 Write Register 9 – Bit Functions

Signal	WR9 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Vector Includes Status								●
No Vector on Interrupt Acknowledge							●	
Disable Lower Chain						●		
Master Interrupt Enable					●			
Status High/$\overline{\text{Status Low}}$				●				
Not Used (Write to 0)			●					
No Reset	0	0						
Channel Reset B	0	1						
Channel Reset A	1	0						
Force Hardware Reset	1	1						

Table 7-11 Write Register 10 - Bit Functions

Signal	WR10 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
6-Bit/$\overline{8\text{-Bit}}$ Sync								●
Loop Mode							●	
Abort/$\overline{\text{Flag}}$ on Underrun						●		
Mark/$\overline{\text{Flag}}$ on Idle					●			
Go Active on Poll				●				
NRZ		0	0					
NRZI		0	1					
FM1 (Tansition = 1)		1	0					
FM0 (Transition = 0)		1	1					
CRC Preset 1/0	●							

Table 7-12 Write Register 11 - Bit Functions

Signal	WR11 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
TRXC_Out = XTAL Output							0	0
TRXC_Out = Transmit Clock							0	1
TRXC_Out = BR Generator Output							1	0
TRXC_Out = DPLL Output							1	1
TRXC_EN						●		
Transmit Clock = RTXCb Pin				0	0			
Transmit Clock = TRXC_IN Pin				0	1			
Transmit Clock = BR Generator Output				1	0			
Transmit Clock = DPLL Output				1	1			
Receive Clock = RTXCb Pin		0	0					

Table 7-12 Write Register 11 – Bit Functions

Signal	WR11 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Receive Clock = TRXC_IN Pin		0	1					
Receive Clock = BR Generator Output		1	0					
Receive Clock = DPLL Output		1	1					
RTXCb XTAL/$\overline{\text{NO XTAL}}$	●							

Table 7-13 Write Register 12 – Time Constant Bits (lower Byte)

Signal	WR12 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Time Constant (Lower Byte)	TC_7	TC_6	TC_5	TC_4	TC_3	TC_2	TC_1	TC_0

Table 7-14 Write Register 13 – Time Constant Bits (Upper Byte)

Signal	WR13 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Time Constant (Upper Byte)	TC_{15}	TC_{14}	TC_{13}	TC_{12}	TC_{11}	TC_{10}	TC_9	TC_8

Table 7-15 Write Register 14 – Bit Functions

Signal	WR14 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
BR Generator Enable								●
BR Generator Source							●	
DTRb/Request Function						●		
Auto Echo					●			

Table 7-15 Write Register 14 - Bit Functions

Signal	WR14 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Local Loopback				●				
Null Command	0	0	0					
Enter Search Mode	0	0	1					
Reset Missing Clock	0	1	0					
Disable DPLL	0	1	1					
Clock Source = BR Generator	1	0	0					
Clock Source = RTXCb	1	0	1					
Clock FM Mode	1	1	0					
Clock NRZI Mode	1	1	1					

Table 7-16 Write Register 15 - Bit Functions

Signal	WR15 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Not Used (Write to 0)								●
Zero Count Interrupt Enable							●	
Not Used (Write to 0)						●		
DCD Interrupt Enable					●			
Sync/Hunt Interrupt Enable				●				
CTS Interrupt Enable			●					
TX Underrun/EOM Interrupt Enable		●						
Break/Abort Interrupt Enable	●							

Read Registers

Table 7-17 Read Register 0 – Bit Functions

Signal	RR0 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
RX Character Available								●
Zero Count Interrupt Enable							●	
Transmit Buffer Empty						●		
DCD					●			
Sync/Hunt				●				
CTS			●					
Transmit Underrun/EOM		●						
Break/Abort	●							

Table 7-18 Read Register 1 – Bit Functions

Signal	RR1 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
All Sent								●
Residue Code 2							●	
Residue Code 1						●		
Residue Code 0					●			
Parity Error				●				
RX Overrun Error			●					
CRC/Framing Error		●						
End of Frame (SDLC)	●							

Table 7-19 Read Register 2 - Interrupt Vector

Signal	RR2 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Interrupt Vector*	V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0
*Modified in B Channel								

Table 7-20 Read Register 3 - Bit Functions

Signal	RR3 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Channel B EXT/STAT Interrupt Pending*								●
Channel B Transmit Interrupt Pending*							●	
Channel B RX Interrupt Pending*						●		
Channel A EXT/STAT Interrupt Pending*					●			
Channel A Transmit Interrupt Pending*				●				
Channel A RX Interrupt Pending*			●					
0	●	●						
*Always 0 in B Channel								

Table 7-21 Read Register 8 - 8-Bit Data Register

Signal	RR8 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Data	●	●	●	●	●	●	●	●

Table 7-22 Read Register 10 - Bit Functions

Signal	RR10 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
On Loop							●	
Loop Sending				●				
Two Clocks Missing		●						
One Clock Missing	●							
0			●		●	●		●

Table 7-23 Read Register 12 - Time Constant Bits (Lower Byte)

Signal	RR12 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Time Constant (Lower Byte)	TC_7	TC_6	TC_5	TC_4	TC_3	TC_2	TC_1	TC_0

Table 7-24 Read Register 13 - Time Constant Bits (Upper Byte)

Signal	WR13 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Time Constant (Upper Byte)	TC_{15}	TC_{14}	TC_{13}	TC_{12}	TC_{11}	TC_{10}	TC_9	TC_8

Table 7-25 Read Register 15 - Bit Functions

Signal	RR15 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Zero Count Interrupt Enable							●	
DCD Interrupt Enable					●			
Sync/Hunt Interrupt Enable				●				

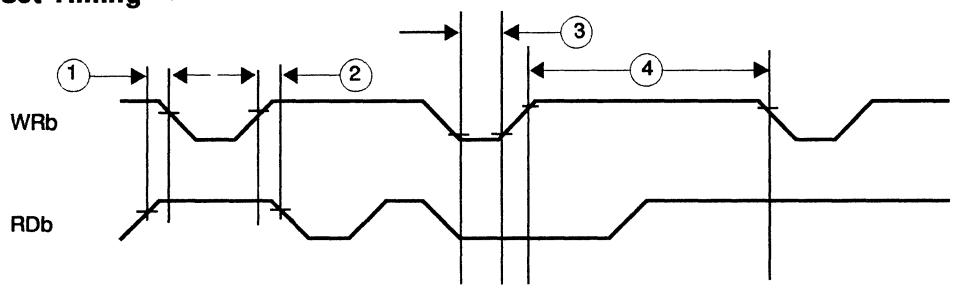
Table 7-25 Read Register 15 – Bit Functions

Signal	RR15 Register MSB ◄ LSB							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CTS Interrupt Enable			●					
TX Underrun/EOM Interrupt enable		●						
Break/Abort Interrupt Enable	●							
0						●		●

Switching Characteristics

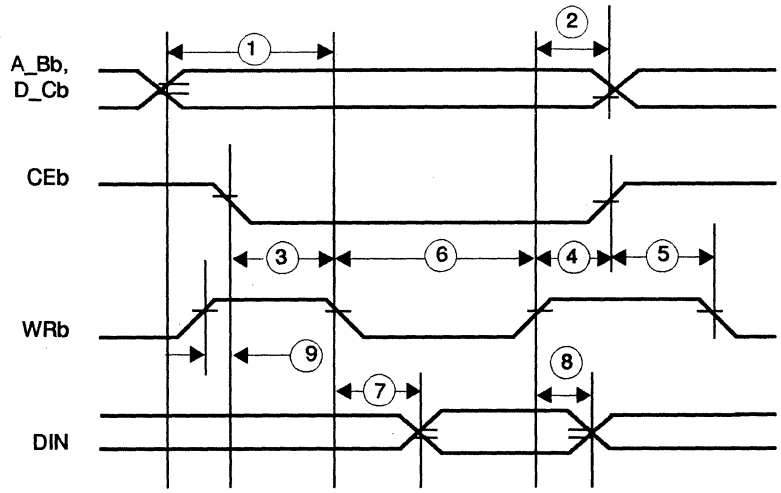
The following parameters apply strictly to the VS700H ASF. These timings are derived from pre-layout estimated capacitances. Post-layout timing variations may occur.

Reset Timing



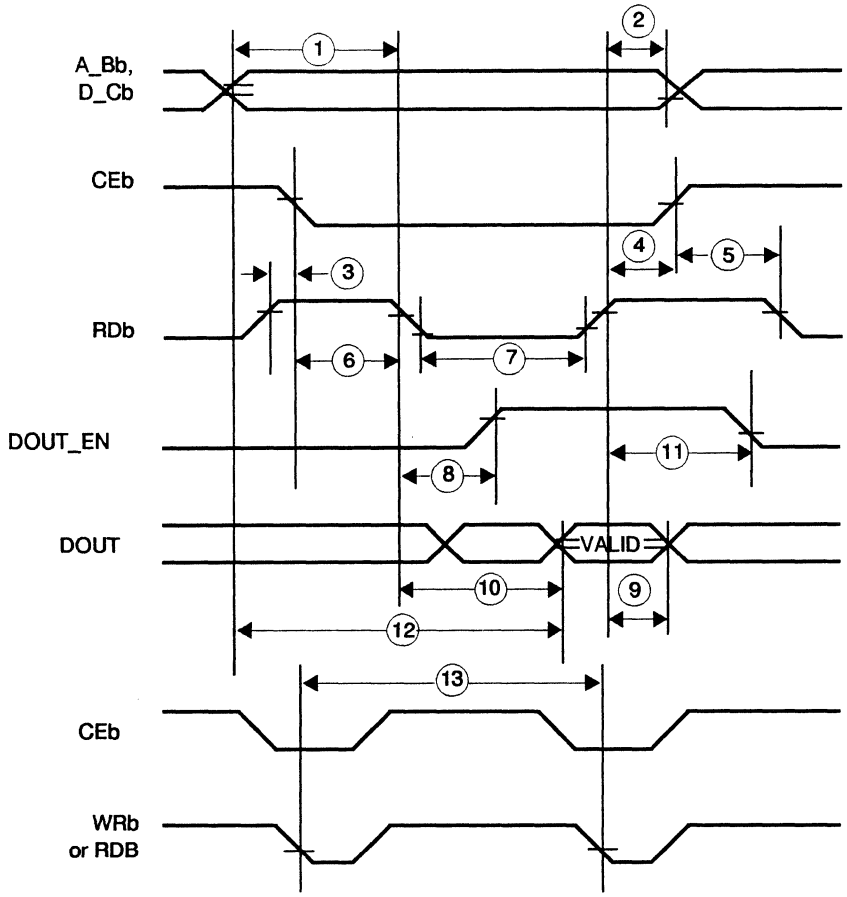
Temperature = 0° to 70° C, VCC = 5 V ± 10%, external load 1 pF					
No.	**Symbol**	**Description**	**Min.**	**Max.**	**Units**
1	TdRD(WR)	RDb ↑ to WRb ↓ Delay for No Reset	10		ns
2	TdWR(RD)	WRb ↑ to RDb ↓ Delay for No Reset	10		ns
3	TwRES	WRb and RDb Coincident LOW for Reset	50		ns
4	TsRES	Reset Recovery Time (WR or RD)	9		TcPC*
*NOTE: TcPC is the PCLK cycle Time.					

Write Timing



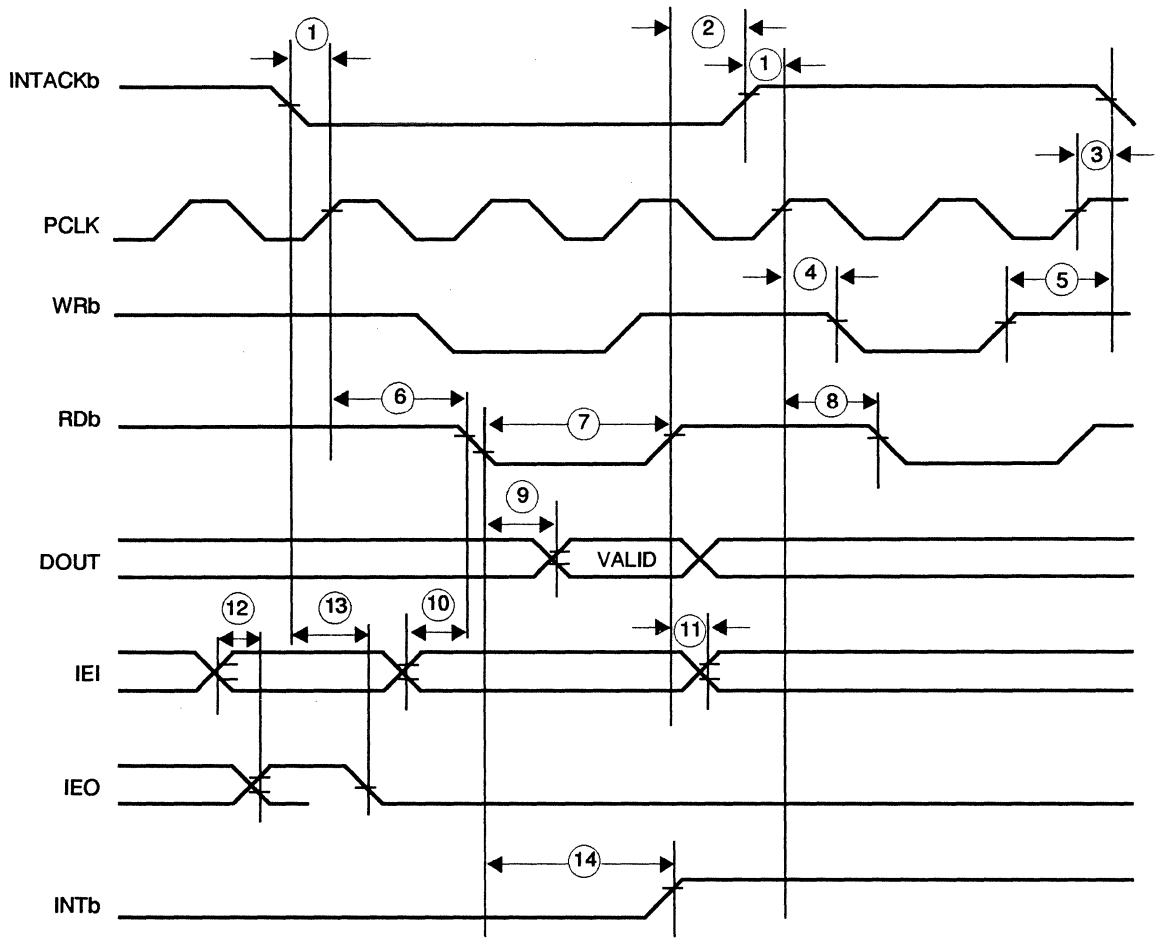
Temperature = 0° to 70° C, VCC = 5 V ± 10%, external load 1 pF					
No.	**Symbol**	**Description**	**Min.**	**Max.**	**Units**
1	TsA(WR)	Address to WRb ↓ Setup Time	30		ns
2	ThA(WR)	Address to WRb ↓ Hold Time	0		ns
3	TsCEI(WR)	CEb Low to WRb ↓ Setup Time	0		ns
4	ThCE(WR)	CEb to WRb ↑ Hold Time	0		ns
5	TsCEh(WR)	CEb HIGH to WRb ↓ Setup Time	30		ns
6	TwWRI	WRb LOW Width	50		ns
7	TdWRf(DW)	WR ↓ to Write Data Valid		35	ns
8	ThDW(WR)	Write Data to WRb ↑ Hold Time	0		ns
9	TsWRr(CE)	WRb ↑ to CEb ↓ Setup Time for No Write	0		ns

Read Timing



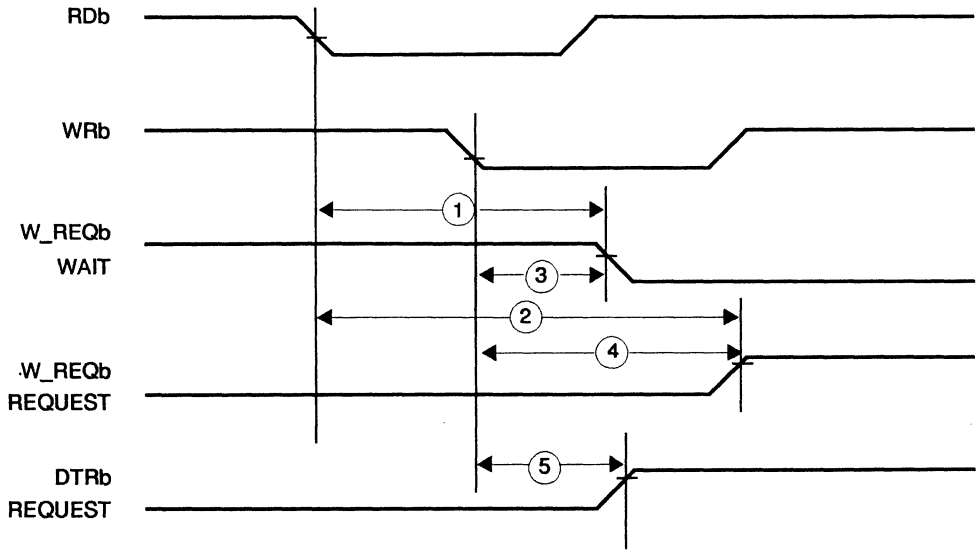
Temperature = 0° to 70° C, VCC = 5 V ± 10%, external load 1 pF					
No.	**Symbol**	**Description**	**Min.**	**Max.**	**Units**
1	TsA(RD)	Address to RDb ↓ Setup Time	30		ns
2	ThA(RD)	Address to RDb ↑ Hold Time	0		ns
3	TsRDr(CE)	RDb ↑ to CEb ↓ Setup Time for No Read	0		ns
4	ThCE(RD)	CEb to RDb ↑ Hold Time	0		ns
5	TsCEh(RD)	CEb HIGH to RDb ↓ Setup Time	30		ns
6	TsCEI(RD)	CEb LOW to RDb ↓ Setup Time	0		ns
7	TwRDI	RDb LOW Width	50		ns
8	TdRD(DRA)	RDb ↓ to DOUT_EN ↑	0		ns
9	TdRDr(DR)	RDb ↑ to Read Data Not Valid Delay	0		ns
10	TdRDf(DR)	RDb ↓ to Read Data Valid Delay		40	ns
11	TdRD(DRz)	RDb ↑ to DOUT_EN ↓		20	ns
12	TdA(DR)	Address Required Valid to Read Data Valid Delay		70	ns
13	TrC	RDb/WRb Recovery Time	4		TcPC*
*NOTE: TcPC is the PCLK cycle time.					

Interrupt Acknowledge Timing



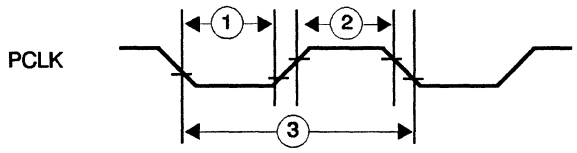
Temperature = 0° to 70° C, VCC = 5 V ± 10%, external load 1 pF					
No.	**Symbol**	**Description**	**Min.**	**Max.**	**Units**
1	TsIA	INTACKb to PCLK ↑ Setup Time	10		ns
2	ThIA(RD)	INTACKb to RDb ↑ Hold Time	0		ns
3	ThIA(PC)	INTACKb to PCLK ↑ Hold Time	10		ns
4	TsIA(WR)	INTACKb to WRb ↓ Setup Time	30		ns
5	ThIA(WR)	INTACKb to WRb ↑ Hold Time	0		ns
6	TdIA(RD)	INTACKb to RDb ↓ (Acknowledge) Delay	30		ns
7	TwRDA	RDb (Acknowledge) Width	50		ns
8	TsIAi(RD)	INTACK to RDb ↓ Setup Time	30		ns
9	TdRDA(DR)	RDb ↓ (Acknowledge) to Read Data Valid Delay		40	ns
10	TsIEI(RDA)	IEI to RDb (Acknowledge) Setup Time	30		ns
11	ThIEI(RDA)	IEI to RDb (Acknowledge) Hold Time	0		ns
12	TdIEI(IEO)	IEI to IEO Delay Time		40	ns
13	TdIA(IEO)	INTACKb to IEO Delay Time		40	
14	TdRDA(INT)	RDb ↓ to INTb Inactive Delay		40	

DMA Wait/Request Timing



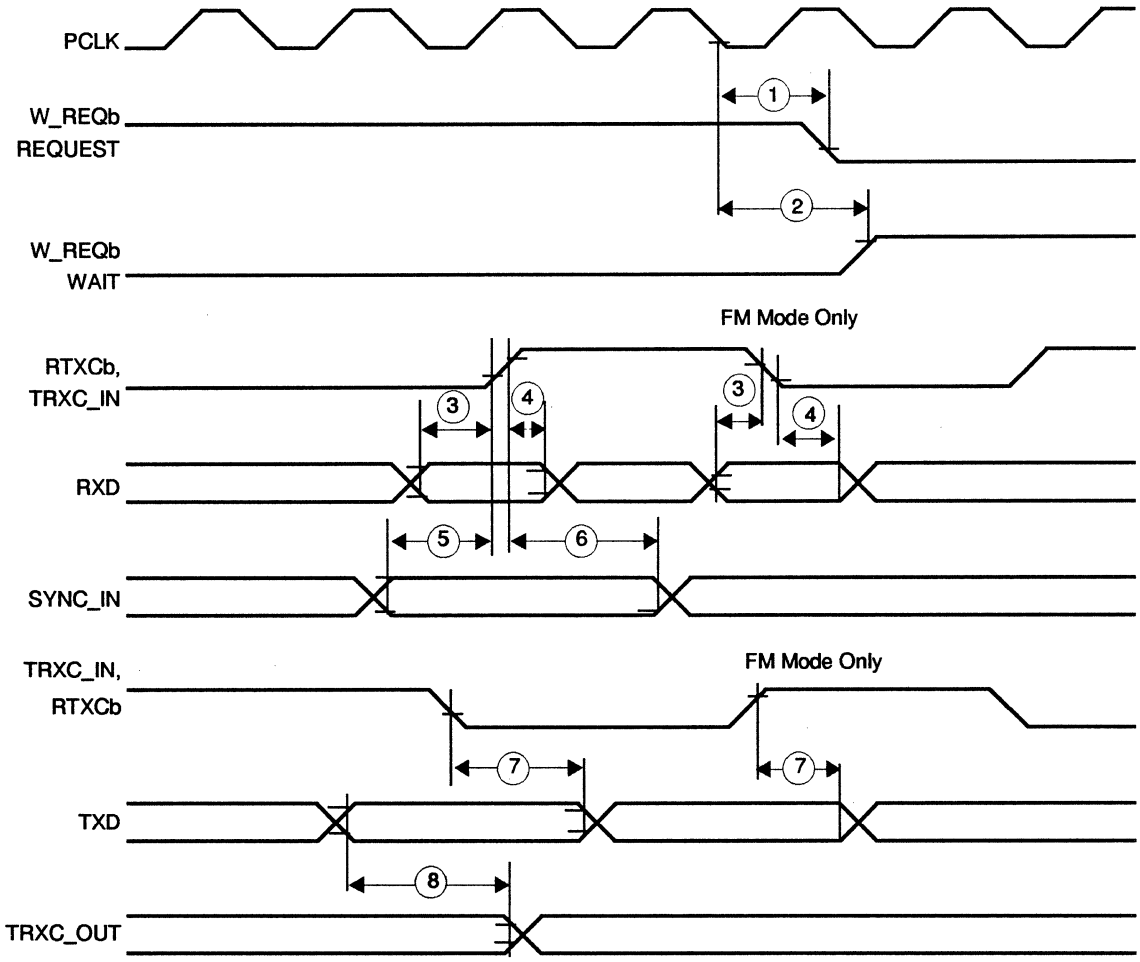
Temperature = 0° to 70° C, VCC = 5 V ± 10%, external load 1 pF					
No.	**Symbol**	**Description**	**Min.**	**Max.**	**Units**
1	TdRD(W)	RDb ↓ to Wait Valid Delay		40	ns
2	TdRD(REQ)	RDb ↓ to Request Not Valid Delay		40	ns
3	TdWR(W)	WRb ↓ to Wait Valid Delay		50	ns
4	TdWRf(REQ)	WRb ↓ to Request Not Valid Delay		50	ns
5	TdWRr(REQ)	WRb ↓ to DTRb Not Valid Delay		2	TcPC*
*NOTE: TcPC is the PCLK cycle time.					

