Products Rights Notice:

You understand that these materials were not prepared for public release and you assume all risks in using these materials. These risks include, but are not limited to errors, inaccuracies, incompleteness and the possibility that these materials infringe or misappropriate the intellectual property right of others. You agree to assume all such risks.

THESE MATERIALS ARE PROVIDED BY THE COPYRIGHT HOLDERS AND OTHER CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS (INCLUDING ANY OF OWNER'S PARTNERS, VENDORS AND LICENSORS) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THESE MATERIALS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Sun, Sun Microsystems, the Sun logo, Solaris, OpenSPARC T1, OpenSPARC T2 and UltraSPARC are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd. The Adobe logo is a registered trademark of Adobe Systems, Incorporated. Part of the products covered by these materials may be derived from the Berkeley BSD systems licensed by the University of California. Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product described in these materials. This distribution may include materials developed by third parties who have intellectual property rights therein. Products covered by and information contained in these materials may be controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists may be prohibited.

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

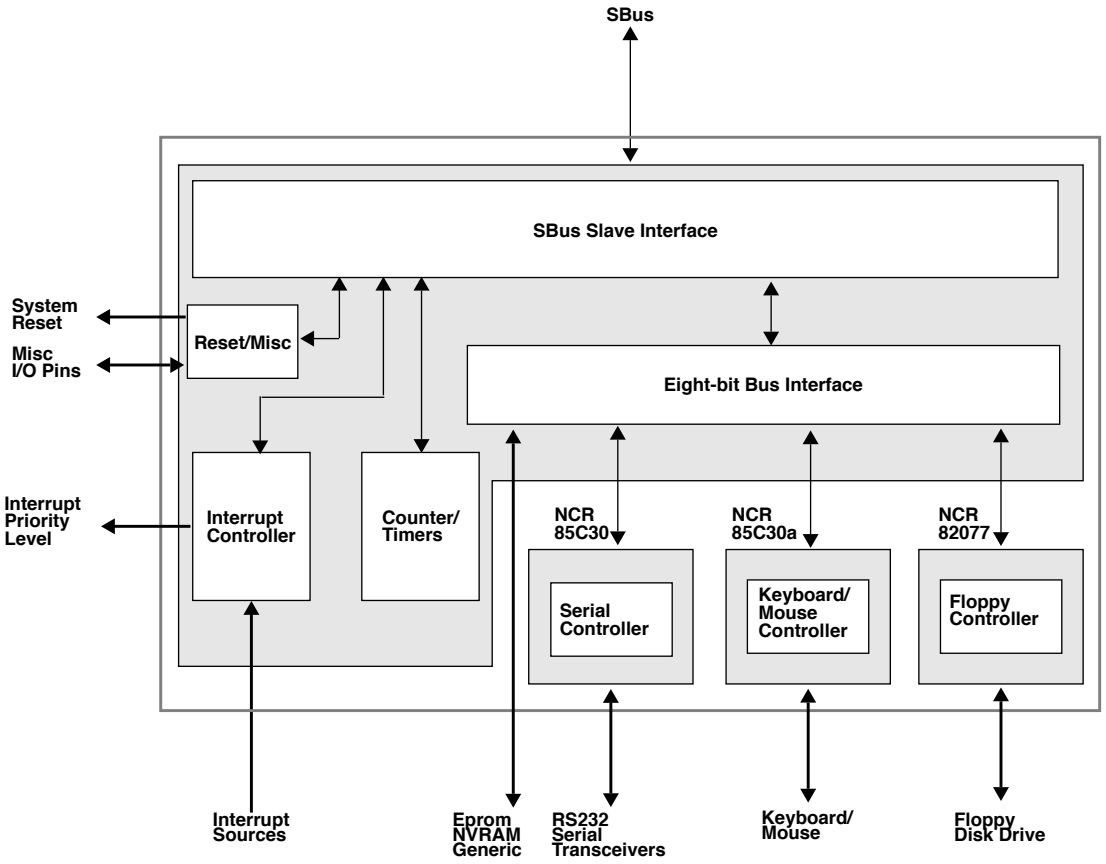

Figure 6-1 89C105 Block Diagram

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

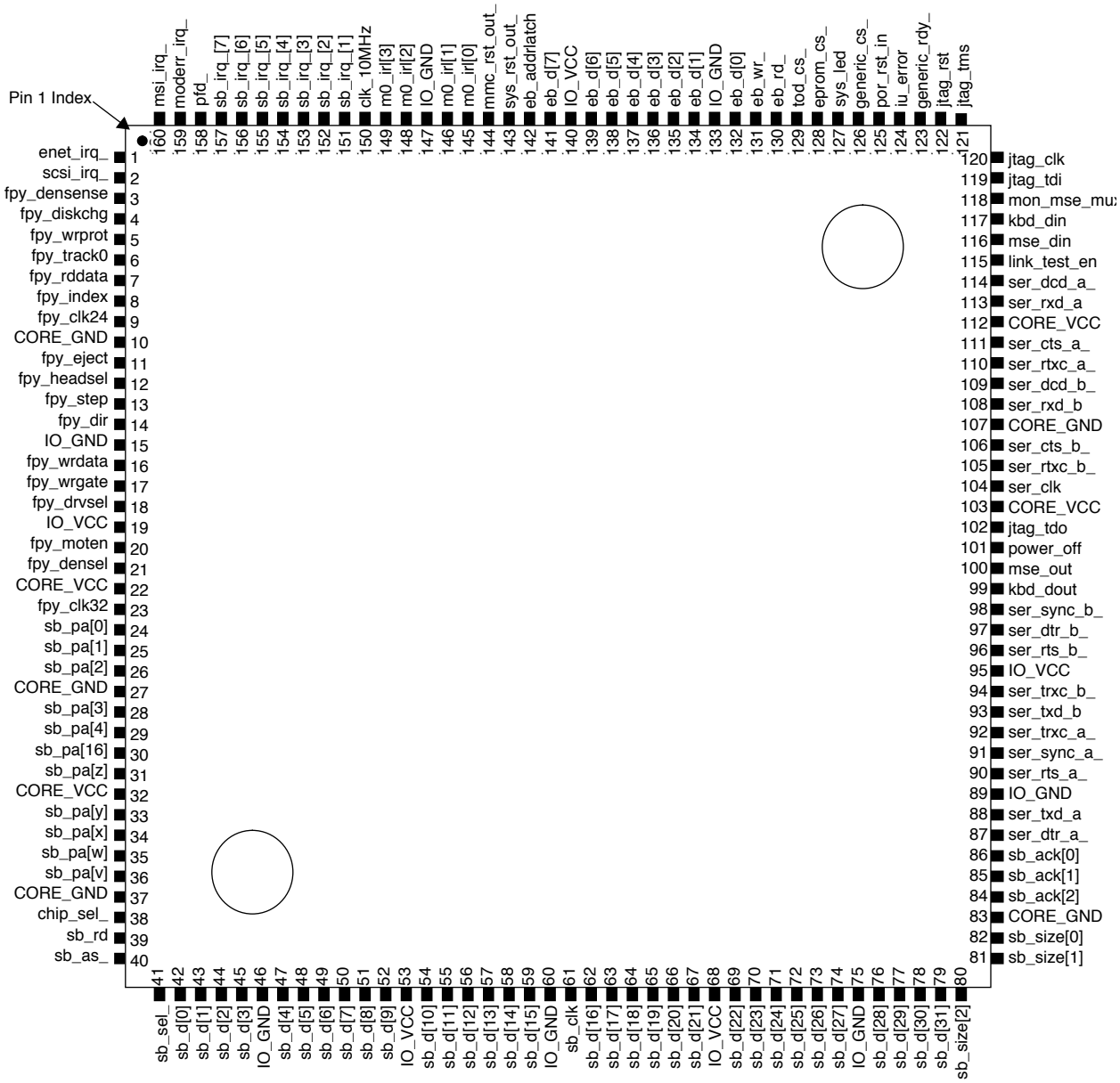

Figure 6-2 Pinout Map

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

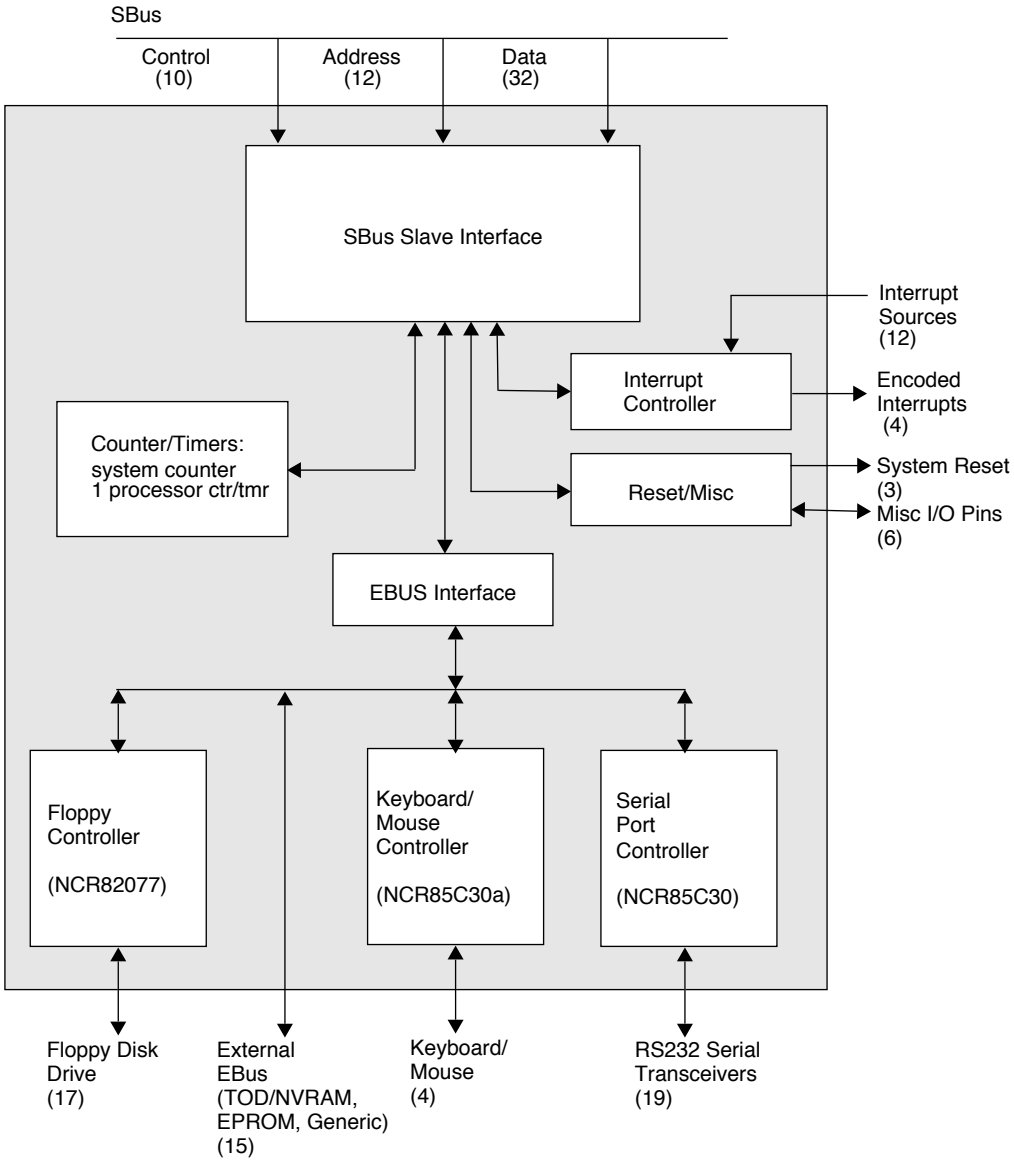

Figure 6-3 Chip block diagram

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 100 0000 100 0002 100 0004 100 0006 110 0000 110 0002 110 0004 110 0006	**Keyboard, Mouse, and Serial Ports** Mouse Control Port