Cycle Timing



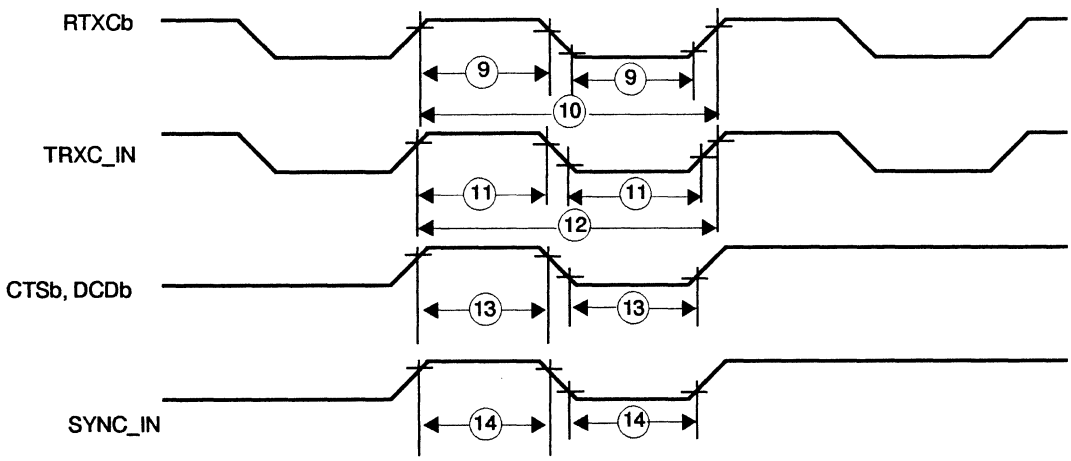
Temperature = 0° to 70° C, VCC = 5 V ± 10%, external load 1 pF					
No.	**Symbol**	**Description**	**Min.**	**Max.**	**Units**
1	TwPCI	PCLK LOW Width	50		ns
2	TwPCh	PCLK HIGH Width	50		ns
3	TcPC	PCLK Cycle Time	125		ns

General Timing



Temperature = 0° to 70° C, VCC = 5 V ± 10%, external load 1 pF					
No.	**Symbol**	**Description**	**Min.**	**Max.**	**Units**
1	TdPC(REQ)	PCLK ↓ to W_REQb Valid Delay		250	ns
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		250	ns
3	TsRXD(RXC)	RXD to RTXCb Setup Time	0		ns
4	ThRXD(RxC)	RXD to RTXCb Hold Time	150		ns
5	TsSY(RXC)	SYNC_IN to RTXCb Setup Time	–200		ns
6	ThSY(RXC)	SYNC_IN to RTXCb Hold Time	5		TcPC*
7	TdTXC(TXD)	TRXC_IN to TXD Delay		200	ns
8	TdTXD(TXC)	TXD to TRXC_OUT Delay		200	ns
*NOTE: TcPC is the PCLK cycle time.					

General Timing Continued



Temperature = 0° to 70° C, VCC = 5 V ± 10%, external load 1 pF					
No.	**Symbol**	**Description**	**Min.**	**Max.**	**Units**
9	TwRTX	RTXCb Width	150		ns
10	TcRTX	RTXCb Cycle Time	4		TcPC*
11	TwTRX	TRXC_IN Width	150		ns
12	TcTRX	TRXC_IN Cycle Time	4		TcPC*
13	TwEXT	DCDb or CTSb Width	200		ns
14	TwSY	SYNC_IN Width	200		ns
*NOTE: TcPC is the PCLK cycle time.					

Integrating the NCR85C30

There are several important issues to consider when integrating the NCR85C30 into a larger design. These issues are outlined below.

Bidirectional Signal Creation

The NCR85C30 Application Specific Function (ASF) has no bidirectional, tristate, or passively driven signals. Signals are provided, however, to allow for the creation of the bidirectional signals associated with the standard product. The figures below show one way to combine ASF signals to create the bidirectional data bus, TRXCb, and SYNCb signals, and the non-standard drive WAIT_REQb and INTRb signals.

Test Philosophy

To allow the usage of the test/simulation patterns provided with the NCR85C30, a specific multiplexed scheme must be used. The patterns provided assume the re-creation of bidirectional and open-drain signals at the pins of the device. All other signals must be multiplexed to the external pins for complete controllability and observability of the ASF. During the NCR85C30 test mode, the ASF must be muxed with the same interface as the standard product, plus SCANIN, SCANOUT, SCANSEL0, and TESTMODE. The multiplexer logic should add as little propagation delay as possible.

Initializing the NCR85C30

The standard product specifies that numerous bits in its register set are not initialized during reset. During simulations, this causes problems with propagating unknowns. However, a test mode reset is provided to bypass this problem. By asserting the TESTMODE signal high and the RDb and WRb signal low, all simultaneously, for longer than 50 ns, all flip-flops will be reset to a known state.

Functional Exception

The NCR85C30 synchronizes the Transmit Buffer Full signal to the back edge of TXCLK. This is a timing issue which results in a one-half bit time reduction in the transmitter service time.

Compatibility Verification Procedures

The design and verification of a high-level ASF is directed towards functional compatibility with the existing standard part. The ASF is designed and verified to match data sheet functionality (when possible), then additional steps are taken to match undocumented functionality.

Creation of a Compatibility Model

A compatibility model of the new ASF is generated from published sources of information on the standard part including functional specifications, data sheets, application notes, review sheets and errata sheets. Exceptions to standard part behavior or function will be identified.

Utilization of Industry Consultants

NCR utilizes the advice of industry consultants who are highly knowledgeable about the operation of the standard part. This additional data includes features and functions which would be apparent to someone who is very familiar with the functionality of the part, but would not necessarily be covered in available documentation.

Use of Hardware Modeling Techniques and Extraction of System Patterns for Design Verification

Compatibility test vectors for the part are developed from two sources. First, a device model is created on a hardware modeling system (HML) to create test vectors based on the actual operation of the industry standard part. A second set of vectors is generated by extracting functional and timing information from the compatibility documentation and creating patterns based on this information.

In-system Test and Firmware Verification

To ensure that the ASF is truly representative of the standard part's functionality, a system test is executed on new sample parts. The test includes the following system parameters to detect any incompatibilities or inconsistencies:

- Selection of controllers and systems
- Special test programs
- System diagnostics
- Selection of operating systems
- Selection of application programs

Final Test Plan

Throughout the compatibility verification process, final test procedures for the ASF are emphasized. The NCR compatibility procedures provide assurance that the ASF will function as the standard part does with documented exceptions where warranted.

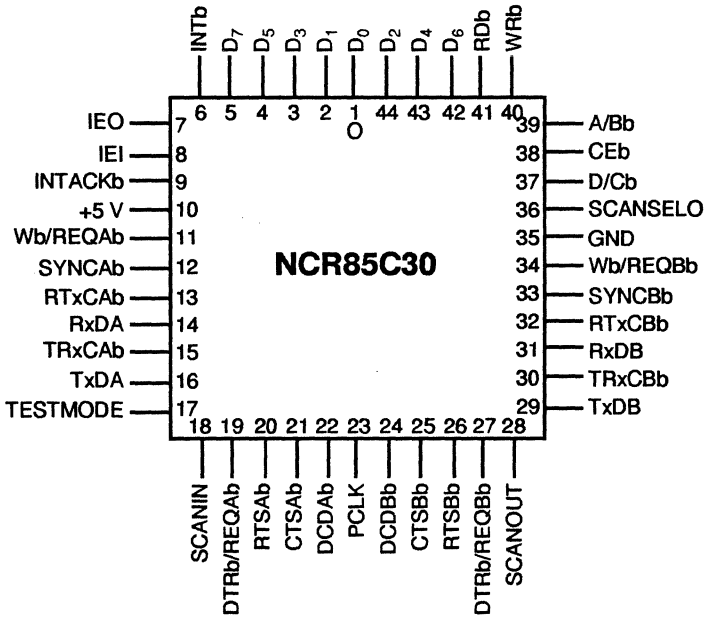
Kit Parts

Provision of kit parts that are plug-compatible with the existing standard part is a very important aspect of in-system verification. The availability of kit parts allows the customer to verify ASF operation and compatibility with the existing system design and software. Kit parts also provide a convenient way to bread-board "System Solutions in Silicon."

Kit Part Exception

The NCR85C30 kit part does not support the placement of a crystal oscillator across the SYNCb and RTxCb pins. For evaluation, an external clock can be created and input directly to the SYNCb pin.

SCC Kit-Part Pinout



NCR82077 Floppy Disk Controller Core
Contents

NCR82077 Floppy Disk Controller Core

Features

- ASIC core cell compatible with 82077AA-1
  - PC/AT® compatible
  - PS/2® compatible
  - PS/2 Model 30 compatible
- Integrated digital data separator
  - 250 Kbits/second
  - 300 Kbits/second
  - 500 Kbits/second
  - 1Mbits/second
- 16-byte FIFO
- High-speed processor interface
- Perpendicular recording support
- Four fully decoded drive select and motor signals
- Programmable write precompensation delays
- Addresses 256 tracks directly, supports unlimited tracks
- Kit part package type: 68-pin PLCC
- MFM encoding format

Product Description

Designed with NCR's VS700H submicron CMOS cell-based technology, the 82077AA compatible floppy disk controller core provides floppy disk control for the PC/AT and PS/2.

Integrated internal logic functions eliminate external compensation and/or auxiliary logic requirements. Programmable thresholds, hardware register compatibility, and selectable polarity are just some of the other useful features this core provides.

Upon reset, the 82077AA-compatible core defaults to NCR765 functionality. New software commands or hardware straps are used to access the expanded functions.

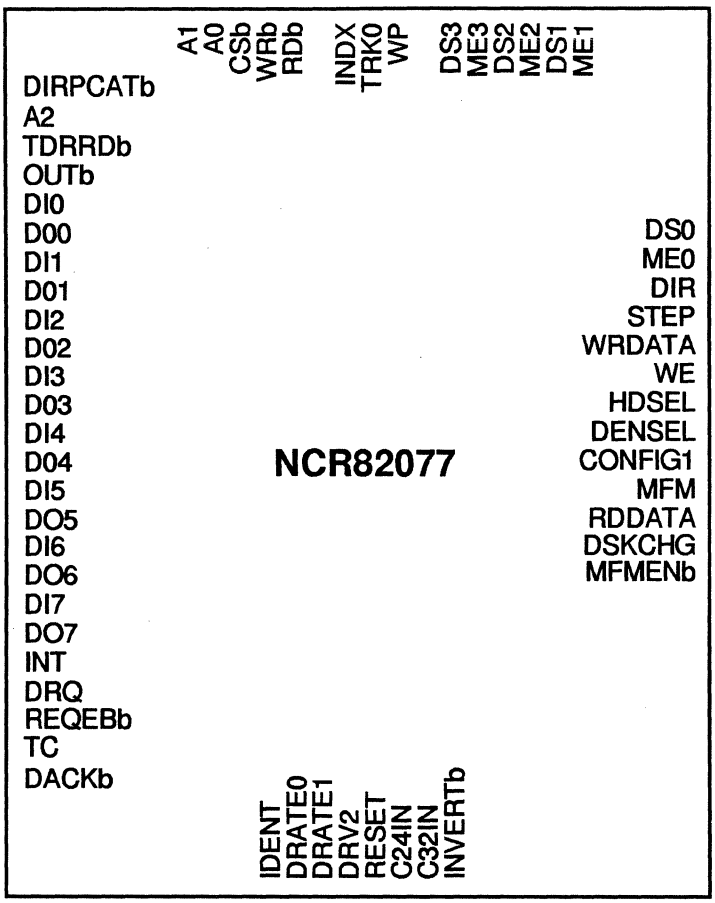


Figure 8-1 Symbol

NOTE: The use of the lowercase "b" at the end of a pin name indicates that the signal is asserted low.

Inputs/Outputs Tables

Table 8-1 Host Interface Pins

<table>
<tr><th>Pin Name</th><th>Type</th><th colspan="5">Description</th></tr>
<tr><td colspan="7">Host Interface</td></tr>
<tr><td rowspan="11">A0-A2</td><td rowspan="11">I</td><td colspan="5">Address. By using the appropriate 3-bit combination from the following table, the desired host interface register can be selected:</td></tr>
<tr><td>A2</td><td>A1</td><td>A0</td><td></td><td>Register</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>R</td><td>Status Register A</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>R</td><td>Status Register B</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>R/W</td><td>Digital Output Register</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>R/W</td><td>Tape Drive Register</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>R</td><td>Main Status Register</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>W</td><td>Data Rate Select Register</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>R/W</td><td>Data (FIFO)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td></td><td>Reserved (NCR Test Mode Enable)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>R</td><td>Digital Input Register</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>W</td><td>Configuration Control Register</td></tr>
<tr><td>CSb</td><td>I</td><td colspan="5">Chip Select. A low on this pin is used to select the device. This qualifies the RDb and WRb inputs.</td></tr>
<tr><td>C24IN</td><td>I</td><td colspan="5">Clock, 24 MHz. This pin serves as a 24 MHz oscillator connection.</td></tr>
<tr><td>C32IN</td><td>I</td><td colspan="5">Clock, 32 MHz. This pin serves as a 32 MHz oscillator connection.</td></tr>
<tr><td>DACKb</td><td>I</td><td colspan="5">DMA Acknowledge. Driving DACKb low during DMA cycles qualifies the RDb and WRb inputs for data accesses. For both PC/AT and PS/2 Model 30 modes, DMAGATEb must also be set high in the Digital Output Register or DACKb will not be recognized.</td></tr>
<tr><td>DI0–DI7</td><td>I</td><td colspan="5">Data Bus In. Data is written on this bus to program the device and transfer data.</td></tr>
<tr><td>DO0–DO7</td><td>O</td><td colspan="5">Data Bus Out. Data is read on this bus to return status and transfer data.</td></tr>
<tr><td>DRQ</td><td>O</td><td colspan="5">DMA Request. Generates a request for a DMA controller. In both PC/AT and PS/2 Model 30 modes, setting DMAGATEb low in the Digital Output Register will place this pin in a state of high impedance.</td></tr>
</table>

Table 8-1 Host Interface Pins (Continued)

Pin Name	Type	Description
Host Interface		
IDENT	I	Identity. The IDENT pin is used during reset to select the mode in which the device is to operate (PC/AT, PS/2, PS/2 Model 30). After reset, IDENT corresponds to the type of drive selected and affects the polarity of DENSEL. Refer to the *Host Interface IDENTITY Addendum* section for greater detail.
CONFIG	I	Configure. This is the MFM related core input signal. The value of this signal is used as a configuration aid in conjunction with the IDENT pin during a hardware RESET.
INT	0	Interrupt. INT is set high to indicate that the device is ready to return status bytes or transfer data bytes (in non-DMA operation). In both PC/AT and PS/2 Model 30 modes, setting DMAGATEb low in the Digital Output Register will place this pin in a state of high impedance.
OUTb	O	Data Bus Enable. This signal is used to drive DO0–DO7 onto a bidirectional data bus. Refer to the *Application Notes* section of this data sheet for a method of creating a bidirectional data bus.
DIRPCATb	O	Data Bus Enable. This signal enables the data bus during a read of the Digital Input Register in PC/AT mode. Refer to the *Application Notes* section later in this data sheet for a complete description.
TRRDb	O	Data Bus Enable. This signal enables the data bus during an access to the Tape Drive Register. Refer to the *Application Notes* section later in this data sheet for a complete description.
RDb	I	Read. This signal is used in conjunction with CSb or DACKb to access device registers.
RESET	I	Reset. When driven high, the device is initialized. With the exception of those set by the Specify command, all of the registers are cleared.
REQENb	O	Request Enable. In PC/AT or PS/2 Model 30 modes, setting DMAGATEb low will force REQENb high, placing INT and DRQ in a state of high impedance.
TC	I	Terminal Count. Driving the TC pin high while driving DACKb low during a disk data transfer will signal that the final data byte has been transferred and the operation can be terminated.
WRb	I	Write. This signal is used in conjunction with CSb or DACKb to write device registers.

Table 8-2 Disk Control Pins

Pin Name	Type	Description
Disk Control		
DENSEL	O	Density Select. This output is used to indicate the data rate selection as shown in the *Host Interface IDENTITY Addendum* section of this document.
DIR	O	Direction. DIR controls the direction that the head on the disk drive moves in response to a STEP pulse. When DIR is high, the head will move toward the inner tracks.
DRV2	I	Drive2. This signal indicates the presence of a second drive. The state of this pin can be accessed in Status Register A.
DSKCHG	I	Disk Change. This signal is used to indicate that the disk in a drive has been changed. The state of this pin can be accessed in the Digital Input Register.
DS0, DS1, DS2, DS3	O	Drive Select 0–3. These outputs are the decoded drive select signals (drives 0–3). A drive is selected by programming the Digital Output Register with the desired drive.
HDSEL	O	Head Select. This output is used for disk side selection. A high selects side 1, while a low selects side 0.
INDX	I	Index. This signal reflects the detection of the index hole from a floppy, indicating the beginning of a track.
INVERTb	I	Invert. This input is used to determine the polarity of the disk interface signals. Strapping this pin low will make the disk control signals active low, allowing the internal buffers and drivers to be used. Strapping this pin high will make the disk control signals active high, requiring inverting receivers and drivers external to the device.
ME0, ME1, ME2, ME3	O	ME0–3. These outputs are the decoded motor enable signals (drives 0–3). These pins are set directly by writing the Digital Output Register.
STEP	O	Step. This output provides step pulses, to the drive, that move the head to a new track.
TRK0	I	Track0. This signal indicates that the head is at track 0.
WE	O	Write Enable. This output enables the selected drive to write data to the disk.
WP	I	Write Protect. Indicates if the disk in the selected drive is write protected.

Table 8-2 Disk Control Pins (Continued)

Pin Name	Type	Description
Disk Control		
WRDATA	O	Write Data. This output carries the MFM encoded serial data that is to be written to the disk when WE is active.

Table 8-3 PLL Interface Pins

Pin Name	Type	Description
PLL Interface		
DRATE0 DRATE1	O	DATA RATE 0–1. These pins indicate the data rate at which the part is programmed to operate. The data rate is programmed in the Data Rate Select Register, bits 1 and 0.
MFM	O	MFM. This is the output of the standard bidirectional MFM signal. This output can be combined with the CONFIG pin to recreate the MFM pin on the industry standard part. Because only MFM mode is supported by the NCR82077AA-1, the value of MFM will always be 1.
MFMENb	O	MFM Enable. This output signal is used for MFM/ CONFIG bidirectional control when they are combined to form one signal.
RDDATA	I	Read Data. This input carries the MFM encoded serial data that is coming from the disk when the device is trying to read.

Input Signal Capacitance		Output Signal Drive Capability	
RESET	SBUF	DO0–DO7	MBUF
CSb	INV3	OUTb	AND2
A0–A2	MBUF	DIRPCATb	NAN2
DI0–DI7	INV	TDRRDb	OR2
RDb	MBUF	DRQ	MBUF
WRb	MBUF	INT	MBUF
DACKb	OR2	REQENb	MBUF
TC	EXOR	MFM	NAN2
C24IN	INV	MFMENb	INVH
C32IN	INV	DRATE0–DRATE1	HBUF
IDENT	SBUF	ME0–ME3	HBUF
RDDATA	MBUF	DS0–DS3	HBUF
CONFIG	SBUF	HDSEL	HBUF
INVERTb	INV8	STEP	HBUF
DSKCHG	EXNOR	DIR	HBUF
DRV2	MUX2, INV	WRDATA	HBUF
TRK0	EXOR, EXOR	WE	HBUF
WP	EXOR, EXOR	DENSEL	HBUF
INDX	EXOR, EXOR		

Host Interface IDENTITY Addendum

During hardware reset, the IDENT and CONFIG pins are used to select one of three interface modes according to the Interface Configurations table that follows.

Table 8-4 Interface Configurations

IDENT Setting	CONFIG Setting	Interface Mode	Interface Mode Major Options		
			DMA Gate Logic	TC	Status Registers A and B
0	0	PS/2 Model 30	Enabled	Active High	Available
0	1 or NC	PS/2	Ignored	Active Low	Available
1	0	——	Not Supported	Not Supported	Not Supported
1	1 or NC	AT	Enabled	Active High	Not Available

After hardware reset, the IDENT pin performs two additional functions: (1) selects drive type and (2) determines polarity of the DENSEL pin. The state of the DENSEL pin is shown in the following table:

Table 8-5 DENSEL Table

Data Rate	IDENT	DENSEL
250 Kbps	0	1
	1	0
300 Kbps	0	1
	1	0
500 Kbps	0	0
	1	1
1 Mbps	0	0
	1	1

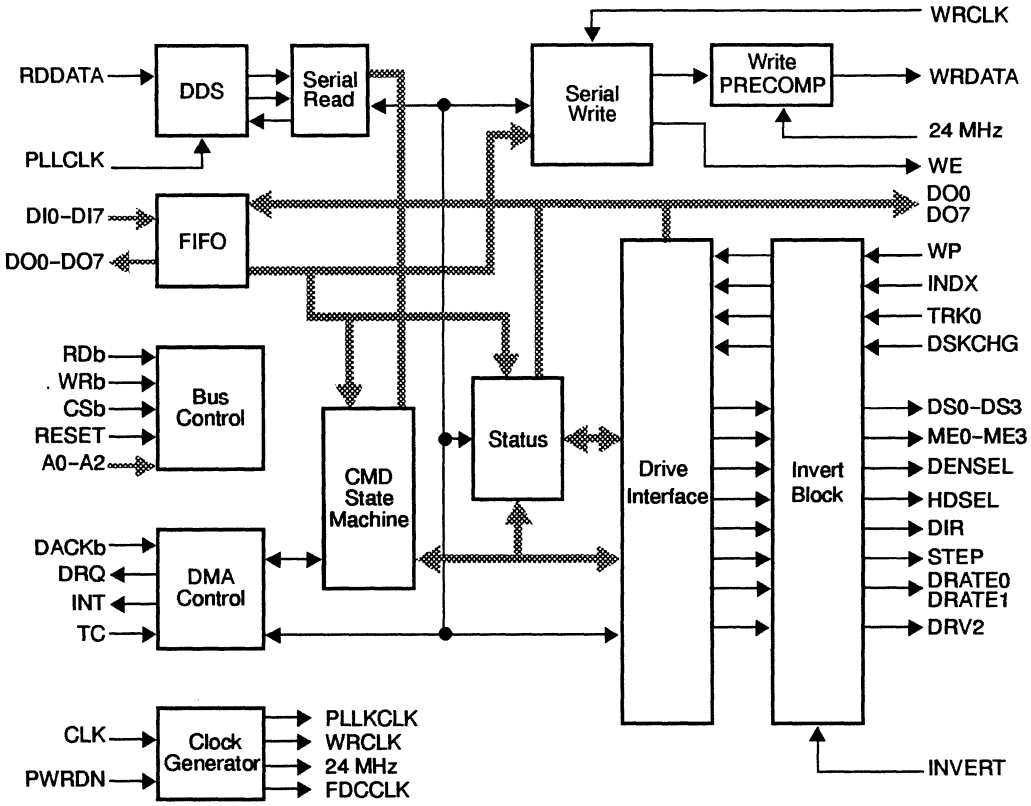


Figure 8-2 NCR82077 Floppy Controller Core Block Diagram

Architectural Functionality

Functional Blocks

This section briefly describes all of the major functional blocks of the NCR82077 Floppy Disk Controller Core (FDCC) design as illustrated in Figure 8-2.

The 82077 has three command handling phases. They are: (1) Command, (2) Execution, and (3) Result. The Command phase inputs up to nine bytes needed to program the 82077 to execute the command. The Execution phase is where the operation occurs, transferring data, moving the disk head, or obtaining status. The Result phase can output up to 10 status bytes depending on the command.

When writing command bytes to or reading result bytes from the controller, it is necessary to poll the Main Status Register, checking the RQM and DIO bits, to make sure that the controller is ready for another byte transfer. When the RQM bit is set, a transfer may occur. The DIO bit indicates the direction (read or write) of the transfer.

FIFO

In order to allow the NCR82077 to be used in systems with long DMA and interrupt response times, a 16 byte FIFO with a programmable threshold value is included and can be optionally used as a buffer between the controller and the host. The FIFO is only used for transferring disk data during the Execution phase of a command. During the Command and Result phases, host accesses to the data register bypass the FIFO and talk directly to the core. The FIFO is cleared when the part exits the Execution phase in order to prepare it for the next data transfer.

When the NCR82077 comes out of reset, the FIFO is automatically disabled to maintain compatibility with earlier versions of the floppy disk controller. If desired, the FIFO can be activated using the EFIFO and FIFOTHR parameters of the Configure command. This bit defaults to 1 upon reset of the part. The threshold of the FIFO is programmed with FIFOTHR, and will directly affect the behavior of read and write commands.

For any of the read commands that use the FIFO, FIFOTHR corresponds to the number of bytes that the controller can place in the FIFO before the FIFO is filled. If no bytes are removed, then trying to transfer the FIFOTHR + 1 byte will cause the FIFO to overrun. In addition to reaching the threshold, transferring the last byte of a sector into the FIFO will also cause the request signal to be set. Read commands which use the FIFO include Read Data, Read Deleted Data, and Read a Track.

For any of the write commands that use the FIFO, FIFOTHR corresponds to the number of data bytes that the controller can take out of the FIFO before the FIFO is empty. If the host has not placed any bytes in the FIFO, then an underrun will occur when the controller tries to access the FIFOTHR + 1 byte. When the FIFO contains FIFOTHR or fewer bytes, the request pin (DRQ or INT) is set to indicate that it is time for the host to transfer bytes to the FIFO. Write commands that use the FIFO include: Write Data, Write Deleted Data, and Format a Track.

Once an INT or DRQ has been set, it remains set until the FIFO has either been emptied, in the case of a read, or filled, in the case of a write. At that time the request is reset and held reset until the threshold condition is met again. If the host sets the TC signal to indicate that the host is finished transferring bytes, the request is removed and will remain off for the duration of the command. The flow charts in Figures C-3 and C-4 illustrate the conditions for setting and resetting the requests for reads and writes.

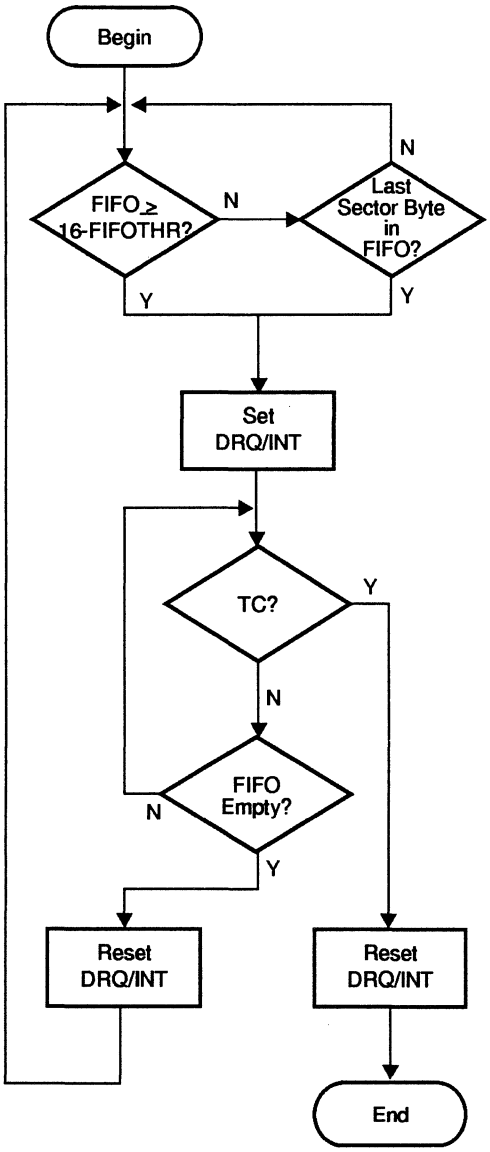


Figure 8-3 Read Commands (Read Data, Read Deleted Data, Read a Track)

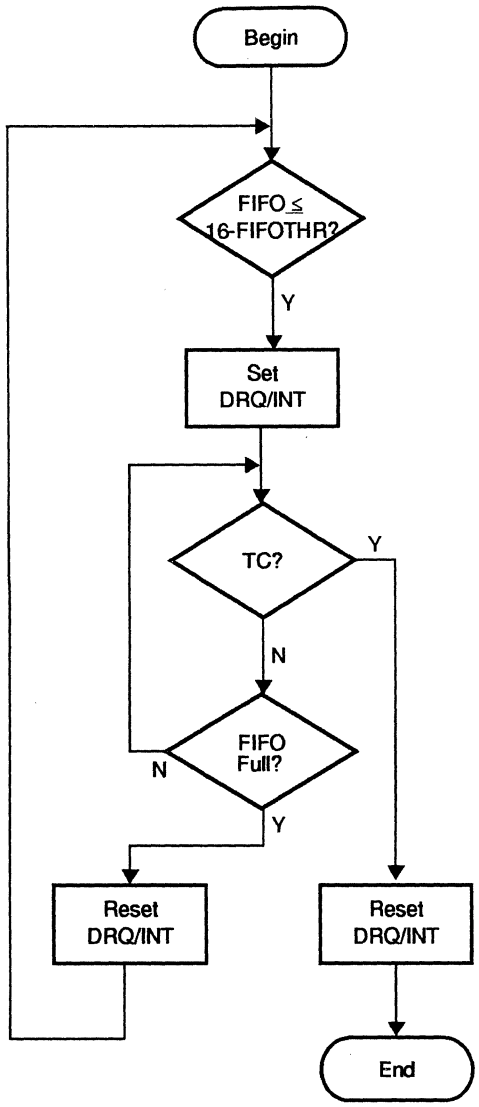


Figure 8-4 Write Commands (Write Data, Write Deleted Data, Format Track)

In choosing the value for FIFOTHR, attention should be paid to the ability of the system being designed to respond to the transfer request. A lower value would be selected for a system with a fast response time, resulting in fewer actual requests for each data transfer. By doing this, more bytes can be transferred during each request. The risk is that with less time to respond to an interrupt, there is a greater chance of over/under running the FIFO. Selecting a higher value would be appropriate for systems which have a longer response time. Fewer bytes will be transferred during each request, but the risk of over/under running the FIFO is reduced.

The maximum response time allowed can be calculated using the value programmed into FIFOTHR. The equation would be:

$$\text{Maximum response time} = (\text{FIFOTHR} + 1) \times \left|\frac{1}{8 \times \text{Data Rate}}\right|,$$

where Data Rate is the highest disk data rate that will be supported by the system. For calculating the maximum response time in a given system, it is necessary to use the highest data rate because it will correspond to the longest allowable response time. Table 8-6 shows the conditions for setting the request and the maximum response times associated with different values of FIFOTHR.

To prevent the FIFO from being disabled during a software reset (via the DOR or DSR), the Lock command is included in the NCR82077. The Lock command sets or resets the LOCK bit, which will prevent software resets from affecting the values of either EFIFO or FIFOTHR when set. A hardware reset (via the RESET pin) will clear the LOCK bit out and reset the FIFO configuration values.

Table 8-6 FIFOTHR Table

FIFOTHR (Hex)	Request Set		Maximum Response Time (μs)			
	Read	Write	500 Kbps	300 Kbps	250 Kbps	1 Mbps
0	= 16	= 0	16	26	32	8
8	≥8	≤8	144	240	288	72
F	≥1	≤15	256	426	512	128

Clock Generator

This functional block is internal to the chip, unlike the NCR765. A 24 MHz clock is supplied by the user. This clock is the reference for all internal timings. For a 1 Mbps operation, a 32 MHz clock is required to provide precision quanitization for the internal Digital Data Separator.

Write Precompensation

Within the Data Rate Select register, bits 4, 3, and 2 are used to modify the WRDATA output that is written to the disk to compensate for bit shifting. Bit shifting is jitter of the bit positions when reading from the disk caused by inherent properties of magnetic media.

This section of logic shifts the data to provide write precompensation for the floppy disk drive. The block contains a 13-bit shift register and a mux to select varying degrees of early, normal, or late bit shift during recording of the magnetic media. Otherwise, the bit jitter could potentially manifest itself as read errors when the data is retrieved from the magnetic media.

Data Separator

The data separator locks onto the data stream coming in from the disk drive and recovers the embedded clock signal. The RDDATA pin transfers a serial MFM encoded data stream which, when fed to the data separator, is used to create an internal clock signal known as the Read Data Window (RDW). Once this clock is recovered, both RDW and RDDATA go to the Serial Read block, where they drive the serial to parallel

conversion logic. RDW is a periodic signal which frames the serial data bits on RDDATA and runs at a frequency equal to the data rate. For each bit in the data stream, RDW has a high and a low phase. One phase frames what is referred to as a data bit and the other phase frames what is referred to as a clock bit. Through the presence or absence of data and clock bits, the actual data to be transferred is encoded in the data stream. The Serial Read logic converts this data stream into bytes used for sector identification, data transfer, and CRC error checks.

The NCR82077 incorporates a digital data separator (DDS) rather than follow the traditional analog phase-locked loop approach. Because of this, the NCR82077 is entirely a digital circuit, eliminating the need for any external filtering.

The DDS has two modes of operation: high gain and low gain. Initially, while trying to synchronize to the data stream, the DDS operates in the high gain mode where it is capable of adjusting rapidly to the incoming data stream. Once the DDS has successfully locked on the data stream and a synchronization field is detected, it shifts into the low gain mode. In the low gain mode, the DDS maintains a lock on the data stream and makes gradual corrections as necessary to adjust to the speed variations which can occur in the data stream.

The DDS samples the incoming data stream at 32 times the data rate. In order to do this at the 1 Mbps data rate, it is necessary to provide a 32 MHz clock to the cell. If the target design system will never need to support the 1 Mbps data rate, the 32 MHz clock can be left out of the design. In the kit part, this pin can be left as a no-connect. When designing an ASIC with the NCR82077 application specific function where the 1 Mbps data rate will not be used, the pin C32IN on the symbol should be tied low.

Bus Control Logic

The bus control logic controls the interface between the system and the 82077. The bus control logic uses the address inputs, chip select, and the read and write signals to select which register is being accessed. The bus state machine determines which of the phases, Command, Execution, or Result, the controller is in. The count select and phase compare logic controls the number of bytes written during the command phase and the number of bytes read during the result phase. The bus control logic is designed to provide access to the test mode register, which is specific to the NCR version of this part.

DMA Control Logic

The DMA control logic provides the timing and control of the data transfers during the execution phase when the FIFO is disabled. The block provides the overrun/underrun timing, controls the DRQ or INT signals and responds to a DACKb or reads/writes to the data register (FIFO). The DMA control block also forces a zero fill for writes less than a full sector.

Command State Machine Logic

The command state machine block stores internal register values, provides command decoding, and controls the serial read, serial write, status, and drive interface blocks. The command decode function stores the first byte of the command phase and decodes which command is to be executed. The remaining bytes of the command phase are stored in the internal data registers. These registers are used to direct the part in the Execution phase and to provide the proper response values during the Result phase.

Other registers store the data control for the commands in the Execution phase. The command sequencer stores error flags and reports them during the Result phase. The implied seek control, if enabled, initiates an implied seek as soon as the Execution phase starts. The normal data transfer operation is started at the completion of the implied seek. The Command state machine block has the clock divider for the remainder of the core logic. The clock divider provides timing for the drive interface block and some of the error condition checks. The Command state machine also contains the specify registers.

Status Logic

The status logic is used to output status during the result phase. The status registers store the results of seek commands, recalibrate commands, and poll interrupts, and reports the status of these commands as well as data transfers. The status logic also contains the value for the main status register.

Drive Interface Logic

The drive interface logic is used to control the head (side) selection and the seek or recalibrate operations. The poll logic included here controls seek and recalibrate functions by issuing step pulses and checking for seek complete. The poll logic also sets the initial power up interrupt unless disabled before the interrupt occurs. The timing for the drive interface signals is generated in the drive timing block. The VCO enable block is used to control the read logic and DDS. The input signals from the drive interface is synchronized with the internal clock in this section.

Invert Logic

The invert logic monitors the invert pin and changes the polarity of the signals to and from the drive.

Serial Read Logic

The serial read logic is used to decode the serial MFM data stream from the DDS. The decoded data is checked for address marks, data marks, and CRC. The serial data stream is converted to bytes and output on the bus if it is a read operation. The read sequencer stores the data in a holding latch and initiates a data transfer to either the host or the FIFO. The read sequencer signals the completion of an ID field and a data field. It indicates an ID match for both the read and write commands.

Serial Write Logic

The serial write logic provides MFM encoded data to the write precompensation logic. The serial write logic also provides early and late flags to selectively control the precompensation circuit. The timing block initiates a write or format operation. The write sequencer controls a write data operation. When an ID match is received, the write sequencer starts loading the write shift register with either gap characters or sync characters. The data is shifted out at write clock time. The write sequencer outputs data when the data mark field is written and calculates CRC for the sector. The CRC is appended to the data stream and the command sequencer is signaled.

The format state machine controls a format operation. The state machine waits for the rising edge of index pulse and starts outputting the index address mark, gap, ID address mark, gap, data mark, format data, gap, ID, gap, data, etc. until the sector number equals the sector count. The state machine waits for the next rising edge of index pulse and signals the command sequencer at completion.

Commands

Unsupported Features

NCR does not support FM recording or scan commands. When performing read and write operations, the part will assume MFM encoding regardless of the value written to the MFM bit. DTL (Data Length) is only supported as SC in the Verify command. In all other commands the byte must be written, but does not have an effect on the part. The parameter N cannot be written with a 0.

Read Data Command

This command causes the controller to read a given number of bytes (by sector) from the diskette and transfer them to the system.

Command Phase — The microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	MT	MFM	SK	0	0	1	1	0
Byte 1	0	0	0	0	0	HDS	DS1	DS0
Byte 2	C							
Byte 3	H							
Byte 4	R							
Byte 5	N							
Byte 6	EOT							
Byte 7	GPL							
Byte 8	DTL							

Execution Phase — The controller reads data from the diskette and transfers it to the system. The controller interrupts when the Execution phase ends.

Result Phase — The system microprocessor reads the following result bytes from the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	ST0							
Byte 1	ST1							
Byte 2	ST2							
Byte 3	C							
Byte 4	H							
Byte 5	R							
Byte 6	N							

Read Deleted Data Command

This command is identical to the Read Data command, except it only reads sectors with data fields containing a Deleted Data Address Mark.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	MT	MFM	SK	0	1	1	0	0
Byte 1	0	0	0	0	0	HDS	DS1	DS0
Byte 2	C							
Byte 3	H							
Byte 4	R							
Byte 5	N							
Byte 6	EOT							
Byte 7	GPL							
Byte 8	DTL							

Execution Phase — The controller reads data from the diskette and transfers it to the system. The controller interrupts when execution phase ends.

Result Phase — The system microprocessor reads the following result bytes from the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	ST0							
Byte 1	ST1							
Byte 2	ST2							
Byte 3	C							
Byte 4	H							
Byte 5	R							
Byte 6	N							

Write Data Command

This command writes a given number of bytes (by sector) to the diskette from the system microprocessor or DMA controller.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	MT	MFM	0	0	0	1	0	1
Byte 1	0	0	0	0	0	HDS	DS1	DS0
Byte 2	C							
Byte 3	H							
Byte 4	R							
Byte 5	N							
Byte 6	EOT							
Byte 7	GPL							
Byte 8	DTL							

Execution Phase — Controller transfers data from system and writes it to the diskette. The controller interrupts when execution phase ends.

Result Phase — System microprocessor reads the following result bytes from the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	ST0							
Byte 1	ST1							
Byte 2	ST2							
Byte 3	C							
Byte 4	H							
Byte 5	R							
Byte 6	N							

Write Deleted Data Command

This command is identical to the Write Data command, except it writes a Deleted Data Address Mark at the beginning of the data field instead of a Data Address Mark.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	MT	MFM	0	0	1	0	0	1
Byte 1	0	0	0	0	0	HDS	DS1	DS0
Byte 2	C							
Byte 3	H							
Byte 4	R							
Byte 5	N							
Byte 6	EOT							
Byte 7	GPL							
Byte 8	DTL							

Execution Phase — Controller transfers data from the system and writes it to the diskette. The controller interrupts when execution phase ends.

Result Phase — System microprocessor reads the following result bytes from the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	ST0							
Byte 1	ST1							
Byte 2	ST2							
Byte 3	C							
Byte 4	H							
Byte 5	R							
Byte 6	N							

Read a Track Command

This command operates much like the Read Data command, except that all the sectors of a track are read as one large data field, regardless of the logical sector order.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	0	MFM	0	0	0	0	1	0
Byte 1	0	0	0	0	0	HDS	DS1	DS0
Byte 2	C							
Byte 3	H							
Byte 4	R							
Byte 5	N							
Byte 6	EOT							
Byte 7	GPL							
Byte 8	DTL							

Execution Phase — Controller reads data from the diskette and transfers it to the system. The controller interrupts when execution phase ends.

Result Phase — System microprocessor reads the following result from the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	ST0							
Byte 1	ST1							
Byte 2	ST2							
Byte 3	C							
Byte 4	H							
Byte 5	R							
Byte 6	N							

Read ID Command

This command locates the current position of the disk drive heads.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	0	MFM	0	0	1	0	1	0
Byte 1	0	0	0	0	0	HDS	DS1	DS0

Execution Phase — The controller reads ID information from the diskette. The controller interrupts when the execution phase ends.

Result Phase — System microprocessor reads the following result bytes from the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	ST0							
Byte 1	ST1							
Byte 2	ST2							
Byte 3	C							
Byte 4	H							
Byte 5	R							
Byte 6	N							

Format a Track Command

This command formats one track of a disk, one sector at a time. After each sector is formatted with the given gaps, address marks, ID information, and data, the host sends the floppy controller the logical sector number for the next sector of the track. This facilitates non-sequential sector addressing (interleaving).

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	0	MFM	0	0	1	1	0	1
Byte 1	0	0	0	0	0	HDS	DS1	DS0
Byte 2	N							
Byte 3	SC							
Byte 4	GPL							
Byte 5	D							

Execution Phase — The controller formats an entire track using four data bytes (supplied by the system) for each sector. The data bytes required from the system for each sector are: cylinder, head, sector number, and sector size (C, H, R, N) in that order. The particular values that will be written to the gap and data field are controlled by the values programmed for N, SC, GPL, and D during the command phase. The controller interrupts at the end of the execution phase.

Result Phase — System microprocessor reads the following result bytes from the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	ST0							
Byte 1	ST1							
Byte 2	ST2							
Byte 3	Not Defined							
Byte 4	Not Defined							
Byte 5	Not Defined							
Byte 6	Not Defined							

Recalibrate Command

This command causes the disk drive read/write head to retract to the track 0 position and resets the floppy controller internal track counter, thus recalibrating the controller and drive.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	0	0	0	0	0	1	1	1
Byte 1	0	0	0	0	0	0	DS1	DS0

Execution Phase — The controller moves the drive head back to track 0. The controller interrupts when the execution phase ends.

Result Phase — This command has no result phase. A Sense Interrupt Status command must be issued immediately after the interrupt to properly terminate the Recalibrate command.

Seek Command

This command moves the read/write head of the disk drive to the specified track.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	0	0	0	0	1	1	1	1
Byte 1	0	0	0	0	0	HDS	DS1	DS0
Byte 2	NCN							

Execution Phase — The controller moves the drive head to the new cylinder. The controller interrupts when the execution phase ends.

Result Phase — This command has no result phase. A Sense Interrupt Status command must be issued immediately after the interrupt to properly terminate the Seek command.

Sense Interrupt Status Command

This command resets the floppy controller interrupt signal and identifies the cause of an interrupt.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	0	0	0	0	1	0	0	0

Execution Phase — The controller resets the interrupt signal and identifies the interrupt cause. No interrupt is generated.

Result Phase — System microprocessor reads the following result bytes from the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	ST0							
Byte 1	PCN							

Sense Drive Status Command

This command obtains the disk drive status, which is read by the system during the result phase of the command.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	0	0	0	0	0	1	0	0
Byte 1	0	0	0	0	0	HDS	DS1	DS0

Execution Phase — This command has no execution phase and goes directly from the command phase to the result phase. No interrupt is generated.

Result Phase — System microprocessor reads the following result bytes from the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	ST3							

Specify Command

This command sets up the timings to be used for controller interaction with the disk drives(s).

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	0	0	0	0	0	0	1	1
Byte 1	SRT				HUT			
Byte 2	HLT							ND

Execution Phase — This command has no execution phase. No interrupt is generated.

Result Phase — This command has no result phase.

Configure Command

The configure command is an 82077 command that enables certain 82077 unique features.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	0	0	0	1	0	0	1	1
Byte 1	0	0	0	0	0	0	0	0
Byte 2	0	EIS	EFIFO	POLL	FIFOTHR			
Byte 3	PRETRK							

Execution Phase — This command has no execution phase.

Result Phase — This command has no result phase.

Relative Seek Command

This command performs a seek operation, moving the head in the specified direction and number of tracks from the current track.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	1	DIR	0	0	1	1	1	1
Byte 1	0	0	0	0	0	HDS	DS1	DS0
Byte 2	RCN							

Execution Phase — The FDC moves the head to the new cylinder and interrupts at the end of the execution phase.

Result Phase — This command has no result phase. A Sense Interrupt Status command must be issued (after the interrupt occurs) to properly terminate the Relative Seek command.

Verify Command

This command is used to verify the data stored on a disk. This command acts exactly like a Read Data command, except that no data is transferred to the system.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	MT	MFM	SK	1	0	1	1	0
Byte 1	EC	0	0	0	0	HDS	DS1	DS0
Byte 2	C							
Byte 3	H							
Byte 4	R							
Byte 5	N							
Byte 6	EOT							
Byte 7	GPL							
Byte 8	DTL/SC							

The EOT value should be set to the final sector to be checked. If EOT is greater than the number of sectors on the disk, the command will stop due to an error and no useful CRC information will be obtained.

Execution Phase — Data is read from the diskette and the CRC is computed and checked against the previously stored value. No data is transferred to the system. The FDC interrupts when the execution phase ends.

Result Phase — System microprocessor reads the following result bytes from the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	ST0							
Byte 1	ST1							
Byte 2	ST2							
Byte 3	C							
Byte 4	H							
Byte 5	R							
Byte 6	N							

Version Command

This command is used to determine if the FDC is an 82077 device.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	0	0	0	1	0	0	0	0

Execution Phase — This command has no execution phase. No interrupt is generated.

Result Phase — System microprocessor reads the following result byte from the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	1	0	0	1	0	0	0	0

This result byte indicates that the FDC is an 82077 enhanced FDC.

Dumpreg Command

This command is used to support system run-time diagnostics and application software development and debug.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	0	0	0	0	1	1	1	0

Execution Phase — This command has no execution phase. No interrupt is generated.

Result Phase — System microprocessor reads the following result byte from the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	PCN – Drive 0							
Byte 1	PCN – Drive 1							
Byte 2	PCN – Drive 2							
Byte 3	PCN – Drive 3							
Byte 4	SRT				HUT			
Byte 5	HLT							ND
Byte 6	SC/EOT							
Byte 7	LOCK	0	D3	D2	D1	D0	GAP	WGATE
Byte 8	0	EIS	EFIFO	POLL	FIFOTHR			
Byte 9	PRETRK							

Perpendicular Mode Command

This command should be issued prior to executing Read/Write/Format commands that access a disk drive with perpendicular recording capability. With this command, the length of the GAP2 field and VCO enable timing can be altered to accommodate the unique requirements of these drives.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	0	0	0	1	0	0	1	0
Byte 1	OW	0	D3	D2	D1	D0	GAP	WGATE

Execution Phase — This command has no execution phase. No interrupt is generated.

Result Phase — This command has no result phase.

Lock Command

This command is used to protect the FIFO in long DMA latency systems from software packages which reset the controller but do not reconfigure the FIFO.

Command Phase — System microprocessor issues the following command and parameter bytes to the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	LOCK	0	0	1	0	1	0	0

Execution Phase — This command has no execution phase. No interrupt is generated.

Result Phase — System microprocessor reads the following result byte from the controller:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte 0	0	0	0	LOCK	0	0	0	0

Invalid Commands

If an invalid command is issued to the floppy controller, the controller enters a standby state and will return ST0 = 80H during the result phase.

Parameter Fields

The floppy controller parameter byte fields are defined in the following table:

Table 8-7 Parameter Fields

Symbol	Name	Description
C	Cylinder Number	The cylinder track number (0 through 255) of the medium.
D	Data	The data pattern to be written to a sector.
DIR	Direction	If 0, the head is stepped out. If 1, the head is stepped in.
DS0, DS1	Drive Select	The binary encoded diskette drive number
EC	Enable Count	Setting this bit to1 causes the Verify Command's DTL parameter to become the number of sectors per cylinder (SC).
EFIFO	Enable FIFO	Setting this bit to 0 causes the FIFO to be enabled. The FIFO is a 16-byte buffer used in the execution phase of read or write commands. A "1" puts the 82077 in the NCR765 compatible mode so that the FIFO is disabled. This bit defaults to "1" after a reset.
EIS	Enable Implied Seek	When set, a seek operation will be performed before executing any read or write command that requires the C parameter in the command phase. A "0" disables the implied seek.
EOT	End of Track	The final sector number of a cylinder.
FIFOTHR	FIFO Threshold	When the FIFO is enabled, this field sets the FIFO threshold level. A "0" selects 1 byte, "F" selects 16 bytes.
GAP	Gap	When using Perpendicular Mode, this parameter alters Gap2 length.
GPL	Gap Length	The length of GAP3 on the media (spacing between sectors excluding the VCO sync field).
H	Head Address	The head number (0 or 1) as specified in the ID field.
HDS	Head Select	The selected head number (0 or 1). H = HDS in all command bytes.
HLT	Head Load Time	The head load time for the floppy drive (2 to 254 ms in 2 ms increments).
HUT	Head Unload Time	The head unload time for the floppy drive after a read or write operation (16 ms to 240 ms in 16 ms increments).

Table 8-7 Parameter Fields (Continued)

Symbol	Name	Description
LOCK	Lock	Defines whether a Software Reset (setting the proper bit in the DSR or DOR registers) can reset the CONFIGURE command parameters EFIFO, FIFOTHR, and PRETRK to their default values.
MFM	MFM Mode	Used to select MFM mode. The NCR82077 will always operate in MFM mode regardless of this value.
MT	Multi-track	If MT = 1, then both sides of a cylinder can be accessed with a single command.
N	Number	The encoded number of data bytes written to a sector.
NCN	New Cylinder Number	The new cylinder number to be reached as result of a seek operation.
ND	Non-DMA Mode	If ND = 1, the NCR82077 will use the interrupt signal rather than DRQ for data transfers.
OW	Overwrite	If OW = 1, then the Perpendicular Mode command bits D3, D2, D1, and D0 can be overwritten.
PCN	Present Cylinder Number	The cylinder number at which a given drive is set.
POLL	Polling Disable	When set, the internal drive polling routine is disabled. When cleared, polling is enabled.
PRETRK	Write Precompensation Start Track	This field selects the track number at which to begin write precompensation. 00 selects track 0, FF selects track 255. At power-on reset, the 82077 defaults to Precompensation at track 0. All tracks equal to or greater than PRETRK will have precompensated WRDATA
R	Record	The sector number to be read or written.
RCN	Relative Cylinder Number	Determines how many tracks to step the head in or out from the current track number.
SC	Sector	The number of sectors per cylinder.
SK	Skip	If SK = 1, deleted data sectors will be skipped during reads. Only detected data sectors will be read during Read Deleted Data commands.
SRT	Step Rate Time	The stepping rate for the floppy drive during seek operations.

Table 8-7 Parameter Fields (Continued)

Symbol	Name	Description
ST0–ST3	Status Registers 0 through 3	The four floppy controller internal status registers which contain the result status following the execution of a command.
WGATE	Write Gate	Allows for pre-erase loads in perpendicular drives, by altering the timing of the Write Enable (WE) signal.

Register Description

The following sections describe the NCR82077 Floppy Disk Controller Core registers. Each register is detailed in terms of bit configuration, the active states of each bit, their definitions, their functions and their effects upon the internal hardware and external pins.

Status, Data, and Control Registers

The base address range is supplied by the CSb pin. This should be 3F0 Hex to 3F7 Hex for PC/AT and PS/2 designs.

A2	A1	A0		Register	
0	0	0	R	Status Register A	SRA
0	0	1	R	Status Register B	SRB
0	1	0	R/W	Digital Output Register	DOR
0	1	1	R/W	Tape Drive Register	TDR
1	0	0	R	Main Status Register	MSR
1	0	0	W	Data Rate Select Register	DSR
1	0	1	R/W	Data (FIFO)	FIFO
1	1	0		Reserved (NCR test mode enable)	
1	1	1	R	Digital Input Register	DIR
1	1	1	W	Configuration Control Register	CCR

NOTE: Within each register, the bit values reflect the state of the drive signals on the cable, independent of the state of the INVERTb pin.

Status Register A (PS/2 Mode)

This read-only register, not accessible in PC/AT mode, serves a monitoring function. It contains the state of the interrupt pin and of several disk interface pins.

D7	D6	D5	D4	D3	D2	D1	D0	
							●	Direction
						●		$\overline{\text{Write Protect}}$
					●			$\overline{\text{Index}}$
				●				Head 1 Select
			●					$\overline{\text{Track 0}}$
		●						Step
	●							$\overline{\text{Second Drive Installed}}$
●								Interrupt Pending

D0—Direction
This bit returns the state of the drive interface DIR signal.
(1 = head movement toward spindle for steps).

D1—$\overline{\text{Write Protect}}$
This bit returns the inverted state of the drive interface WP signal.
(0 = write protected).

D2—$\overline{\text{Index}}$
This bit returns the inverted state of the drive interface INDX signal
(0 = at index).

D3 Head 1 Select.
This bit returns the state of the drive interface HDSEL signal.
(0 = head 0).

D4—$\overline{\text{Track 0}}$
This bit returns the inverted state of the drive interface TRK0 signal.
(0 = at track 0).

D5—Step
This bit returns the state of the drive interface STEP signal.
(1 = Step active).

D6—$\overline{\text{2nd Drive Installed}}$
This bit returns the inverted state of the drive interface DRV2 signal.
(0 = 2nd drive installed).

D7—Interrupt Pending
This bit returns the state of the controller's INT signal.
(1 = interrupt pending).

Status Register A (PS/2 Model 30 Mode)

In the PS/2 Model 30 mode, five of the eight bits are inverted (bits 0 through 4) and two others have been changed.

D7	D6	D5	D4	D3	D2	D1	D0	
							●	$\overline{\text{Direction}}$
						●		Write Protect
					●			Index
				●				$\overline{\text{Head Select}}$
			●					Track 0
		●						Step
	●							DMA Request
●								Interrupt Pending

D—$\overline{\text{Direction}}$
This bit returns the inverted state of the drive interface DIR signal. (1 = head movement toward spindle for steps).

D1—Write Protect
This bit returns the state of the drive interface WP signal. (0 = write protected).

D2—Index
This bit returns the state of the drive interface INDX signal (0 = at index).

D3—$\overline{\text{Head Select}}$
This bit returns the state of the drive interface HDSEL signal. (0 = head 0).

D4—Track 0
This bit returns the inverted state of the drive interface TRK0 signal. (1 = at track 0).

D5—Step
This bit returns the state of the drive interface STEP signal. (1 = Step active).

D6—DMA Request
This bit returns the status of the DRQ pin.

D7—Interrupt Pending
This bit returns the state of the controller's INT signal. (1 = interrupt pending).

Status Register B (PS/2 Mode)

This read-only register, not accessible in PC/AT mode, serves a monitoring function. It reflects the state of several disk interface pins.

D7	D6	D5	D4	D3	D2	D1	D0	
							●	Motor Enable 0
						●		Motor Enable 1
					●			Write Enable
				●				Read Data Toggle
			●					Write Data Toggle
		●						Drive Select Bit 0
●	●							Always read as 1s

D0—Motor Enable 0
This bit returns the state of the Digital Output Register Motor Enable 0 bit.

D1—Motor Enable 1
This bit returns the state of the Digital Output Register Motor Enable 1 bit.

D2—Write Enable
This bit returns the state of the drive interface WE signal.

D3—Read Data
This bit returns the state of a flip-flop which produces a wider more reliably read pulse produced from the drive interface RDDATA signal.

D4—Write Data
This bit returns the state of a flip-flop which produces a wider more reliably read pulse produced from the drive interface WRDATA signal.

D5—Drive Select Bit 0
This bit returns the state of the drive interface DS0 signal.

Status Register B (PS/2 Model 30 Mode)

This register in PS/2 Model 30 mode returns inverted values of drive select bits in the DOR register, as well as other changes.

D7	D6	D5	D4	D3	D2	D1	D0	
							●	$\overline{\text{Drive Select 2}}$
						●		$\overline{\text{Drive Select 3}}$
					●			Write Enable
				●				Read Data
			●					Write Data
		●						$\overline{\text{Drive Select 0}}$
	●							$\overline{\text{Drive Select 1}}$
●								$\overline{\text{Second Drive Installed}}$

D0—$\overline{\text{Drive Select 2}}$
This bit returns the inverted state of the drive interface DS2 signal.

D1—$\overline{\text{Drive Select 3}}$
This bit returns the inverted state of the drive interface DS3 signal.

D2—Write Enable
This bit returns the state of the drive interface WE signal.

D3—Read Data
This bit returns the state of the drive interface RDDATA signal.

D4—Write Data
This bit returns the state of the drive interface WRDATA signal.

D5—$\overline{\text{Drive Select 0}}$
This bit returns the inverted state of the drive interface DS0 signal.

D6—$\overline{\text{Drive Select 1}}$
This bit returns the inverted state of the drive interface DS1 signal.

D7—$\overline{\text{2nd Drive Installed}}$
This bit returns the inverted state of the drive interface DRV2 signal.
(0 = 2nd Drive installed)

Digital Output Register

This is both a read and a write register. It contains the motor enable and drive select bits, which by standard programming practice are set at the same time to control their respective pins. A DMAGate bit and a reset bit are also included with this register.

D7	D6	D5	D4	D3	D2	D1	D0	
							●	Drive Select Bit 0
						●		Drive Select Bit 1
					●			$\overline{\text{Controller Reset}}$
				●				$\overline{\text{DMAGate}}$
			●					Motor Enable 0
		●						Motor Enable 1
	●							Motor Enable 2
●								Motor Enable 3

D0–D1—Drive Select Bits 0, 1
These bits control which Drive Select signal will be active when the respective motor enable bit is set.

D1	D0	Drive Selected
0	0	Drive 0
0	1	Drive 1
1	0	Drive 2
1	1	Drive 3

D2—$\overline{\text{Controller Reset}}$
If 0, this bit causes a reset of the floppy disk controller. The reset remains active until this bit is written as a 1.

D3—$\overline{\text{DMAGate}}$
This bit is enabled only in PC/AT and PS/2 Model 30 modes. A high setting is used to system enable INT, DRQ, TC, and DACKb.

D4—Motor Enable 0
If this bit is set to a 1 and Drive Select bits (1, 0) = (00), drive interface signals DS0 and ME0 are active.

D5—Motor Enable 1
If this bit is set to a 1 and Drive Select bits (1, 0) = (01), drive interface signals DS1 and ME1 are active.

D6—Motor Enable 2
If this bit is set to a 1 and Drive Select bits (1, 0) = (10), drive interface signals DS2 and ME2 are active.

D7—Motor Enable 3
If this bit is set to a 1 and Drive Select bits (1, 0) = (11), drive interface signals DS3 and ME3 are active.

Tape Drive Register

During initialization, this read and write register permits user-specified tape support assignment to a specific drive. Tape support is then automatically invoked by any subsequent references to that drive number. A hardware reset is required to clear this register.

D7	D6	D5	D4	D3	D2	D1	D0	
						●	●	Tape Select Bits 1, 0
●	●	●	●	●	●			Tristated

D1–D0—Tape Select Bits 1, 0

These bits determine which drive is to be assigned tape support during initialization, and can only be cleared by a hardware reset. The possible tape drive selections are as follows:

Tape Select Bit 1	Tape Select Bit 0	Drive Selected
0	0	Reserved (for floppy boot drive)
0	1	Drive 1
1	0	Drive 2
1	1	Drive 3

Main Status Register

This read-only register indicates the device status for all command inputs and result outputs.

D7	D6	D5	D4	D3	D2	D1	D0	
							●	Drive 0 Busy
						●		Drive 1 Busy
					●			Drive 2 Busy
				●				Drive 3 Busy
			●					Command Busy
		●						Non-DMA Mode
	●							Data Input/Output (DIO)
●								Request for Master (RQM)

D0—Drive 0 Busy
When this bit is 1, floppy drive 0 is in the seek mode.

D1—Drive 1 Busy
When this bit is 1, floppy drive 1 is in the seek mode.

D2—Drive 2 Busy
When this bit is 1, floppy drive 2 is in the seek mode.

D3—Drive 3 Busy
When this bit is 1, floppy drive 3 is in the seek mode.

D4—Command Busy
When this bit is 1, the floppy controller has a read or write command in progress.

D5—Non-DMA Mode
This bit is only set during the execution phase of a command when the floppy controller is being operated in the non-DMA mode. The transition from 1 to 0 indicates the completion of a command in non-DMA mode.

D6—Data Input/Output (DIO)
This bit indicates the permitted direction of data transfer between the system's microprocessor and the floppy controller data register. When this bit is a 0, the direction is from the system microprocessor to the floppy controller. When this bit is a 1, the direction is from the floppy controller to the system microprocessor

D7—Request for Master (RQM)
When this bit is a 1, the floppy controller data register is ready for a transfer to/from the system microprocessor as indicated by the Data Input/Output bit.

Data Rate Select Register

This write-only register, with the exception of an absent internal/external PLL select bit, replicates that used in the 82077 and thus lends itself to compatibility. Drive control signal timings are changed by modifying the data rates.

D7	D6	D5	D4	D3	D2	D1	D0	
						●	●	Data Rate Select bits 1, 0
			●	●	●			PRECOMP 2, 1, 0
		●						Always Read as 0
	●							Power Down
●								Controller Reset

D1–D0—Data Rate Select Bits 1, 0
These bits determine the data rate to be used as follows:

D1	D0	MFM
1	1	1 Mbps
0	0	500 Kbps
0	1	300 Kbps
1	0	250 Kbps

D4–D2—PRECOMP Bits 2, 1, 0
These bits are used to adjust the WRDATA output to the disk to compensate for bit shifting, and should normally be written as 0 to select the default write precompensation delay of 125 ns.

Table 8-8 Selectable Precompensation Delays

PC2	PC1	PC0	Precompensation Delay
0	0	0	Default
0	0	1	41.67 ns
0	1	0	83.34 ns
0	1	1	125.00 ns
1	0	0	166.67 ns
1	0	1	208.00 ns
1	1	0	250 ns
1	1	1	0.00 ns – Disabled

Table 8-9 Default Precompensation Delays

Default Data Rate	Precompensation Delay
1 Mbps	41.67 ns
500 Kbps	125 ns
300 Kbps	125 ns
200 Kbps	125 ns

D6—Power Down
This bit is used to activate the internal clocks and also shut off the oscillator.

D7—Controller Reset
This bit exhibits the same behavior pattern as DOR RESET, with the exception of being self-clearing.

FIFO (Data) Register

The FIFO (Data) Register buffers the system by 16 bytes to provide a wider latitude of DMA latency before over/underrun errors occur.

Digital Input Register (PC/AT Mode)

In all modes, this is a read-only register. For this mode, other than bit 7 being driven, all other bits remain tristated.

D7	D6	D5	D4	D3	D2	D1	D0	
	●	●	●	●	●	●	●	Tristated
●								Diskette Change

D7—Diskette Change
This bit returns the state the Diskette Change signal. (1 = diskette changed).

Digital Input Register (PS/2 Mode)

In this mode, the valued of the current data rate is added to the information provided in the PC/AT mode.

D7	D6	D5	D4	D3	D2	D1	D0	
							●	$\overline{\text{High Density Select}}$
					●	●		Data Rate Select Bits 1, 0
	●	●	●	●				Always Read as 1s
●								Diskette Change

D0—$\overline{\text{High Density Select}}$
This bit returns the inverted state of the drive interface DENSEL signal. For example, 0 = high density, that is, 500 Kbps or 1 Mbps.

D2–D1—Data Rate Select Bits 1, 0
These bits together return the currently selected data rate.

D7—Diskette Change
This bit returns the state of the disk interface DSKCHG signal (1 = diskette changed).

Digital Input Register (PS/2 Model 30 Mode)

In this mode, the significant changes are as follows: inverting bit 7, moving the bit positions of the data rate selection bits, and adding DOR and CCR information..

D7	D6	D5	D4	D3	D2	D1	D0	
						●	●	Data Rate Select Bits 1, 0
					●			NOPREC
				●				$\overline{\text{DMAGATE}}$
	●	●	●					Reserved (Read as 0s)
●								$\overline{\text{Diskette Change}}$

D1–D0—Data Rate Selection Bits 1, 0
These bits together return the currently selected data rate, reflecting their combined settings MFM table rate for the Data Rate Select register.

D2—NOPREC
This bit returns the NOPREC bit setting value in the Configuration Control register.

D3—$\overline{\text{DMAGATE}}$
This bit returns the $\overline{\text{DMAGATE}}$ bit setting value in the Digital Output register.

D7—$\overline{\text{Diskette Change}}$
This bit returns the inverted state of the disk interface DSKCHG signal (1 = diskette changed).

Configuration Control Register (PC/AT and PS/2 Modes)

This write-only register is the Data Rate Select register in the PC/AT mode, with the exception of bits 2 through 7 being always read as 0. It is used to select the data rate.

D7	D6	D5	D4	D3	D2	D1	D0	
						●	●	Data Rate Select Bits 1, 0
●	●	●	●	●	●			Always Read as 0s

D1–D0—Data Rate Select Bits 1, 0
These bits together return the currently selected data rate, reflecting their combined settings MFM table rate for the Data Rate Select register.

Configuration Control Register (PS/2 Model 30 Mode)

This is identical to PC/AT and PS/2 modes, with the exception of the NOPREC bit 2.

D7	D6	D5	D4	D3	D2	D1	D0	
						●	●	Data Rate Select Bits 1, 0
					●			NOPREC
●	●	●	●	●				Always Read as 0s

D1–D0—Data Rate Select Bits 1, 0
These bits together return the currently selected data rate, reflecting their combined settings MFM table rate for the Data Rate Select register.

D2—NOPREC
This bit in this particular register situation has no function, being reset to 0 only on a hardware RESET.

Command Structure

Command Phases

Each controller operation is performed in a three-phase sequence.

- Command Phase
- Execution Phase
- Result Phase

Command Phase

The command and parameter bytes are written to the floppy Controller Data register. The microprocessor must poll the Main Status register before each write to the FDC Data register to determine when the controller is prepared to accept a transfer.

Execution Phase

The command given to the controller is performed. All data transfers occur in the execution phase. The FDC will generate an interrupt when it completes the execution phase of all commands except the following:

- Configure
- Dumpreg
- Perpendicular Mode
- Sense Drive Status
- Sense Interrupt Status
- Specify
- Version

Result Phase

The results of the given command are read from the FDC Data register by the system microprocessor. The microprocessor must poll the Main Status register to determine when the floppy controller is prepared to transfer each byte of the result. The system microprocessor must read all result bytes from the controller before attempting to send a new command. Some commands do not have a result phase and require the issue of a Sense Interrupt Status command to properly terminate.

Result Status Registers

Status Register 0

The content of this floppy controller internal register is returned to the system microprocessor during the result phase of most commands.

D7	D6	D5	D4	D3	D2	D1	D0	
						●	●	Drive Select Bits 1, 0 (DS1, DS0)
					●			Head Address (H)
				●				Always Read as 0s
			●					Equipment Check (EC)
		●						Seek End (SE)
●	●							Interrupt Code (IC)

D1–D0—Drive Select Bits 1, 0
These bits indicate which drive interface–Drive Select signal is active, in the following manner:

D1=0, D0=0 — Drive 0 selected

D1=0, D0=1 — Drive 1 selected

D1=1, D0=0 — Drive 2 selected

D1=1, D0=1 — Drive 3 selected

D2—Head Address
This bit indicates which drive head is selected.

D4—Equipment Check
Set to a 1 if the track 0 signal fails to occur within 80 step pulses during a recalibrate command, or the relative seek command steps the 82077 out past track 0.

D5—Seek End
Set to a 1 when the controller completes a seek command (or a read or write command with implied seek).

D7–D6—Interrupt Code
D7=0, D6=0 — Successful command completion.

D7=0, D6=1 — Command was started, but terminated abnormally.

D7=1, D6=0 — Invalid command (never started)

D7=1, D6=1 — Command was started, but terminated abnormally due to polling.

Status Register 1

The content of this register is returned to the microprocessor during the Result phase of most commands.

D7	D6	D5	D4	D3	D2	D1	D0	
							●	Missing Address Mark (MA)
						●		Not Writable (NW)
					●			No Data (ND)
				●				Always Read as 0
			●					Over Run/Under Run (OR)
		●						Data Error (DE)
	●							Always Read as 0
●								End of Cylinder (EN)

D0—Missing Address Mark
This bit is set if the floppy controller cannot detect the Address ID Mark after encountering the media's Index Mark twice.

This bit is also set if the floppy controller cannot detect the Data Address Mark or Deleted Data Address Mark on the designated track. In this case, D0 of ST2 will also be set.

D1—Not Writable
This bit is set if the floppy controller detects write protection during a Write Data, Write Deleted Data, or Format command.

D2—No Data
This bit is set under the following conditions: if the controller cannot find the sector specified in the ID register during a Read Data, Read Deleted Data, Write Data, or Write Deleted Data command; if a Read ID command cannot read the ID field without error; or if the Read Track command cannot locate the correct sector sequence.

D4—Over Run/Under Run
This bit is set if the floppy controller is not serviced in time by the system microprocessor or DMA controller during data transfers.

D5—Data Error
This bit is set if the floppy controller detects a CRC error reading either the ID field or data field on the media.

D7—End of Cylinder
This bit is set if the controller tries to access a sector further than the final sector of a track. Also set if Terminal Count is not reached after a Read or Write Data command.

Status Register 2

The content of this register is returned to the system microprocessor during the Result phase of most commands.

D7	D6	D5	D4	D3	D2	D1	D0	
							●	Missing Address Mark in Data Field (MD)
						●		Bad Cylinder (BC)
					●			Always Read as 0
				●				Always Read as 0
			●					Wrong Cylinder (WC)
		●						Data Error in Data Field (DD)
	●							Control Mark (CM)
●								Always Read as 0

D0—Missing Address Mark in Data Field
This bit is set if the floppy controller cannot find a Data Address Mark or Deleted Data Address Mark when data is read from the medium.

D1—Bad Cylinder
This bit is set when the cylinder number read from the medium differs from that in the floppy controller ID register, and the medium cylinder number = FFH (bad track with hard error in IBM soft-sectored format). This bit is related to D2 in ST1.

D4—Wrong Cylinder
This bit is set if the cylinder number read from the medium is different than that in the controller's ID register.

D5—Data Error in Data Field
This bit is set if the floppy controller detects a CRC error in the data field.

D6—Control Mark
This bit is set if the floppy controller encounters a sector with a Deleted Data Address Mark during a Read Data command, or the opposite (a Data Address Mark during a Read Deleted Data command).

Status Register 3

The content of this register is returned to the system microprocessor during the result phase of a Sense Drive Status command.

D7	D6	D5	D4	D3	D2	D1	D0	
						●	●	Drive Select Bits 1, 0 (DS1, DS0)
					●			Head Address (HD)
				●				Always Read as 1
			●					Track 0 (T0)
		●						Always Read as 1
	●							Write Protected (WP)
●								Always Read as 0

D1–D0—Drive Select Bits 1, 0
These bits indicate which drive interface DS1, DS0 signal is active in the following manner:

D1=0, D0=0 Drive 0 selected
D1=0, D0=1 Drive 1 selected
D1=1, D0=0 Drive 2 selected
D1=1, D0=1 Drive 3 selected

D2—Head Address
This bit indicates the state of the drive interface HDSEL signal.

D4—Track 0
This bit indicates the state of the TRK0 signal from the floppy drive.

D6—Write Protected
This bit indicates the state of the WP signal from the floppy drive.

Timing Information

All timings were measured at the NCR82077 macrocell boundaries. Timings were measured using the 70° C., 4.5 V, worst-case process. Actual timing will vary depending on the output loading and layout. All timings should be verified by the macrocell user. Additional delays will be added to the output signals depending on the selected drivers and output pads.

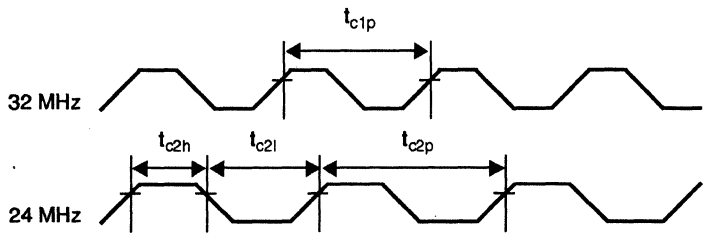


Figure 8-5 Clock Inputs

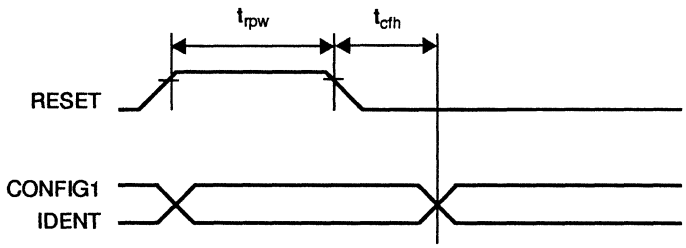


Figure 8-6 Reset Timing

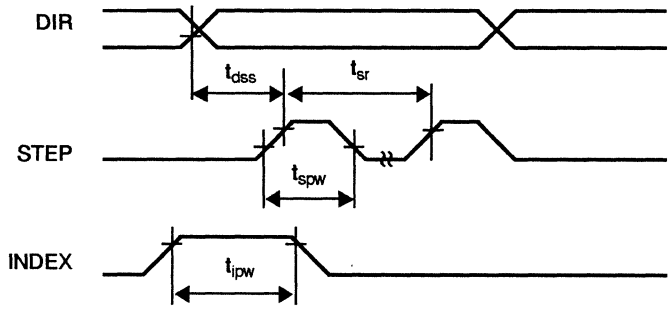


Figure 8-7 Drive Interface Timing

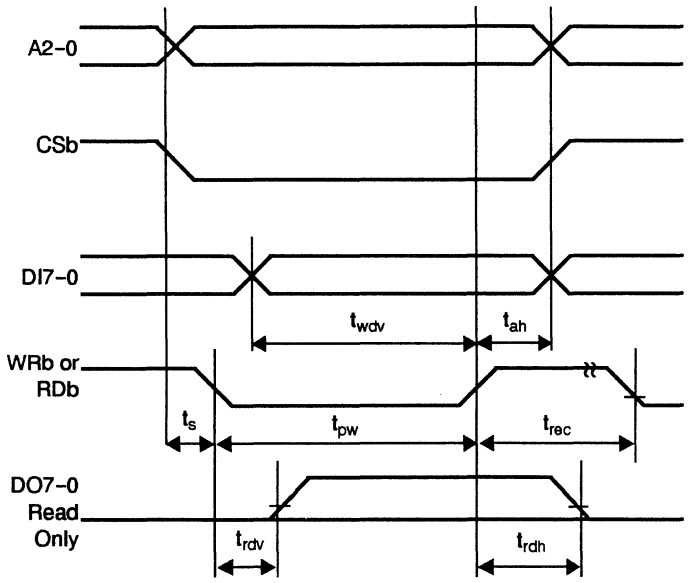


Figure 8-8 Register Operation

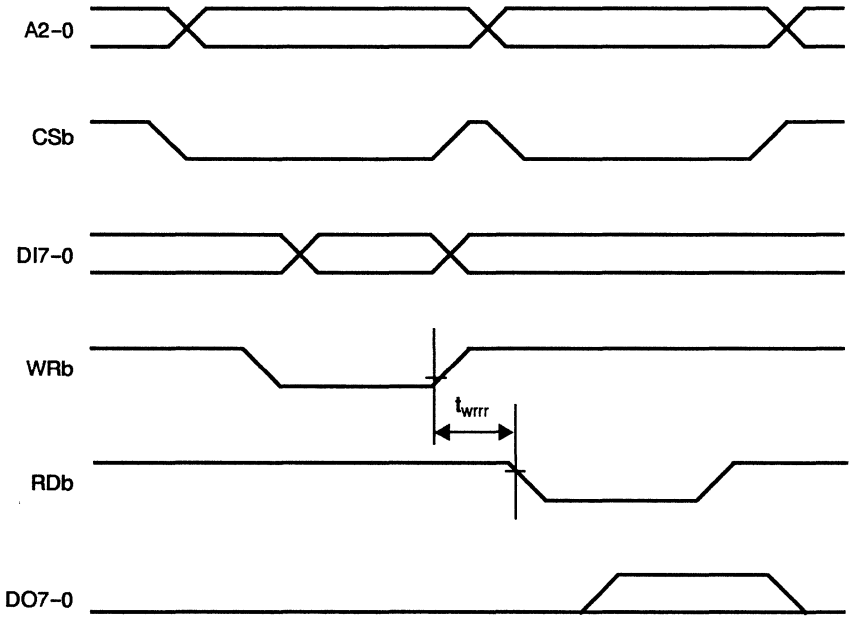


Figure 8-9 Register Write Register Read Operation

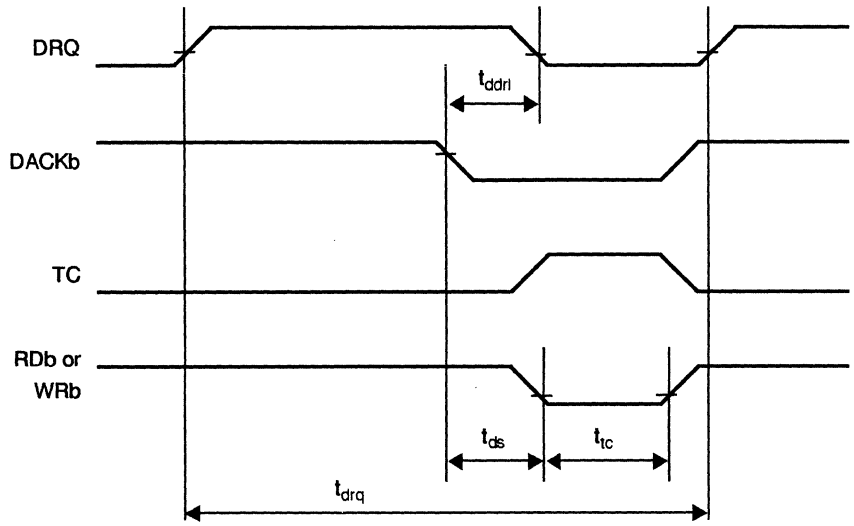
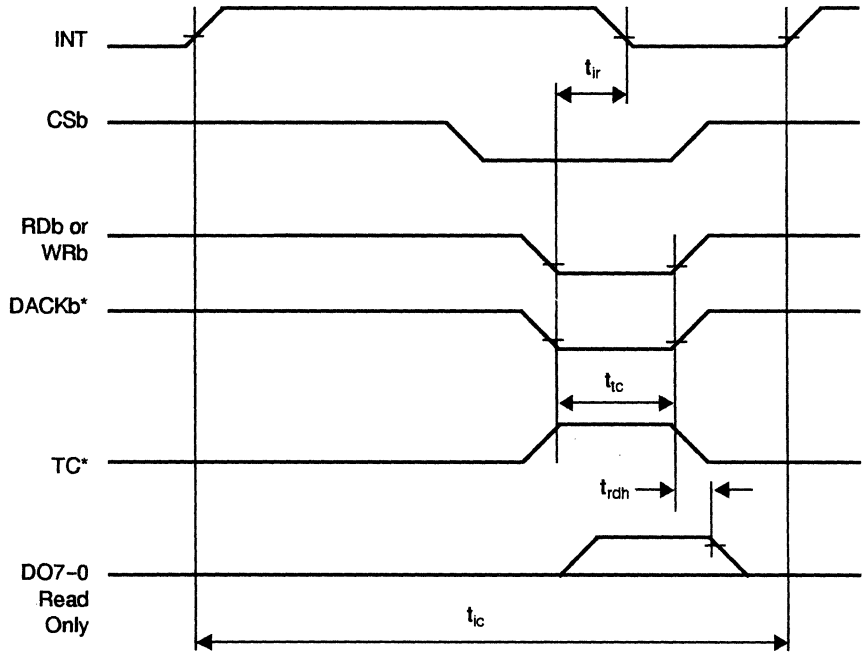


Figure 8-10 DMA Cycle - FIFO Disabled



* Both DACKb and TC must be asserted to terminate the data transfer for both DMA and Non-DMA data transfers

Figure 8-11 Non-DMA Cycle - FIFO Disabled

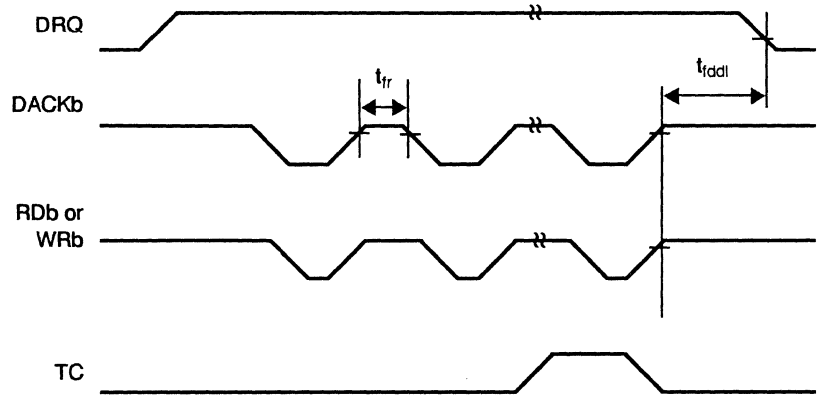
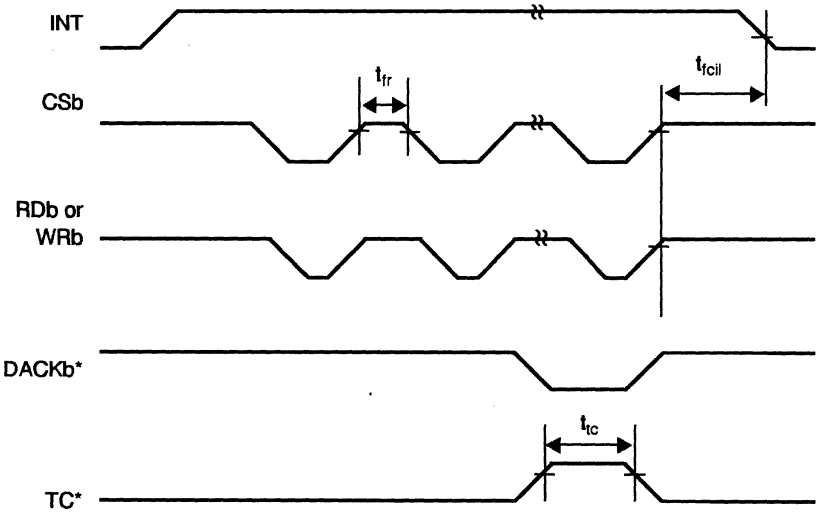


Figure 8-12 DMA Cycle - FIFO Enabled



* Both DACKb and TC must be asserted to terminate the data transfer for both DMA and Non-DMA data transfers

Figure 8-13 Non-DMA Cycle - FIFO Enabled

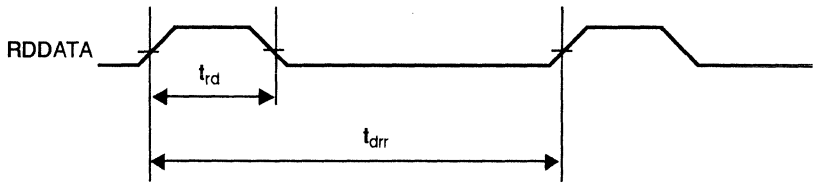


Figure 8-14 Read Data

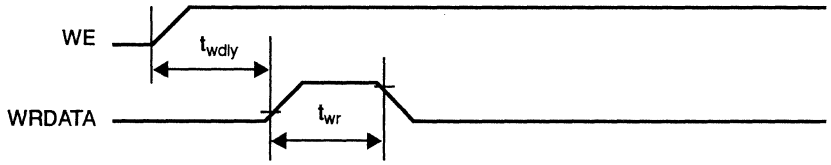


Figure 8-15 Write Data

Temperature and Voltage Range = 0° C, V_{CC} = 5 V ± 10%, External Load 1 pF

Symbol	Parameter	Min	Max	Unit	Comments/ Conditions
t_{c1p}	32 MHz Clock Period	31.25	31.25	ns	
t_{c2h}	24 MHz Clock High Time	16.67	25	ns	
t_{c2l}	24 MHz Clock Low Time	16.67	25	ns	
t_{c2p}	24 MHz Clock Period	41.67	41.67	ns	
t_{ci}	Internal Clock Period	62.5	250	ns	See Note 1
t_{rpw}	Reset Pulse Width	125		ns	
t_{cfh}	RESET Low to CONFIG1, IDENT Change Hold Time	5		ns	
t_{dss}	DIR Change to STEP Setup Time	1.5		µs	
t_{spw}	STEP Pulse Width	7	8	µs	
t_{sr}	STEP Rate	1	15	µs	See Note 8
t_{ipw}	INDEX Pulse Width	4		t_{ci}	
t_{pw}	RDb or WRb Pulse Width	80		ns	
t_s	Address, CSb Setup to WRb or RDb	0		ns	
t_{wdv}	Data Valid Before Rising Edge of WRb	20		ns	
t_{rec}	Delay between data accesses	4		t_{ci}	See Note 2
t_{ah}	Address, CSb Hold Time	0		ns	
t_{rdv}	Data Valid from RDb Low	4	0	ns	
t_{rdh}	Data Hold from RDb Rising	0		ns	
t_{wrr}	Delay from WRb↑to RDb↓	60		ns	
t_{drq}	DRQ Cycle Period	6.5		µs	See Note 3
t_{ddrl}	DACKb Low to DRQ Low – FIFO Disabled		20	ns	

Temperature and Voltage Range = 0° C, V_{CC} = 5 V ± 10%, External Load 1 pF

Symbol	Parameter	Min	Max	Unit	Comments/ Conditions
t_{ds}	DACKb Low to RDb or WRb low	0		ns	
t_{tc}	TC Pulse Width	80		ns	See Note 4
t_{ic}	INT Cycle Period	6.5		µs	See Note 3
t_{ir}	RDb or WRb low to INT Low – FIFO disabled		40	ns	
t_{fr}	FIFO Read or Write Recovery Time	80		ns	
t_{fddl}	DACKb to DRQ Low – FIFO Enabled		75	ns	See Note 5
t_{fcil}	RDb or WRb High to INT Low – FIFO Enabled		75	ns	See Note 5
t_{rd}	RDDATA Pulse Width	50		ns	
t_{drr}	RDDATA Data Rate	1.0	4.0	µs	See Note 6
t_{wdly}	WE to WRDATA Setup Time	320		ns	
t_{wr}	WRDATA Pulse Width	125	500	ns	See Note 7

NOTE 1: Internal clock period is a function of the selected data rate.

Data Rate	Frequency	Period
1 Mbs	16 MHz	62.5 ns
500 Kbs	8 MHz	125 ns
300 Kbs	4.8 MHz	208 ns
250 Kbs	4 MHz	250 ns

NOTE 2: t_{REC} is defined as the time between successive read or writes. t_{WRREC} is defined as the time between a data register write and main status register read.

NOTE 3: DRQ and INT cycle period is dependent on the selected data rate and the FIFO enable. The following values are valid if the FIFO is disabled:

- 1 Mbs – 6.5µs
- 500 Kbs – 13µs
- 300 Kbs – 21.6µs
- 250 Kbs – 26µs

NOTE 4: TC pulse width is determined by TC and DACKb both being active. The combined width must be at least two 24 MHz clock periods.

NOTE 5: DRQ or INT will remain asserted until the FIFO is filled or emptied. The DACKb (DMA enabled) or WRb, RDb falling (DMA disabled) that fills or empties the FIFO will remove DRQ or INT until the FIFO reaches the configured threshold. DRQ or INT will be reasserted.

NOTE 6: RDDATA data rate is determined by the floppy tape drive or tape drive. The values are:

- 1 Mbs – 1.0µs minimum
- 500 Kbs – 2.0µs minimum
- 300 Kbs – 3.3µs minimum
- 250 Kbs – 4.0µs minimum

NOTE 7: WRDATA pulse width is based on the selected data rate. The values are 2 X t_{CI}:

- 1 Mbs - 2 X 62.5 = 125 ns
- 500 Kbs - 2 X 125 = 250 ns
- 300 Kbs - 2 X 208 = 416 ns
- 250 Kbs - 2 X 250 = 500 ns

NOTE 8: Step Rate time is selected by a SPECIFY command. Failure to issue a specify command before issuing a Recalibrate or Seek command or implied seek will cause unpredictable results.

Application Notes

The following application note describes some of the differences between the Intel 82077AA standard part and the NCR82077 Floppy Disk Controller Core (FDCC). These operations were verified by actual system testing.

The main operational difference is that the FDCC does not pass data through the FIFO except in the execution phase of the command. Software that does not sample the Main Status register before writing a command byte or reading a result byte must allow sufficient time for one of the internal state machines to process the write or read. This time is four internal clock periods. The internal clock period is a function of the selected data rate. The internal clock period for 500 Kbs data rate is 125 ns. The user can write or read the FIFO/data register then read the Main Status register within 60 ns. The recommended user operation is to write or read the FIFO/data register, read the Main Status register until the RQM and DIO bits indicate the FDCC is ready for another write or read of the FIFO/data register during command or result phases.

The data sheet for the 82077AA implies that the device supports overlapped seeks. The floppy disk controller core of the 82077AA supports overlapped seek as does the core of the NCR82077. The core uses its internal drive selects to select a drive and issues a step pulse to each drive selected. If four drives are seeking at the same time, the core issues a burst of four step pulses, one for each drive. However, the actual drive selection is done using the DOR register. The DOR register must be written to select a drive and turn the motor on for the drive. It would be difficult to write the register and select different drives between the step pulses issued by the core. There would be no way to synchronize the drive selected with the proper step pulse. The user is cautioned to select and seek to only one drive at a time.

Detailed information on programming the counter modes and sample timing diagrams can be found in any standard 82077 data sheet.

Additional information on implementation of soft macrocells in a chip design can be found in the NCR application note for soft macrocells.

A full logic schematic of the 82077 design can be provided for modification in cases where the designer needs only a portion of the full macrocell. NCR assumes no responsibility for the functionality of a soft macrocell which has been altered in this manner. For more information, please contact your NCR Field Applications Engineer.

Compatibility Verification Procedures

The design and verification of a high-level ASF is directed towards functional compatibility with the existing standard part. The ASF is designed and verified to match data sheet functionality (when possible), then additional steps are taken to match undocumented functionality.

Creation of a Compatibility Model

A compatibility model of the new ASF is generated from published sources of information on the standard part including functional specifications, data sheets, application notes, review sheets and errata sheets. Exceptions to standard part behavior or function will be identified.

Utilization of Industry Consultants

NCR utilizes the advice of industry consultants who are highly knowledgeable about the operation of the standard part. This additional data includes features and functions which would be apparent to someone who is very familiar with the functionality of the part, but would not necessarily be covered in available documentation.

Use of Hardware Modeling Techniques and Extraction of System Patterns for Design Verification

Compatibility test vectors for the part are developed from two sources. First, a device model is created on a hardware modeling system (HML) to create test vectors based on the actual operation of the industry standard part. A second set of vectors is generated by extracting functional and timing information from the compatibility documentation and creating patterns based on this information.

In-system Test and Firmware Verification

To ensure that the ASF is truly representative of the standard part's functionality, a system test is executed on new sample parts. The test includes the following system parameters to detect any incompatibilities or inconsistencies:

- Selection of controllers and systems
- Special test programs
- System diagnostics
- Selection of operating systems
- Selection of application programs

Final Test Plan

Throughout the compatibility verification process, final test procedures for the ASF are emphasized. The NCR compatibility procedures provide assurance that the ASF will function as the standard part does with documented exceptions where warranted.

Kit Parts

Provision of kit parts that are plug-compatible with the existing standard part is a very important aspect of in-system verification. The availability of kit parts allows the customer to verify ASF operation and compatibility with the existing system design and software. Kit parts also provide a convenient way to bread-board "System Solutions in Silicon."

Application Notes
Contents

Application Notes

Frequently Asked Questions

Will SunSoft support the chipset?

The functions (SCSI, Ethernet, serial communication controllers, floppy disk controller) implemented in the chipset will be backward compatible with the devices which currently exist. These functions are all either the same ones that were used in earlier Sparcstations or a superset of these devices. New drivers are included in the Solaris 2.1 release from SunSoft which take advantage of both enhancements to the devices and new functions in the chipset (such as the parallel port).

How far is the NCR89C105 from SS-10 compatibility for the uniprocessor version?

The NCR89C105 contains a subset of the logic necessary to implement an SS-10 multiprocessing system. It contains one of the required four sets of timers and one interrupt controller. For a uniporocessor SS-10 implementation, the C105 includes the hooks necessary to allow it to work.

How compatible is the SCSI device in the NCR89C100 with the SCSI device in the SS-10?

The DMA2 bus mastering I/O functions in the SS-10 are identical to the functions integrated in the NCR89C100. Specifically, the SCSI in the SS-10 is the same core included in the C100 master device. It is capable of operating at 10 MB/s synchronous transfer and 7 MB/s asynchronous transfer.

When will the SCSI drivers be able to take advantage of the 10 MB/s data rate available in the NCR89C100?

The SunSoft release of Solaris 2.1, which is now available, takes advantage of the fast SCSI capability included in the C100.

How fast can the SCSI in the NCR89C100 execute asynchronous ata transfers?

The SCSI core in the C100 device is capable of transferring data at 7 MB/s asynchronously and 10 MB/s synchronously.

Will these devices be compatible with a 64-bit SBus Version B.0 system?

These devices should work in a system which is capable of performing 64-bit transfers in accordance with SBus Version B.0 due to the fact that version B.0 is to be fully backward compatible. They have not been tested

in any B.0 systems, however. It should be noted that while these devices should work in a 64-bit SBus system, they will not be able to take advantage of the extended (64-bit) transfers.

Is it necessary to provide SCSI termination or is it provided on the NCR89C100?

There is no SCSI termination on the C100 device. It is recommended that active termination be used for any implementation of the fast SCSI and that active termination also be used for the slower speeds to provide better margins on the timing.

How does the POR circuitry work on the NCR89C105? Is the entire POR circuit included on the chip or is it necessary to include additional external circuitry?

There is a pin on the NCR89C105 which looks for a signal from a power monitor chip to determine when it needs to reset the SBus. There is no power sensing circuitry on the NCR89C105 except for the digital signal that comes from the external POR circuitry.

What range of EPROM addressing is the NCR89C105 capable of?

The C105 can access up to 16 MBytes of memory.

Will Verilog models for both devices be available?

There are no plans to include the Verilog models as a part of the chipset.

Will the NCR89C100 work with an LXT901 Universal Ethernet Interface Adapter?

The NCR89C100 is capable of working with any of the Ethernet adapters supported by the industry standard 7990 Ethernet controller. At this time, all system testing has been done with the T7213 Dual Interface Station Chip from AT&T. A special AUI/TP select pin is designed in that directly interfaces to the T7213 to determine which output mode to use.

Will the 16-byte FIFO for the floppy controller in the NCR89C105 allow floppy accesses at the 1Mbit/s data rate?

Even with the addition of the 16-byte FIFO for the floppy, it is believed that there is not enough buffering to allow 1Mbit/s floppy accesses. Possibly, with a highly tuned driver, the system could respond fast enough to prevent underflow/overflow of the FIFO. The 82077 itself is capable of reading the data at the 1Mbit/s data rate.

How does extending the memory error timeout from 25.6 us to 102.4 us improve the performance of the Ethernet controller?

The memory error timeout has been extended because of the way that the cache is drained in the DMA2 portion of the NCR89C100. The cache is organized as two lines of 8 words each, and has a number of different conditions which can initiate a drain. When the cache starts to drain to clear a line for the transfer, the Ethernet controller can begin to start trying to transfer data into the cache. The cache will not be ready to accept the data until it has drained the line where the data will be stored. In the

worst case situation, this will take longer than the 25.6 us which the 7990 is set to wait, causing a memory timeout error and requiring that the Ethernet controller be reconfigured before it can be used again.

By increasing the memory timeout error time to 102.4 us, the worst case cache drains are given time to complete and the data is then successfully moved into the cache from the Ethernet controller. Because of the rapid internal transfers from the Ethernet FIFO to the DMA2 cache (effectively a second level of buffering) and the depth of both buffers, the data will successfully be transferred before an overflow of the Ethernet FIFO can occur. The increase of the timeout error eliminates the unnecessary reporting of an error which the system is designed to prevent. This, in turn, prevents the need to reconfigure the Ethernet core each time the timeout shows up.

Can the chipset handle the 12.5 MHz SBus operation in accordance with IEEE P1496?

At this time, no testing has been done below the original low SBus operating range of 16.67 MHz. It is believed that the chipset itself should function correctly at 12.5 MHz, but that the lower speeds would introduce SBus latency problems in a system.

Can the Ethernet core support chained transfers?

The Ethernet core in the NCR89C100 is compatible with the industry standard 7990 Ethernet controller. This device supports chained buffer transfers.

Can the PS/2 keyboard be supported with the 8530 interface?

The only way to support a PS/2 keyboard with this chipset is to use the generic 8 bit expansion slot off of the NCR89C105 slave device to interface to the keyboard controller. It is recommended that a Sun-compatible keyboard be used, as the hooks are already in place and there is a dedicated keyboard channel in the chipset.

How many crystals are needed to support all chipset features? What are their operating frequencies?

The crystals needed are as follows:

NCR89C100
- NCR89C100
- SCSI crystal—40.0 Mhz
- NCR89C105

NCR89C105
- 8530 SCC crystal—19.66 Mhz
- 82077 crystal—24.0 Mhz
- 82077 crystal—32.0 Mhz
- Timers—10.0 Mhz (clock input - not oscillator)

These are the crystals which directly tie to the devices in the chipset. In addition, both devices share a connection to the SBus clock and an

optional JTAG clock. The Ethernet controller receives its' clock from the T7213 DISC, which requires a 20.0 Mhz crystal. It should be noted that while the crystals above have been broken ouy by the chip that the clock is used on, all of the crystals are connected to oscillators on the C100, and the required clocks are then output to the C105. The timer clock in the C105 ca be obtained by dividing the 40.0 Mhz SCSI crystal. The 32.0 Mhz clock for the floppy is only used in systems which require the 1 Mbps data rate. If 500 kbps is the highest rate required, this crystal can be left out.

Does the chipset support atomic transfers?

The DMA2 is not set up to perform atomic transfers on the SBus. It appears at the time that this is being written that atomic transfers are being eliminated with IEEE P1496 standardization of the SBus.

Does the NCR89C105 do byte or word acknowledges on the SBus for the 8 bit devices that it supports?

The C105 will always do a single byte transfer across the SBus when transferring data from any of the devices on the 8 bit expansion bus, including the 82077 and both 8530s internal to the part. Some of the other registers in the device are organized as words and half-words and do take advantage of the larger bus sizing.

Does the chipset support rerun?

The NCR89C100 master will support rerun as part of the function of the DMA2 core. The NCR89C105 is not capable of requesting a rerun.

How are the interrupts cleared out of the interrupt circuitry?

The interrupts are cleared by disabling and then re-enabling them in the interrupt controller registers. Responding to an interrupt will clear it out.

Why doesn't the chipset support 16 word bursts?

The chipset does not support 16 word bursts because it would require larger buffers and the initial target systems will not be able to utilize a 16 word burst.

Have any bandwidth studies been done on the chipset?

No formal studies of the bandwidth of these devices on the SBus have been done aside from what falls out of normal system debug and testing. It should be noted, however, that the greatest impact on system bandwidth will typically be a function of the SBus controller chip. This is due to the overhead of the address translation and the time required to grant the bus.

Will the NCR89C100 support differential SCSI operation with external logic?

The NCR89C100 is set up to run exclusively in single-ended mode. The additional pins that would be required for differential SCSI are not available.

Why is the processor timer configurable as either a 64-bit or 22-bit timer? Why is it not always a 64-bit counter?

The processor timer is configurable for either 64-bit or 22-bit operation to maintain compliance with system architecture specification that it was designed to.

Is it possible to support multiple instances of the devices in the same system?

The devices can appear multiple times upon the same SBus. There can be one NCR89C100 master device per slot, and each of these may be paired with an NCR89C105 slave device in the same slot. It would also be possible, by adding an inverter to the system, to place two NCR89C105s in the same SBus slot.

What systems can the SBus I/O demonstration board run on?

The SBus I/O demonstration board that we provide is targeted for a SPARCstation 2 running the Solaris 2.1 O/S. The board can be plugged into a SPARCstation 2 running Sun O/S 4.1.x and successfully boot, however there is difficulty getting all of the drivers under the O/S to work. This is due to the fact that both the serial and floppy driver can only support one device in the system. When the board is plugged in, a second floppy controller and a second serial controller (in addition to the devices already in an SS-2) are introduced. Special drivers have been developed under Solaris 2.1 that recognize the serial and floppy ports on the I/O board, rather than the ones resident on the SS-2. These drivers will be sent to all customers who have an SBus I/O board when the Solaris 2.1 release of the board becomes available.

In addition, the Sun O/S 4.1.x drivers are not able to take advantage of the fast SCSI port (10 MB/sec) or the parallel port, as these features did not exist when the drivers were written. The Solaris 2.1 O/S is the first O/S designed specifically to work with the NCR89C100 and NCR89C105 devices. As such, the drivers utilize all of the new features that the chipset offers. For this reason, Sun is not supporting any O/S prior to Solaris 2.1 on the new workstations that they are shipping with the chipset (SPARCstation LX and SPARC Classic).

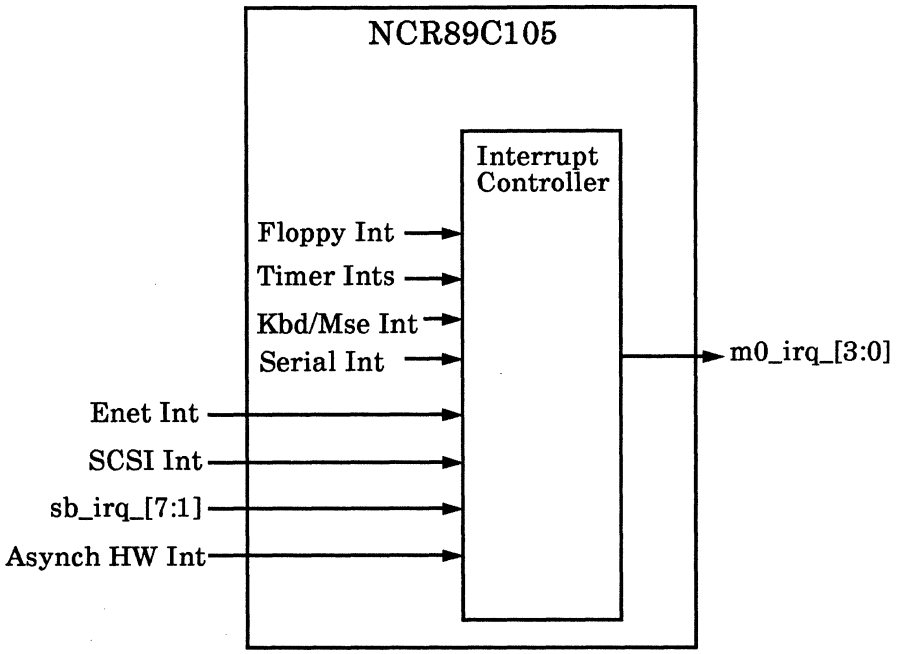
What configuration does the floppy controller in the NCR89C105 operate in?

The floppy controller in the NCR89C105 is configured internally to run in the PC/AT mode.

1) What is the relationship between all of the interrupt signals on the NCR89C105 and the system interrupts?

The way that the interrupt controller in the 89C105 works is that it collects all of the interrupts in the system and then feeds them to the processor over the m0_irq lines. The interrupts collected include ones found on the SBus (sb_irq_[7:1]), ones internal to the C105 (floppy, serial, kbd/mse, timers), and ones external to the C105 (SCSI, Enet) and possibly an asynchronous HW interrupt from the processor, depending on which processor is used. The follwing diagram shows how each of these

interrupts comes in and goes out of the chip.



The interrupt control section of the data manual shows a listing of how each of the interrupts coming into the C105 is mapped to the interrupt bus coming out. The relationships are as follows:

Level 0 - 15: this is the interrupt level output by the C105. This corresponds to output lines m0_irq_[3:0].

SBus L(1-7): these are the SBus interrupt levels, sb_irq_[7:1]

In addition, each of the other sources is mapped to a level on m0_irq_ lines and are listed. Level 15 is the highest interrupt level possible.

One thing that is worth noting here when designing a system using both the C100 and C105 is that the C105 has pins for the SCSI and Enet interrupts but not for the parallel port interrupt. The parallel port interrupt should be taken from the C100 pin and put on SBus level 2, which will cause it to appear as Level3 on the m0_irq_ lines.

Because the demonstration board plugs into an SPARCstation 2, which already has its' own interrupt controller, the PALs were added to handle this. All of the interrupts from the C100 and C105 are placed back on the SBus through the PALs, allowing them to be seen by the SS2 interrupt controller. It is also possible to write a register to set any of the SBus level interrupts which will go directly to the C105, allowing the user to observe how the C105 handles the interrupts off of the SBus. Note that although the internal interrupt circuitry in the C105 works on the demo board, the interrupts will come out on the m0_irq_ lines and will be placed on the SBus to be seen by the SS2 controller.

How do the enables work on the drivers for the parallel port?

The parallel port is designed so that there are direction signals for the Data lines, Data_strobe_L, Acknowledge_L, and Busy signals. This direction signal will guard the C100 device from any problems, as it has full control of these signals. On the demo board, the output enables are tied high on the BCT245 and the ABT240. This will not cause any problems for the C100. It is necessary for the designer to make sure that it will not cause any problems for the devices that are intended to be used with the parallel port. There is some opportunity to define what the parallel port will actually be like in the system that is designed and to design for a desired data transfer protocol. Again, this should be targeted to the devices that will be used by the parallel port. The paralle port implementation shown here is very similar to the one that is used in the LX, Classic and SS10. It is Centronics compatible (unidirectional, master writes) and capable of "Xerox mode" (master write/read) and "IBM mode" (master write, slave read). These protocols are identical in their write (output) mode used for pport printers. The input modes are the only ones that differ. Looking at the "Bidirectional Parallel Port Interface" section of the DMA2 chapter of the data manual should help clarify this.

Is the Power Fail signal on the NCR89C105 edge or level sensitive?

This signal is level sensitive.

Where is the parallel port interrupt signal connected to the interrupt controller?

The parallel port is intended to interrupt at SBus level 2. It does not actually tie directly to the C105, but instead ties to the SBus level2 interrupt signal (the sb_p_irq_ pin is open-drain). The C105 monitors the seven SBus interrupt signals and will then recognize the parallel port interrupt.

Will the NVRAM access 1 Mbyte?

The space reserved for both the TOD and NVRAM chip is 1 MByte wide. Any space outside of the TOD portion can be used for NVRAM. If you choose to use this entire space, it is necessary to take this into consideration when designing the EBus address latch.

Sample Kit Errata Information

In the early stages of the NCR SBus I/O chipset, the sample parts contain six known differences from the actual production version of the chipset. This section details these differences. Sample versions of the devices can be found in early sample kit packages, on early demonstration boards and in early small quantity sample orders. There are a number of ways to detremine if a part is a sample version or not, as described below.

An NCR89C100 can be recognized as a sample part if it has the label NCR89C100-ES (Engineering Sample) or if it has the part number 0390771. The production version will have the part number 0391159 or 0391120 and will not have the -ES marking. Also, the label MACIO B.1 will indicate that the part is a sample version.

An NCR89C105 can be recognized as a sample part if it has the label NCR89C105-ES or if it has the part number 0390814. The production version will have the part number 0391187 or 0391167 and will not have the -ES marking. Also, the label SLAVIO B.2 will indicate that the part is a sample version.

It should be noted that while there are slight differences between the sample and production versions of each of these parts, these differences are very minor and should not hinder system debug efforts using these devices.

WARNING: The earliest sample parts were distributed with a paper label designed to look like the mark that will be placed on the production parts. These labels are flammable and can NOT be subjected to high heat, such as in a wave solder or vapor phase machine.

NCR89C100 Errata

92C990 Multicast Addressing

There is a special condition where a multicast addressed packet on the Ethernet can leak through the logical address filter in the 92C990. This condition occurs when the rising edges of the RCLK and TCLK signals align with the internal multicast abort signal, causing the abort signal to be masked. The production silicon is designed to insure that the abort signal is recognized in all variations of RCLK and TCLK.

92C990 Loss of Carrier

When the 92C990 loses the carrier, it will try 16 times to obtain the signal. After the 16th failed try to recover the carrier, the part should report a loss of carrier. If, however, a collision is detected on the 16th try, the industry standard part will not report the loss of carrier, while the 92C990 in the sample silicon will still report this error. In the production silicon, the loss of carrier error is masked in the situation where a collision occurs during the 16th retry.

92C990 Stop/Init

When the 92C990 is issued a Stop command, it is necessary to reset the part before issuing an Initialize command. This is because the receiver may be receiving data when the Stop command is sent, and it will continue receiving data while other parts of the controller halt, potentially corrupting the data. It will be unnecessary to reset the part after a Stop command in the production version of the part, as this has been corrected.

NCR89C105 Errata

85C30 Break Status

When parity is enabled in the full function 85C30 SCC and a break occurs, a Special Condition Receive interrupt is reported when it actually should be reported as an External Status Change interrupt. This occurs due to an internal inversion of the parity signal. This is not an issue with the 85C30 keyboard/mouse controller. The correct polarity of the parity signal is used in the production version of the parts.

85C30 Break Interrupt

When an external status interrupt occurs, signifying the end of a break condition, the 85C30 should simultaneously generate a break for external status and received data (a null byte). The version of the 85C30 in the sample silicon (both serial ports and keyboard/mouse ports) will generate the external status interrupt immediately and then take up to one receive clock to generate the receive data interrupt. Software drivers designed to automatically respond to the receive data interrupt first will access the null byte before it is ready to transfer, causing a sequencing error in the part. The production parts are modified to generate these two interrupts simultaneously.

EPROM Address Space

In the sample version of the NCR89C105, the address space for the EPROM on the EBus is limited to 4 megabytes. In the production version of the part, this address space is extended to 16 megabytes.

SBus Demonstration Board

The NCR SBus I/O Chipset Demonstration Board is intended for use in a SPARCstation 2 workstation and occupies two SBus slots. A boot prom is included on this board that will allow an SS-2 to boot and create a space in the device tree for each of the devices found on the board. This section contains address maps, PAL equations, and schematics for the demo board.

Demo Board Address Map

Following is the base address map for each of the devices on the demonstration board. Note that these addresses differ from the addresses given in each of the chip-level address maps of the NCR89C100 and NCR89C105 found in their respective sections.

NCR89C105 Address Map

PA[27:0]	Device
000 0000 -> 3ff ffff	Boot PROM
040 0000 -> 047 ffff	Keyboard, Mouse, Serial Ports
048 0000 -> 04b ffff	TOD/NVRAM
04c 0000 -> 04d ffff	General Purpose (Gen. Port)
050 0000 -> 053 ffff	Floppy Controller
054 0000 -> 05f ffff	Reserved
060 0000	89C105 Configuration Reg.
064 0000 -> 067 ffff	Auxiliary I/O Registers
068 0000	Diagnostic Message Register
06c 0000	Modem Register
070 0000 -> 071 ffff	Reserved
074 0000 -> 077 ffff	Counter/timers
078 0000 -> 07b ffff	Interrupt Controller
07c 0000	System Control/Status Register

NCR89C100 Address Map

PA[27:0]	Device
100 0000	DMA2 Internal ID Register
120 0000 -> 120 000f	DMA2 ESP Registers
120 0010 -> 120 001f	DMA2 Ethernet Registers
1c0 0000 -> 1c0 001f	DMA2 Parallel Port Registers
140 0000 -> 140 003f	SCSI Controller Registers
160 0000 -> 160 0003	Ethernet Controller Registers

Demo Board PLD Equations

This section contains the equations used to program the PALs and GALs which are found on the demonstration board. The programmable devices are labeled both on the demonstration board and in the schematic as REG0, REG1, and INT_CTL.

Register 0 (REG0)

```
module _reg000 flag '-r3';
title ' write control for register 0'

reg000device 'P22V10';

SB_CLK,D0,D1,D2,D3,D4,D5,D6,D7 pin 1,13,11,10,9,8,7,6,5;
GENCS_L,EB_WR_L,A2,SB_RESET_L pin 4,3,2,23;

GENERIC_RDY_L,INT_ENABLE,INT_SELECT pin 22,21,19;
PFD,MODERR_L,IU_ERROR,MSI_IRQ pin 18,17,16,15;
REG0_CS_L,VALID_CYCLE pin 14,20;

write_cycle = (!GENCS_L & !EB_WR_L & !A2);

equations

!REG0_CS_L := !GENCS_L & !A2 & SB_RESET_L # !REG0_CS_L & GENERIC_-
RDY_L & SB_RESET_L;

INT_SELECT := !GENCS_L & D0 & SB_RESET_L & !EB_WR_L & !A2 # INT_SE-
LECT & SB_RESET_L & !write_cycle;

!MODERR_L := !GENCS_L & D1 & SB_RESET_L & !EB_WR_L & !A2 # !MOD-
ERR_L & SB_RESET_L & !write_cycle;

!MSI_IRQ := !GENCS_L & D2 & SB_RESET_L & !EB_WR_L & !A2 # !MSI_IRQ &
SB_RESET_L & !write_cycle;

!PFD := !GENCS_L & D3 & SB_RESET_L & !EB_WR_L & !A2 # !PFD & SB_RE-
SET_L & !write_cycle;

!IU_ERROR := !GENCS_L & D4 & SB_RESET_L & !EB_WR_L & !A2 # IU_ERROR &
SB_RESET_L & !write_cycle;

VALID_CYCLE := !GENCS_L & SB_RESET_L;

INT_ENABLE := VALID_CYCLE & D5 & !EB_WR_L & !A2 # INT_ENABLE &
SB_RESET_L & !write_cycle;

!GENERIC_RDY_L := VALID_CYCLE) # (!GENERIC_RDY_L & !GENCS_L &
SB_RESET_L);
end
```

Register 1 (REG1)

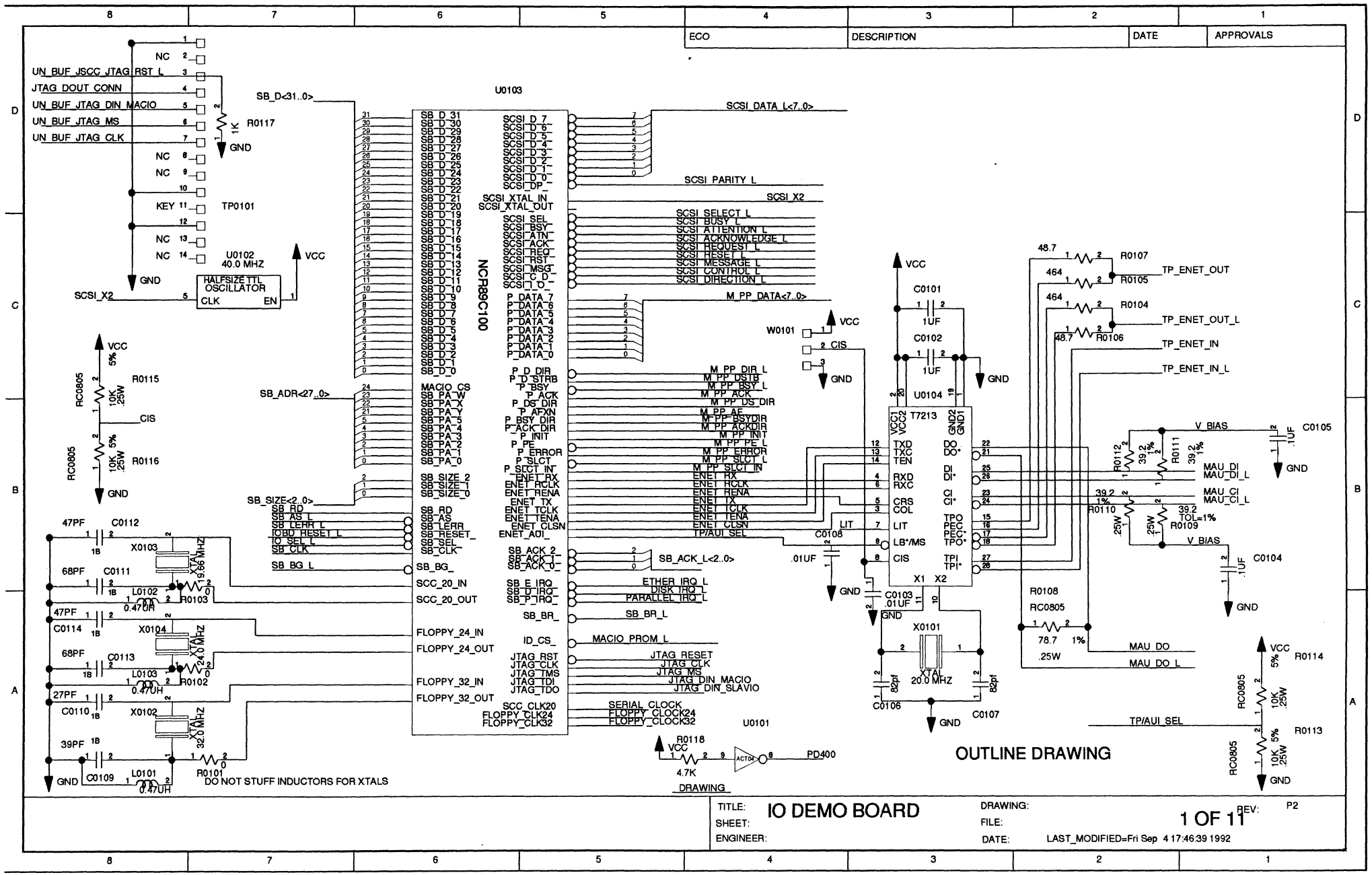
```
module _reg111 flag '-r3';
title ' write control for register 1'

reg111 device 'P22V10';

SB_CLK,D0,D1,D2,D3,D4,D5,D6,D7 pin 1,13,11,10,9,8,7,6,5;
GENCS_L,A2,EB_WR_L,SB_RESET_L pin 4,3,2,23;
INT_SEL pin 22;

IOSB_IRQ7_L,IOSB_IRQ6_L,IOSB_IRQ5_L,IOSB_IRQ4_L pin 20,19,18,17;
IOSB_IRQ3_L,IOSB_IRQ2_L_MUX,IOSB_IRQ1_L pin 16,15,14;
REG1_CS_L pin 21;

write_cycle = (!GENCS_L & !EB_WR_L & A2);

equations

!REG1_CS_L := !GENCS_L & A2 & SB_RESET_L # !REG1_CS_L & GENCS_L &
SB_RESET_L;

!IOSB_IRQ7_L := !GENCS_L & A2 & !EB_WR_L & D6 & SB_RESET_L # !IOS-
B_IRQ7_L & SB_RESET_L & !write_cycle;

!IOSB_IRQ6_L := !GENCS_L & A2 & !EB_WR_L & D5 & SB_RESET_L # !IOS-
B_IRQ6_L & SB_RESET_L & !write_cycle;

!IOSB_IRQ5_L := !GENCS_L & A2 & !EB_WR_L & D4 & SB_RESET_L # !IOS-
B_IRQ5_L & SB_RESET_L & !write_cycle;

!IOSB_IRQ4_L := !GENCS_L & A2 & !EB_WR_L & D3 & SB_RESET_L # !IOS-
B_IRQ4_L & SB_RESET_L & !write_cycle;

!IOSB_IRQ3_L := !GENCS_L & A2 & !EB_WR_L & D2 & SB_RESET_L # !IOS-
B_IRQ3_L & SB_RESET_L & !write_cycle;

!IOSB_IRQ2_L_MUX := !GENCS_L & A2 & !EB_WR_L & D1 & SB_RESET_L # !IOS-
B_IRQ2_L_MUX & SB_RESET_L & !write_cycle;

!IOSB_IRQ1_L := !GENCS_L & A2 & !EB_WR_L & D0 & SB_RESET_L # !IOS-
B_IRQ1_L & SB_RESET_L & !write_cycle;

end
```

Interrupt Control Register (INT_CTL)

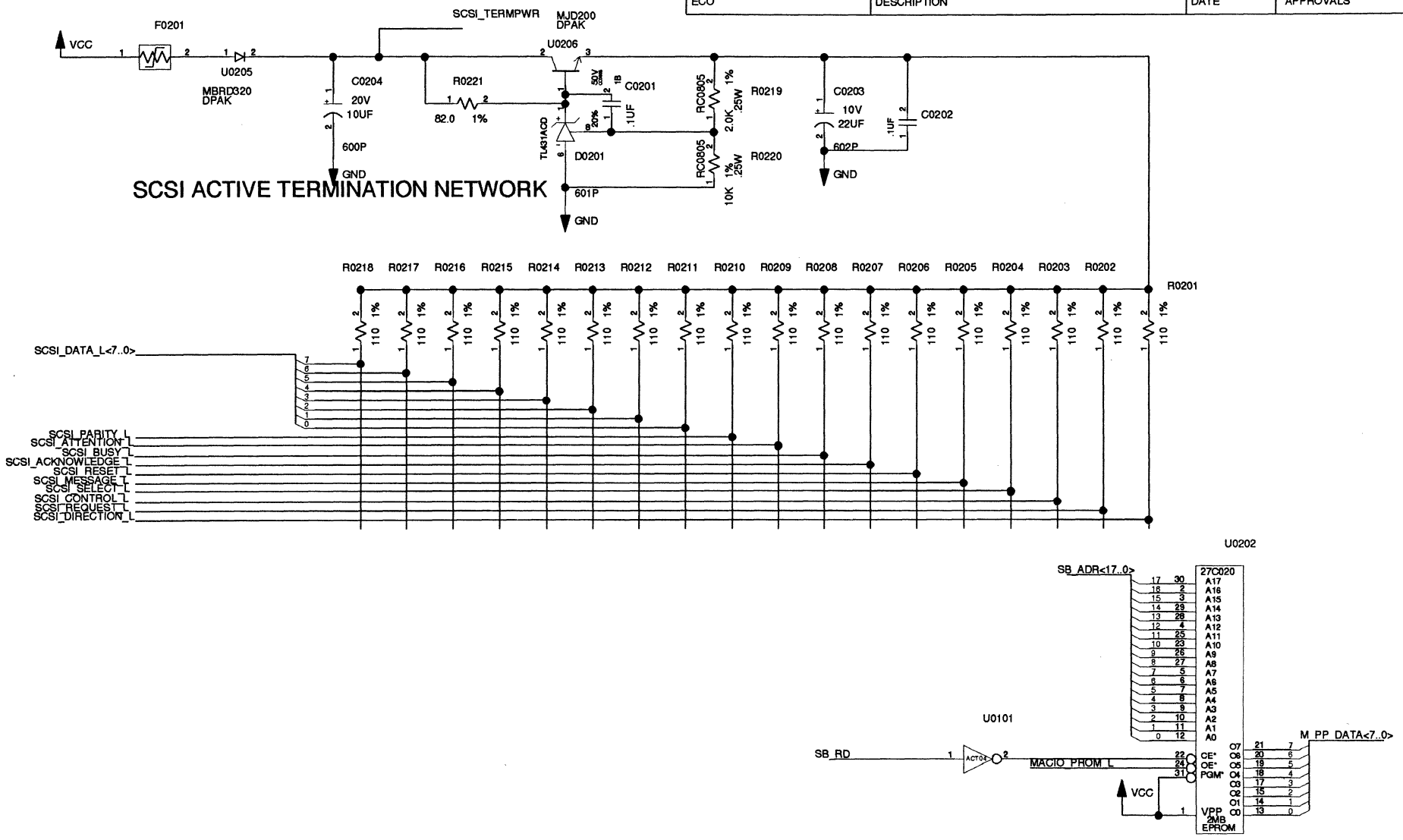
```
module _int_ctl flag '-r3';
title ' Interrupt control logic for IO test board '

int_ctl device 'P16V8C';

SLAVIO_IRL3,SLAVIO_IRL2,SLAVIO_IRL1,SLAVIO_IRL0 pin 5,4,3,2;
IO_SB_INTR_L2_MUX,INT_SELECT,SB_RESET_L,PP_IRQ_L,DISK_IRQ_L,ETHE
R_IRQ_L pin 1,6,7,8,9,11;
INT_ENABLE pin 17;

IO_SB_INTR_L2,IO_INTERRUPT,ETHERNET,SCSI,PARALLEL pin 19,18,15,14,13;

SLAVIO_INT = [SLAVIO_IRL3,SLAVIO_IRL2,SLAVIO_IRL1,SLAVIO_IRL0];

equations

!IO_SB_INTR_L2 = (!PP_IRQ_L & INT_ENABLE & INT_SELECT)  # (!IO_SB_IN-
TR_L2_MUX & INT_ENABLE & !INT_SELECT);

IO_INTERRUPT = (INT_ENABLE) & (SLAVIO_INT != ^b0000) ;

ETHERNET = INT_ENABLE & !ETHER_IRQ_L;

SCSI = INT_ENABLE & !DISK_IRQ_L;

PARALLEL = INT_ENABLE & !PP_IRQ_L;

end
```

Demo Board Schematic

The following pages contain the complete schematic for the NCR SBus I/O demonstration board.

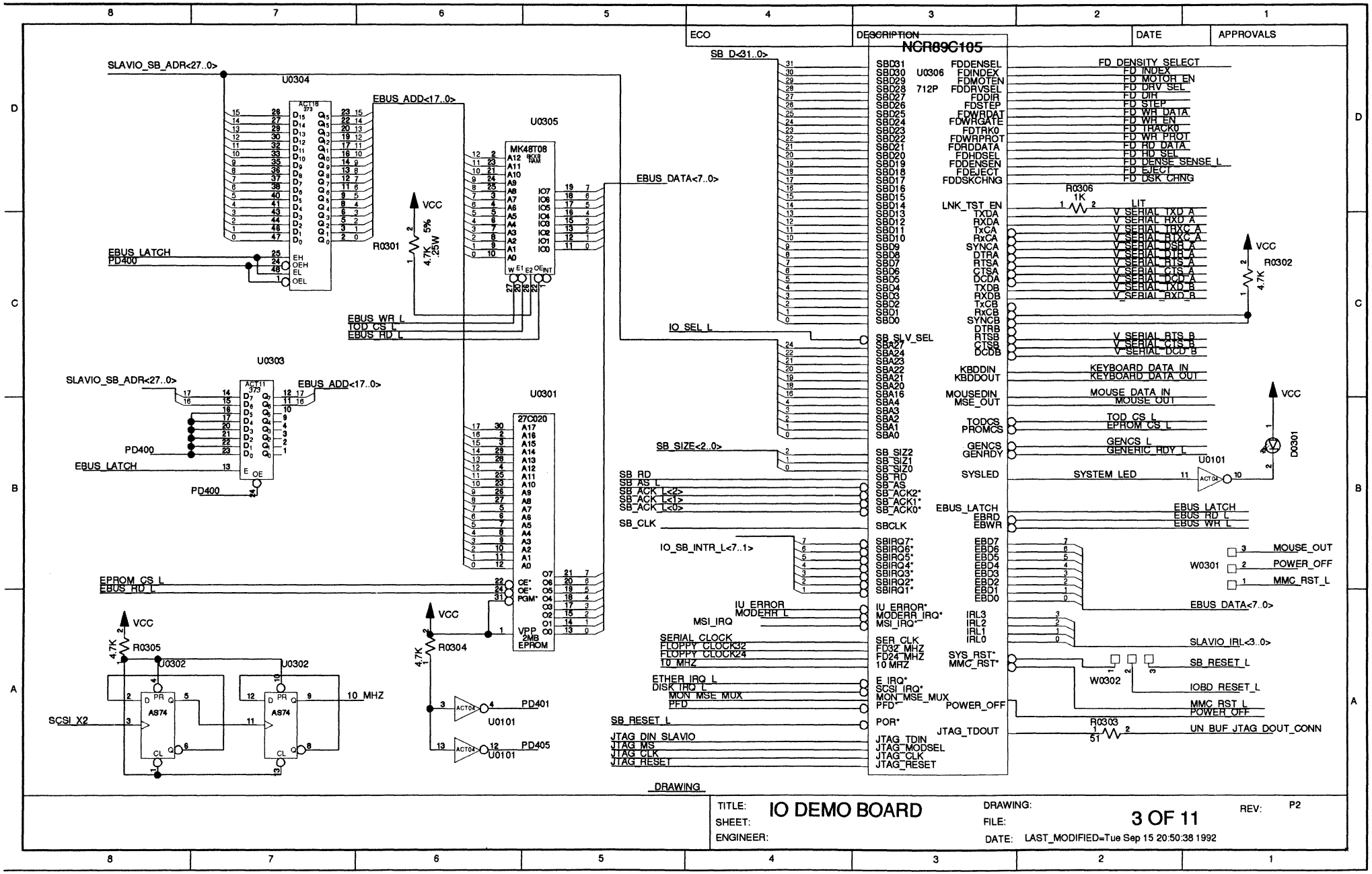

ECO
DESCRIPTION
DATE
APPROVALS
U0103
NCR89C100
SCSI_D_7
SCSI_D_6
SCSI_D_5
SCSI_D_4
SCSI_D_3
SCSI_D_2
SCSI_D_1
SCSI_D_0
SCSI_DP_
SCSI_XTAL_IN
SCSI_XTAL_OUT
SCSI_SEL_
SCSI_BSY_
SCSI_ATN_
SCSI_ACK_
SCSI_REQ_
SCSI_RST_
SCSI_MSG_
SCSI_C_D_
SCSI_I_O_
P_DATA_7
P_DATA_6
P_DATA_5
P_DATA_4
P_DATA_3
P_DATA_2
P_DATA_1
P_DATA_0
P_D_DIR
P_D_STRB
P_BSY
P_ACK
P_DS_DIR
P_AFXN
P_BSY_DIR
P_ACK_DIR
P_INIT
P_PE
P_ERROR
P_SLCT
P_SLCT_IN
ENET_RX
ENET_RCLK
ENET_RENA
ENET_TX
ENET_TCLK
ENET_TENA
ENET_CLSN
ENET_AUI_
SB_ACK_2
SB_ACK_1
SB_ACK_0
SB_E_IRQ
SB_D_IRQ
SB_P_IRQ
SB_BR_
ID_CS_
JTAG_RST
JTAG_CLK
JTAG_TMS
JTAG_TDI
JTAG_TDO
SCC_CLK20
FLOPPY_CLK24
FLOPPY_CLK32
MACIO_CS
SB_PA_W
SB_PA_X
SB_PA_Y
SB_PA_5
SB_PA_4
SB_PA_3
SB_PA_2
SB_PA_1
SB_PA_0
SB_SIZE_2
SB_SIZE_1
SB_SIZE_0
SB_RD
SB_AS
SB_LERR_L
TOBD_RESET_L
IO_SEL_L
SB_CLK
SB_BG_
SCC_20_IN
SCC_20_OUT
FLOPPY_24_IN
FLOPPY_24_OUT
FLOPPY_32_IN
FLOPPY_32_OUT
SB_D<31..0>
SB_ADR<27..0>
SB_SIZE<2..0>
SCSI_DATA_L<7..0>
SCSI_PARITY_L
SCSI_X2
SCSI_SELECT_L
SCSI_BUSY_L
SCSI_ATTENTION_L
SCSI_ACKNOWLEDGE_L
SCSI_REQUEST_L
SCSI_RESET_L
SCSI_MESSAGE_L
SCSI_CONTROL_L
SCSI_DIRECTION_L
M_PP_DATA<7..0>
M_PP_DIR_L
M_PP_DSTB
M_PP_BSY_L
M_PP_ACK
M_PP_DS_DIR
M_PP_AF
M_PP_BSYDIR
M_PP_ACKDIR
M_PP_INIT
M_PP_PE_L
M_PP_ERROR
M_PP_SLCT_L
M_PP_SLCT_IN
ENET_RX
ENET_RCLK
ENET_RENA
ENET_TX
ENET_TCLK
ENET_TENA
ENET_CLSN
TP/AUI_SEL
SB_ACK_L<2..0>
ETHER_IRQ_L
DISK_IRQ_L
PARALLEL_IRQ_L
SB_BR_L
MACIO_PROM_L
JTAG_RESET
JTAG_CLK
JTAG_MS
JTAG_DIN_MACIO
JTAG_DIN_SLAVIO
SERIAL_CLOCK
FLOPPY_CLOCK24
FLOPPY_CLOCK32
UN_BUF_JSCC_JTAG_RST_L
JTAG_DOUT_CONN
UN_BUF_JTAG_DIN_MACIO
UN_BUF_JTAG_MS
UN_BUF_JTAG_CLK
R0117
1K
GND
NC
KEY
TP0101
U0102
40.0 MHZ
HALFSIZE TTL OSCILLATOR
CLK
EN
VCC
R0115
10K .25W 5%
R0116
RC0805
CIS
47PF C0112
X0103
19.66 MHZ
XTAL
68PF C0111
L0102
0.47UH
R0103
0
47PF C0114
X0104
24.0 MHZ
68PF C0113
L0103
R0102
27PF C0110
X0102
32.0 MHZ
39PF
C0109
L0101
R0101
DO NOT STUFF INDUCTORS FOR XTALS
W0101
C0101 1UF
C0102 1UF
U0104
T7213
VCC1
VCC2
GND1
GND2
TXD
TXC
TEN
RXD
RXC
CRS
COL
LIT
LB*/MS
CIS
DO
DO*
DI
DI*
CI
CI*
TPO
PEC
PEC*
TPO*
TPI
TPI*
X1
X2
C0103 .01UF
C0108 .01UF
X0101
20.0 MHZ
XTAL
82pf
C0106
C0107
R0118
4.7K
ACT04
PD400
U0101
DRAWING
48.7
464
R0107
R0105
R0104
R0106
TP_ENET_OUT
TP_ENET_OUT_L
TP_ENET_IN
TP_ENET_IN_L
V_BIAS
C0105
C0104
R0112
R0111
R0110
R0109
39.2 1%
39.2 TOL=1%
MAU_DI
MAU_DI_L
MAU_CI
MAU_CI_L
R0108
78.7 1%
.25W
MAU_DO
MAU_DO_L
R0114
R0113
10K .25W 5%
OUTLINE DRAWING
TITLE: IO DEMO BOARD
SHEET:
ENGINEER:
DRAWING:
FILE:
DATE: LAST_MODIFIED=Fri Sep 4 17:46:39 1992
REV: P2
1 OF 11

ECO	DESCRIPTION	DATE	APPROVALS



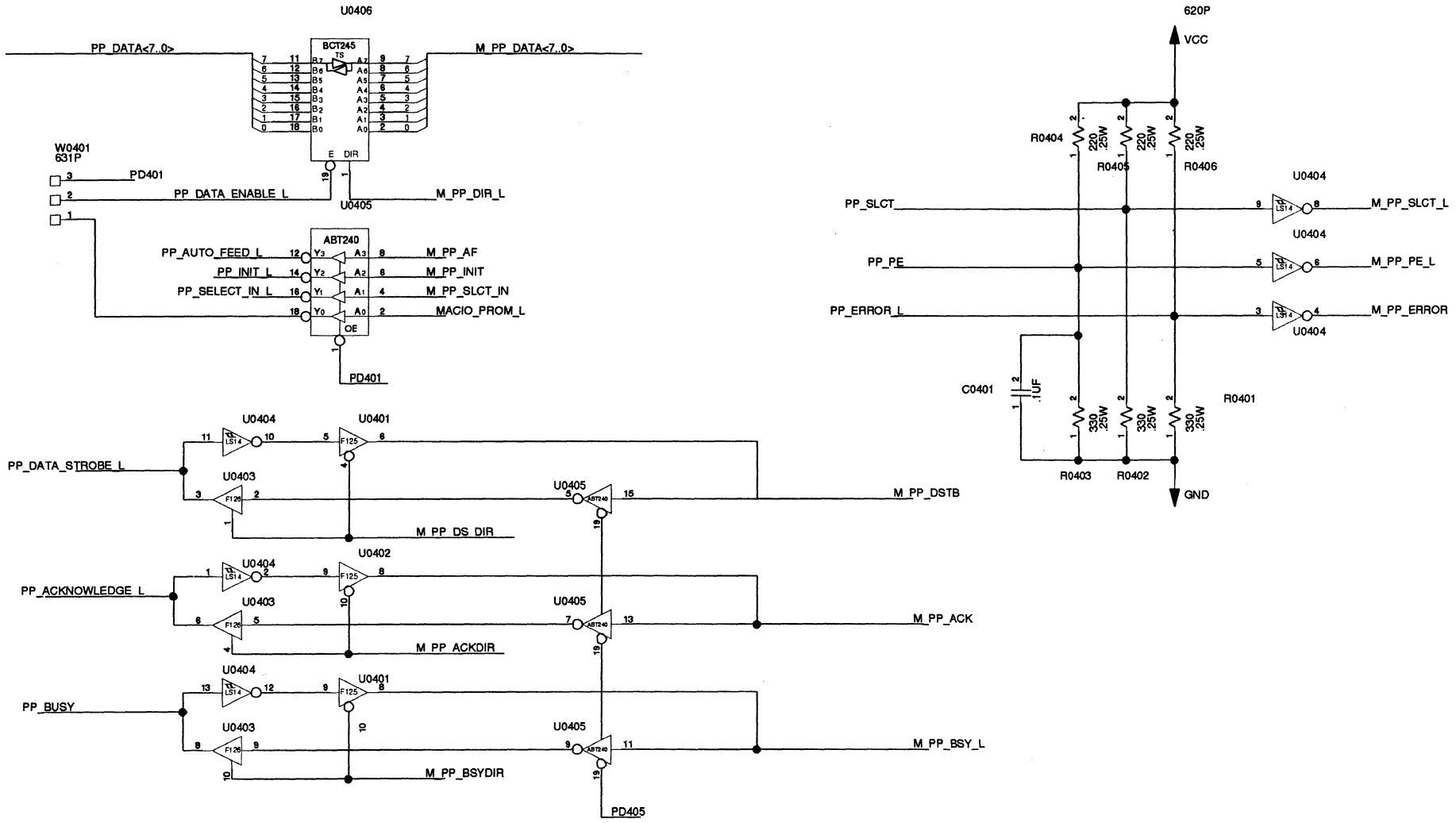
DRAWING

TITLE: IO DEMO BOARD	DRAWING:		REV: P2
SHEET:	FILE:	2 OF 11	
ENGINEER:	DATE:	LAST_MODIFIED=Tue Jun 23 11:24:15 1992	


ECO
DESCRIPTION
DATE
APPROVALS
NCR89C105
SLAVIO_SB_ADR<27..0>
EBUS_ADD<17..0>
EBUS_DATA<7..0>
SB_D<31..0>
U0304
U0305
U0303
U0301
R0301
R0302
R0303
R0304
R0305
R0306
W0301
W0302
EBUS_LATCH
PD400
EBUS_WR_L
TOD_CS_L
EBUS_RD_L
IO_SEL_L
SB_SIZ<2..0>
SB_RD
SB_AS_L
SB_ACK_L<2>
SB_ACK_L<1>
SB_ACK_L<0>
SB_CLK
IO_SB_INTR_L<7..1>
IU_ERROR
MODERR_L
MSI_IRQ
SERIAL_CLOCK
FLOPPY_CLOCK32
FLOPPY_CLOCK24
10_MHZ
ETHER_IRQ_L
DISK_IRQ_L
MON_MSE_MUX
PFD
SB_RESET_L
JTAG_DIN_SLAVIO
JTAG_MS
JTAG_CLK
JTAG_RESET
EPROM_CS_L
SCSI_X2
FDDENSEL
FDINDEX
FDMOTEN
FDDRVSEL
FDDIR
FDSTEP
FDWRDAT
FDWRGATE
FDTRK0
FDWRPROT
FDRDDATA
FDHDSEL
FDDENSEN
FDEJECT
FDDSKCHNG
LNK_TST_EN
TXDA
RXDA
TxCA
RxCA
SYNCA
DTRA
RTSA
CTSA
DCDA
TXDB
RXDB
TxCB
RxCB
SYNCB
DTRB
RTSB
CTSB
DCDB
KBDDIN
KBDDOUT
MOUSEDIN
MSE_OUT
TODCS
PROMCS
GENCS
GENRDY
SYSLED
EBRD
EBWR
EBD7
EBD6
EBD5
EBD4
EBD3
EBD2
EBD1
EBD0
IRL3
IRL2
IRL1
IRL0
SYS_RST*
MMC_RST*
POWER_OFF
JTAG_TDOUT
JTAG_TDIN
JTAG_MODSEL
JTAG_CLK
JTAG_RESET
FD_DENSITY_SELECT
FD_INDEX
FD_MOTOR_EN
FD_DRV_SEL
FD_DIR
FD_STEP
FD_WR_DATA
FD_WR_EN
FD_TRACK0
FD_WR_PROT
FD_RD_DATA
FD_HD_SEL
FD_DENSE_SENSE_L
FD_EJECT
FD_DSK_CHNG
V_SERIAL_TXD_A
V_SERIAL_RXD_A
V_SERIAL_TRXC_A
V_SERIAL_RTXC_A
V_SERIAL_DSR_A
V_SERIAL_DTR_A
V_SERIAL_RTS_A
V_SERIAL_CTS_A
V_SERIAL_DCD_A
V_SERIAL_TXD_B
V_SERIAL_RXD_B
V_SERIAL_RTS_B
V_SERIAL_CTS_B
V_SERIAL_DCD_B
KEYBOARD_DATA_IN
KEYBOARD_DATA_OUT
MOUSE_DATA_IN
MOUSE_OUT
TOD_CS_L
EPROM_CS_L
GENCS_L
GENERIC_RDY_L
SYSTEM_LED
D0301
U0101
MOUSE_OUT
POWER_OFF
MMC_RST_L
SLAVIO_IRL<3..0>
SB_RESET_L
IOBD_RESET_L
MMC_RST_L
POWER_OFF
UN_BUF_JTAG_DOUT_CONN
VCC
5%
.25W
4.7K
1K
LIT
MK48T08
27C020
2MB EPROM
AS74
ACT373
ACT11
ACT04
DRAWING
TITLE: IO DEMO BOARD
SHEET:
ENGINEER:
DRAWING:
FILE:
3 OF 11
REV: P2
DATE: LAST_MODIFIED=Tue Sep 15 20:50:38 1992

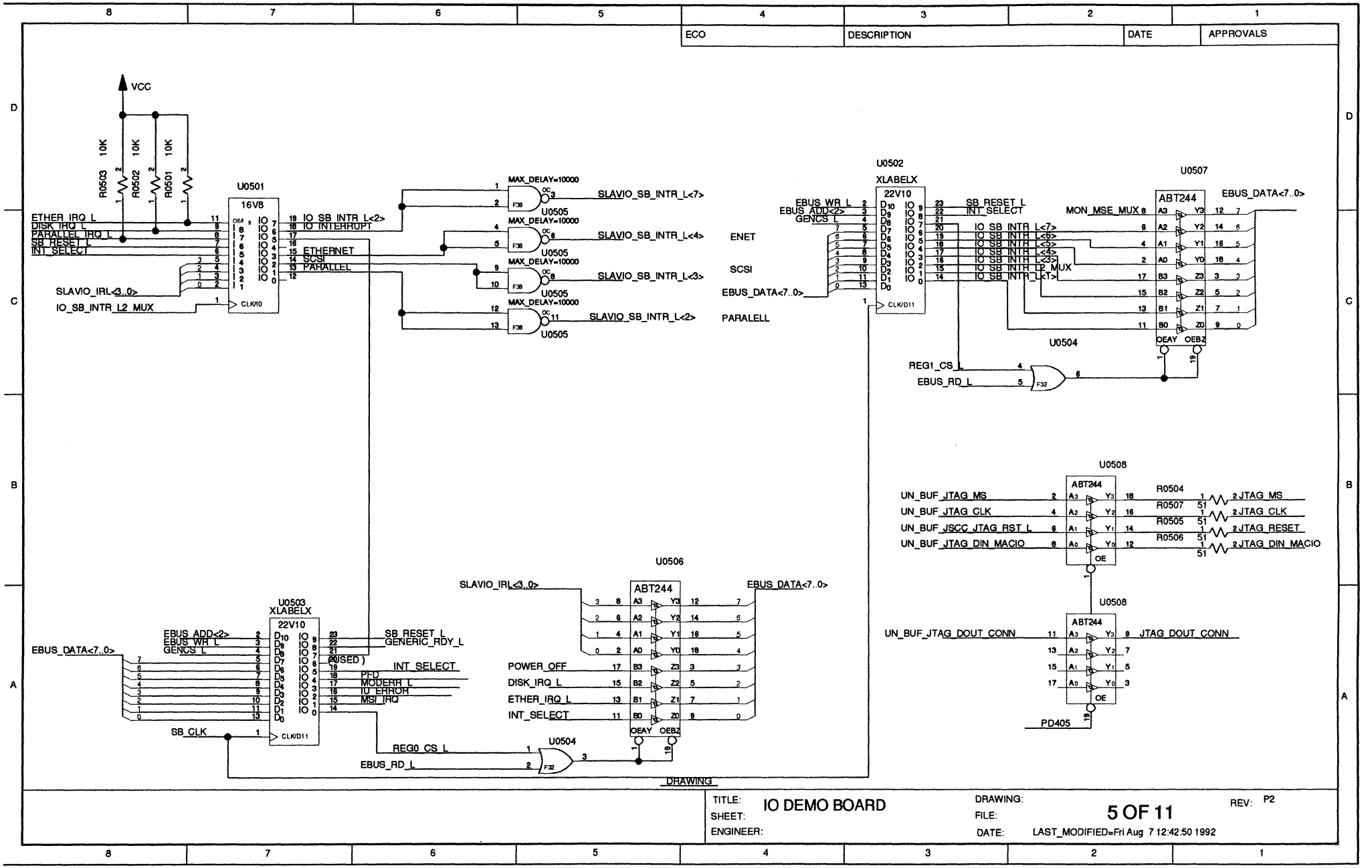
PARALLEL PORT BUFFERS / TRANSCEIVERS

ECO	DESCRIPTION	DATE	APPROVALS



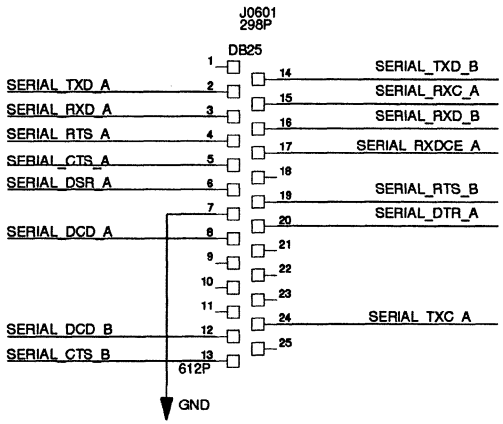
DRAWING

TITLE: IO DEMO BOARD	DRAWING: 4 OF 11	REV: P2
SHEET:	FILE:	
ENGINEER:	DATE: LAST_MODIFIED=Tue Jun 23 11:27:34 1992	

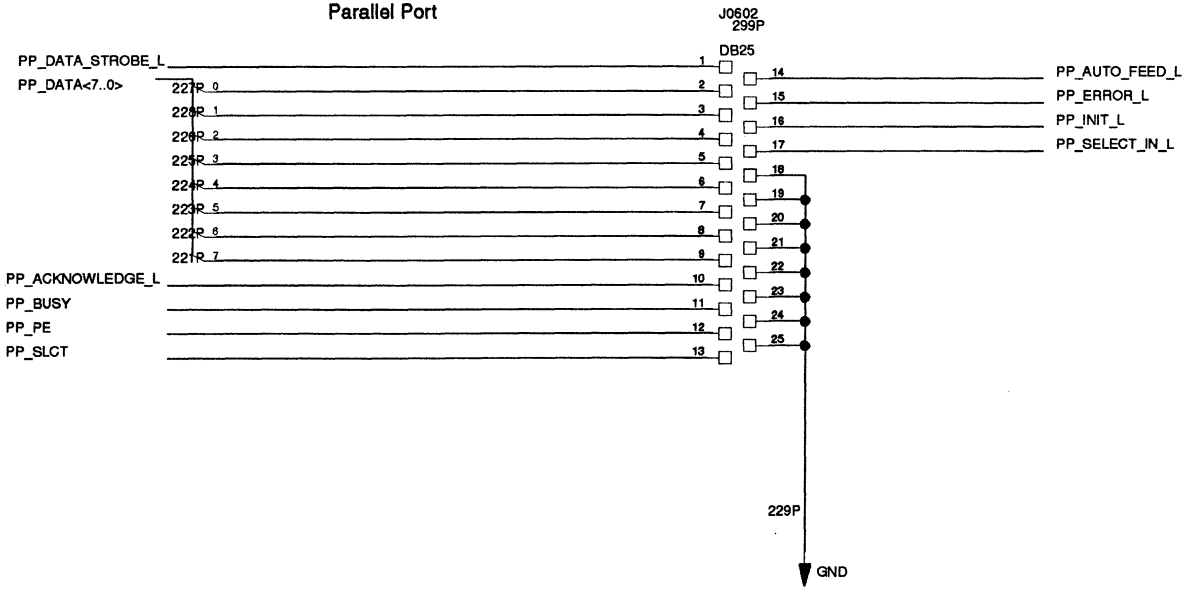

ECO
DESCRIPTION
DATE
APPROVALS
VCC
R0503 10K
R0502 10K
R0501 10K
U0501
16V8
ETHER_IRQ_L
DISK_IRQ_L
PARALLEL_IRQ_L
SB_RESET_L
INT_SELECT
SLAVIO_IRL<3..0>
IO_SB_INTR_L2_MUX
IO_SB_INTR_L<2>
IO_INTERRUPT
ETHERNET
SCSI
PARALLEL
MAX_DELAY=10000
U0505
SLAVIO_SB_INTR_L<7>
SLAVIO_SB_INTR_L<4>
SLAVIO_SB_INTR_L<3>
SLAVIO_SB_INTR_L<2>
ENET
SCSI
EBUS_DATA<7..0>
PARALELL
U0502
XLABELX
22V10
EBUS_WR_L
EBUS_ADD<2>
GENCS_L
SB_RESET_L
INT_SELECT
IO_SB_INTR_L<7>
IO_SB_INTR_L<6>
IO_SB_INTR_L<5>
IO_SB_INTR_L<4>
IO_SB_INTR_L<3>
IO_SB_INTR_L2_MUX
IO_SB_INTR_L<1>
CLK/D11
U0507
ABT244
MON_MSE_MUX
EBUS_DATA<7..0>
OEAY
OEBZ
U0504
REG1_CS_L
EBUS_RD_L
F32
U0508
UN_BUF_JTAG_MS
UN_BUF_JTAG_CLK
UN_BUF_JSCC_JTAG_RST_L
UN_BUF_JTAG_DIN_MACIO
R0504
R0507
R0505
R0506
51
JTAG_MS
JTAG_CLK
JTAG_RESET
JTAG_DIN_MACIO
OE
UN_BUF_JTAG_DOUT_CONN
JTAG_DOUT_CONN
PD405
U0503
XLABELX
22V10
EBUS_ADD<2>
EBUS_WR_L
GENCS_L
EBUS_DATA<7..0>
SB_RESET_L
GENERIC_RDY_L
(UNUSED)
INT_SELECT
PFD
MODERR_L
IO_ERROR
MSI_IRQ
SB_CLK
CLK/D11
U0506
ABT244
SLAVIO_IRL<3..0>
EBUS_DATA<7..0>
POWER_OFF
DISK_IRQ_L
ETHER_IRQ_L
INT_SELECT
OEAY
OEBZ
U0504
REG0_CS_L
EBUS_RD_L
F32
DRAWING
TITLE: IO DEMO BOARD
SHEET:
ENGINEER:
DRAWING:
FILE:
5 OF 11
REV: P2
DATE: LAST_MODIFIED=Fri Aug 7 12:42:50 1992

ECO	DESCRIPTION	DATE	APPROVALS

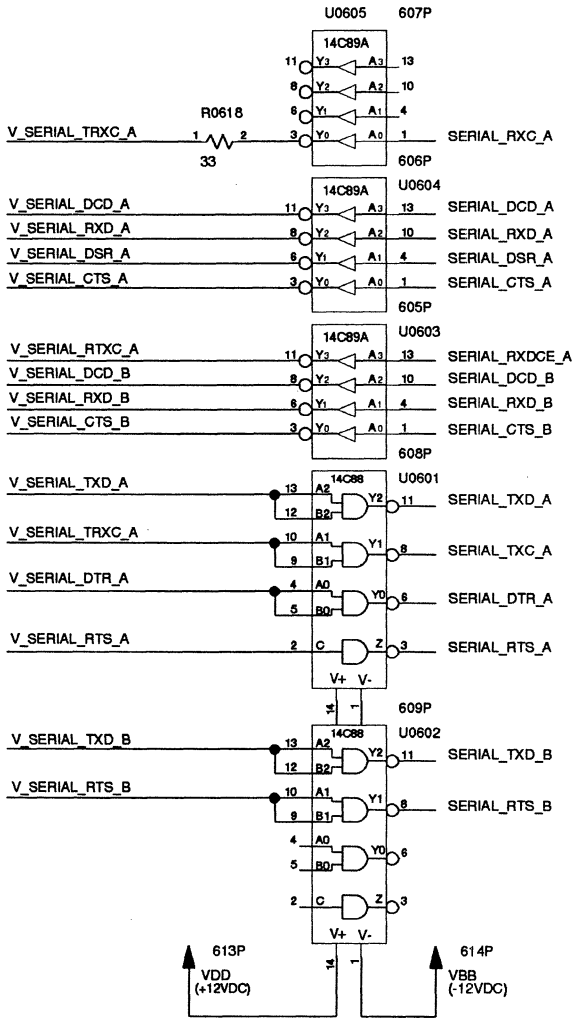
Serial Ports



Parallel Port



SERIAL PORT DRIVER / RECEIVERS



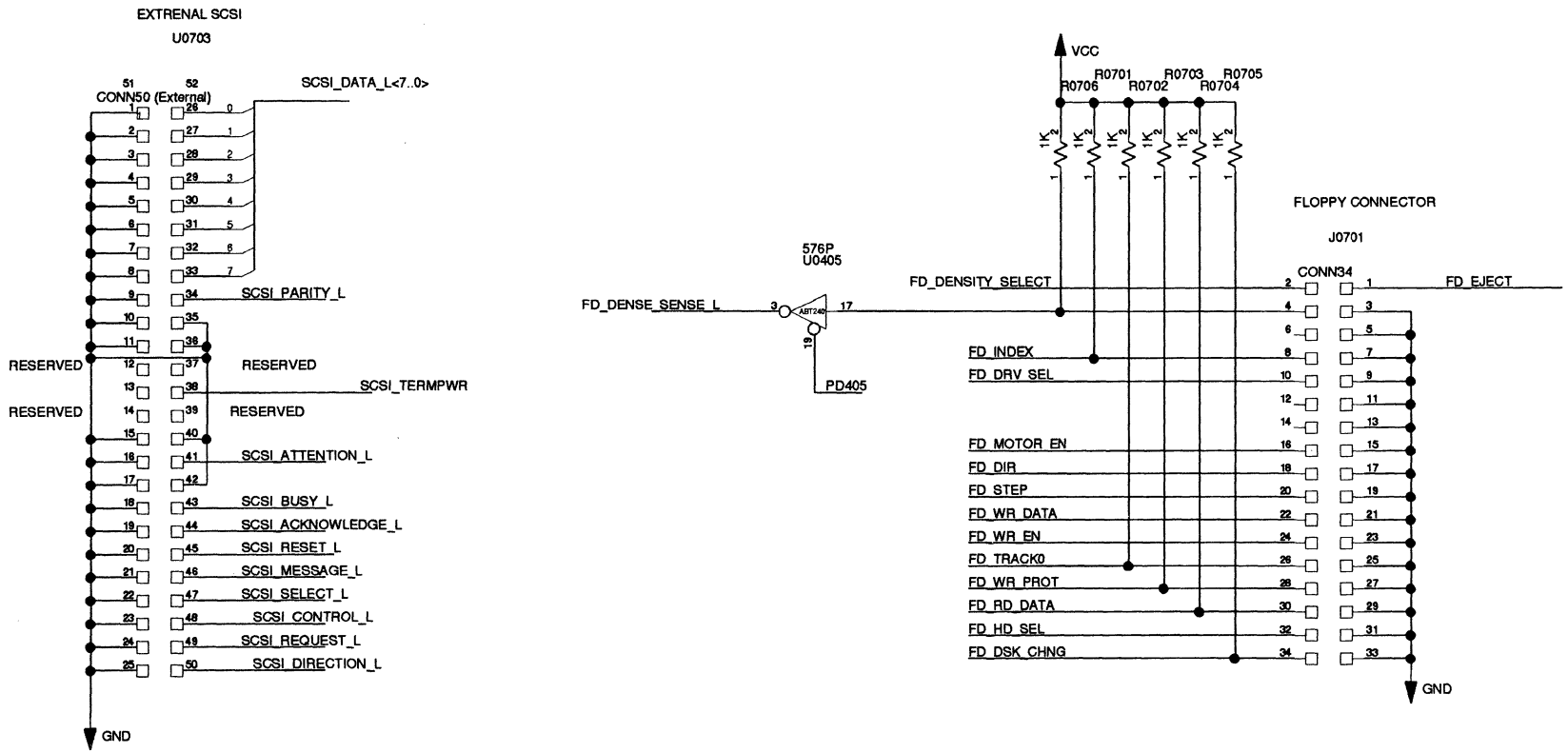
DRAWING

TITLE: IO DEMO BOARD	DRAWING: 6 OF 11	REV: P2
SHEET:	FILE:	
ENGINEER:	DATE: LAST_MODIFIED=Tue Jun 23 11:39:10 1992	

ECO	DESCRIPTION	DATE	APPROVALS

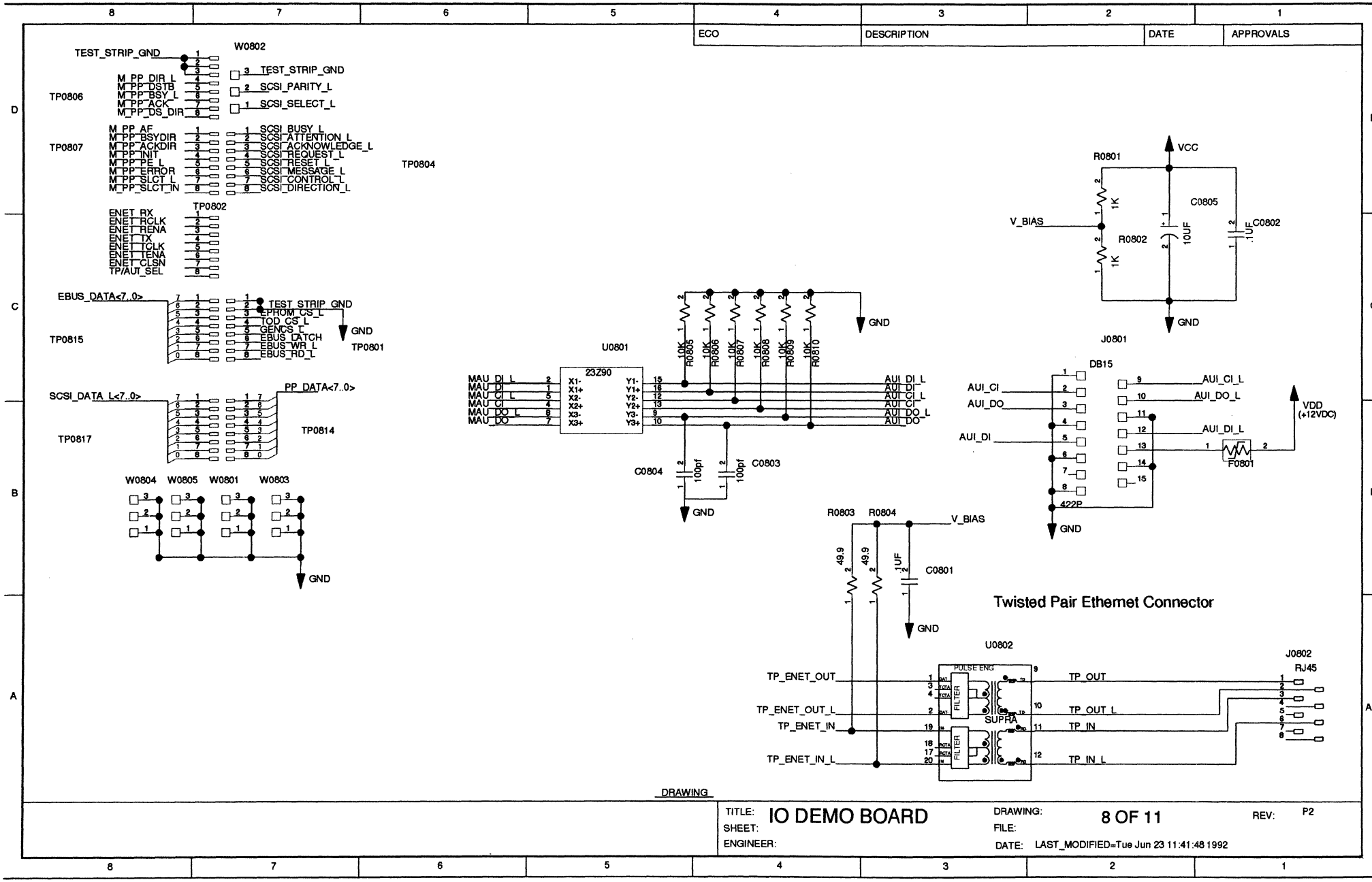
SCSI Connector

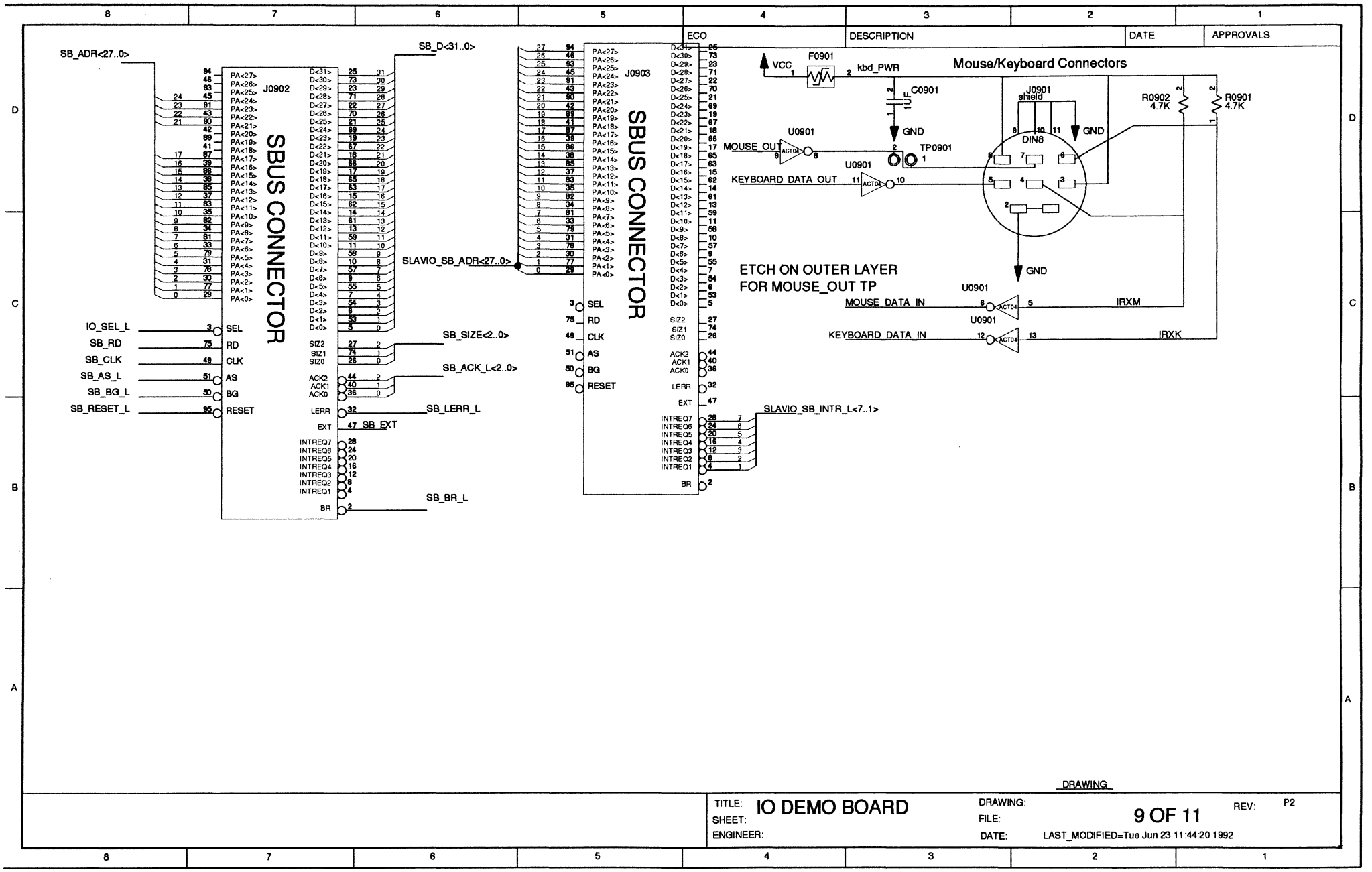
Housing connected to chassis ground.



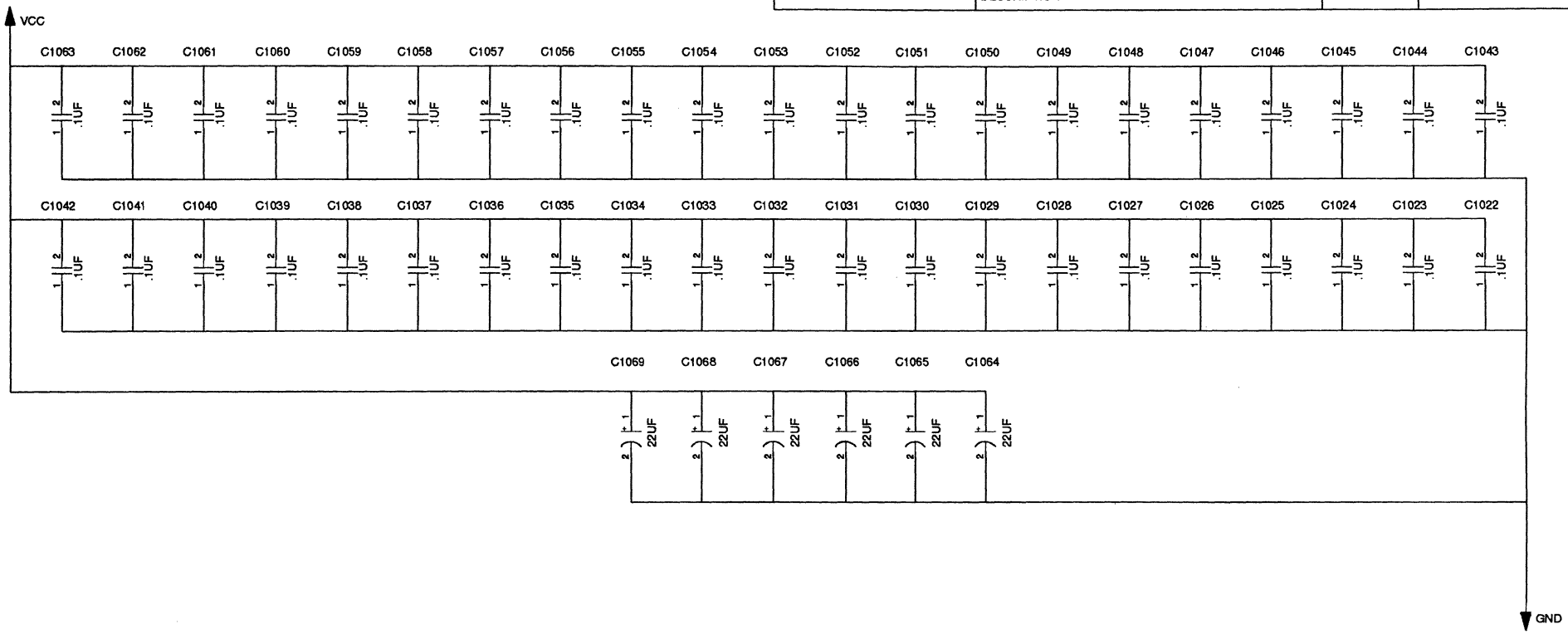
DRAWING

TITLE: IO DEMO BOARD	DRAWING:	7 OF 11	REV: P2
SHEET:	FILE:		
ENGINEER:	DATE: LAST_MODIFIED=Tue Jun 23 11:40:27 1992		


ECO
DESCRIPTION
DATE
APPROVALS
W0802
TEST_STRIP_GND
M_PP_DIR_L
M_PP_DSTB
M_PP_BSY_L
M_PP_ACK
M_PP_DS_DIR
TP0806
TEST_STRIP_GND
SCSI_PARITY_L
SCSI_SELECT_L
M_PP_AF
M_PP_BSYDIR
M_PP_ACKDIR
M_PP_INIT
M_PP_PE_L
M_PP_ERROR
M_PP_SLCT_L
M_PP_SLCT_IN
TP0807
SCSI_BUSY_L
SCSI_ATTENTION_L
SCSI_ACKNOWLEDGE_L
SCSI_REQUEST_L
SCSI_RESET_L
SCSI_MESSAGE_L
SCSI_CONTROL_L
SCSI_DIRECTION_L
TP0804
TP0802
ENET_RX
ENET_RCLK
ENET_RENA
ENET_TX
ENET_TCLK
ENET_TENA
ENET_CLSN
TP/AUT_SEL
EBUS_DATA<7..0>
TEST_STRIP_GND
EPROM_CS_L
TOD_CS_L
GENCS_L
EBUS_LATCH
EBUS_WR_L
EBUS_RD_L
GND
TP0801
TP0815
SCSI_DATA_L<7..0>
PP_DATA<7..0>
TP0814
TP0817
W0804
W0805
W0801
W0803
GND
U0801
23Z90
MAU_DI_L
MAU_DI
MAU_CI_L
MAU_CI
MAU_DO_L
MAU_DO
X1-
X1+
X2-
X2+
X3-
X3+
Y1-
Y1+
Y2-
Y2+
Y3-
Y3+
AUI_DI_L
AUI_DI
AUI_CI_L
AUI_CI
AUI_DO_L
AUI_DO
R0805
R0806
R0807
R0808
R0809
R0810
10K
GND
C0804
C0803
100pf
GND
R0801
R0802
1K
1K
VCC
V_BIAS
C0805
10UF
C0802
.1UF
GND
J0801
DB15
AUI_CI
AUI_DO
AUI_DI
AUI_CI_L
AUI_DO_L
AUI_DI_L
VDD
(+12VDC)
F0801
422P
GND
R0803
R0804
V_BIAS
49.9
49.9
C0801
.1UF
GND
Twisted Pair Ethernet Connector
U0802
PULSE ENG
SUPHA
FILTER
FILTER
TP_ENET_OUT
TP_ENET_OUT_L
TP_ENET_IN
TP_ENET_IN_L
TP_OUT
TP_OUT_L
TP_IN
TP_IN_L
J0802
RJ45
DRAWING
TITLE: IO DEMO BOARD
SHEET:
ENGINEER:
DRAWING: 8 OF 11
FILE:
DATE: LAST_MODIFIED=Tue Jun 23 11:41:48 1992
REV: P2


ECO
DESCRIPTION
DATE
APPROVALS
SB_ADR<27..0>
SB_D<31..0>
SLAVIO_SB_ADR<27..0>
SBUS CONNECTOR
J0902
J0903
IO_SEL_L
SB_RD
SB_CLK
SB_AS_L
SB_BG_L
SB_RESET_L
SB_SIZE<2..0>
SB_ACK_L<2..0>
SB_LERR_L
SB_EXT
SB_BR_L
SLAVIO_SB_INTR_L<7..1>
Mouse/Keyboard Connectors
VCC
F0901
kbd_PWR
C0901
1UF
J0901
shield
R0902
4.7K
R0901
4.7K
GND
DIN8
U0901
TP0901
MOUSE_OUT
KEYBOARD_DATA_OUT
ETCH ON OUTER LAYER
FOR MOUSE_OUT TP
MOUSE_DATA_IN
KEYBOARD_DATA_IN
IRXM
IRXK
DRAWING
TITLE: IO DEMO BOARD
SHEET:
ENGINEER:
DRAWING:
FILE:
9 OF 11
REV: P2
DATE: LAST_MODIFIED=Tue Jun 23 11:44:20 1992

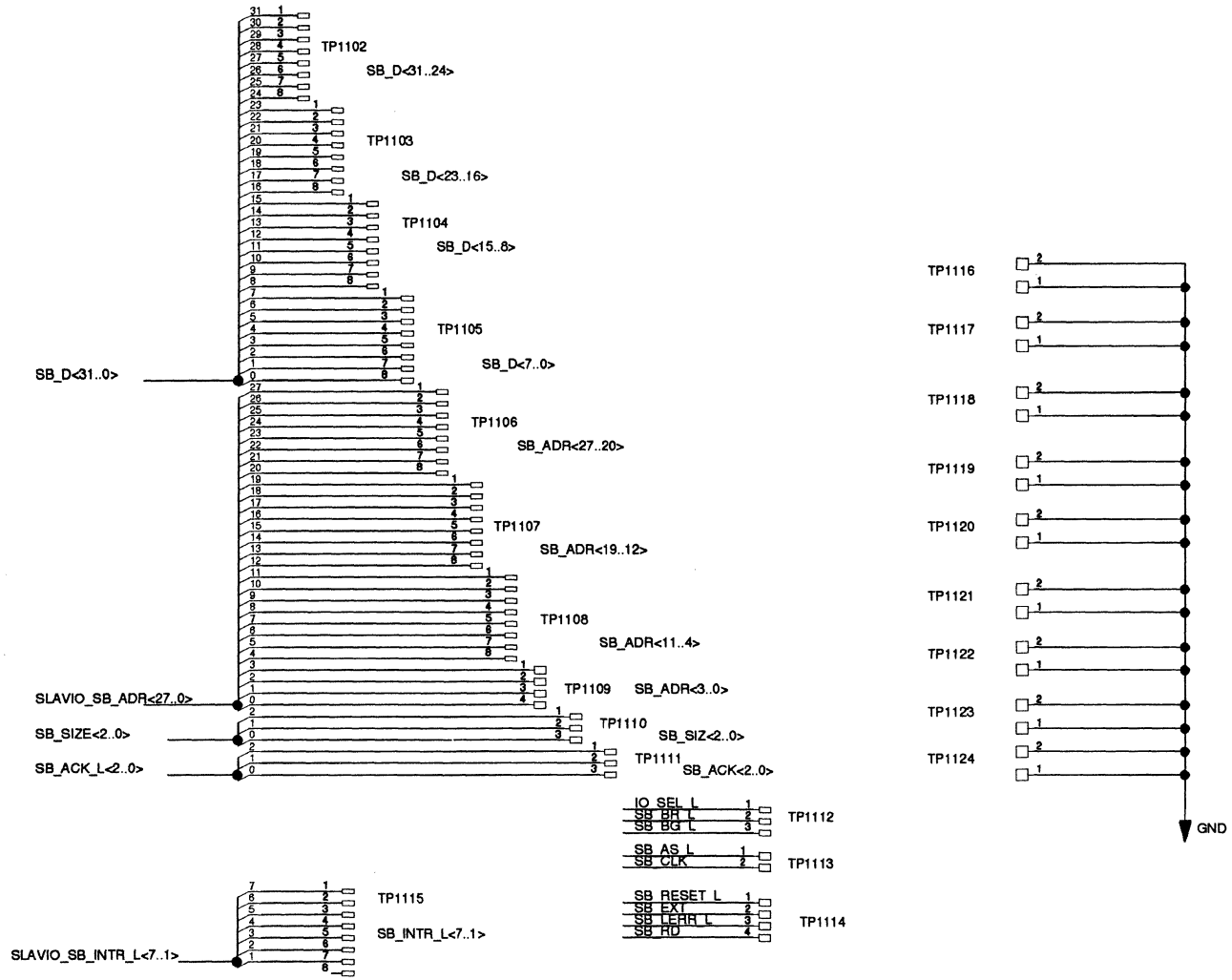
ECO	DESCRIPTION	DATE	APPROVALS



DRAWING

TITLE: IO DEMO BOARD	DRAWING:	REV: P2
SHEET:	FILE: 10 OF 11	
ENGINEER:	DATE: LAST_MODIFIED=Tue Jun 23 11:46:56 1992	

ECO	DESCRIPTION	DATE	APPROVALS



TITLE:	IO DEMO BOARD	DRAWING:	REV: P2
SHEET:	11 OF 11	FILE:	
ENGINEER:		DATE:	

Revision History

Revision	*Date*	*Comments*
First Draft	February 13, 1992	Incorporated redlines from Jim Dexter, Ron Lee, Raymond Ho, Binh Pham.
Second Draft	August 13, 1992	Incorporated redlines from Ray Ho, Binh Pham, Jim Dexter.
Third Draft	October 19, 1992	Incorporated redlines from Mike Cavanaugh, Ron Lee, Stefen Boyd, Brian Healy, Chau Nguyen, Binh Pham, Ben Toy, Jim Dimino, Ken Kochi, Ron Rose, Harvey Siegel, Raymond Ho, Francisco De Anda, Scott Hickman, Elaine Miller, and Jim Dexter.
Forth Draft	November 23, 1992	Peter Palm, Geoff George, Binh Pham, Loren Chan, Raymond Tam, Ken Kochi, Kaamel Kermaani.
FCS	May 2, 1993	Jim Dexter, Binh Pham, Raymond Ho, Mike Cavanaugh, Chau Nguyen, Loren Chan, Mark Elgood, Peter Palm, Brian Healy.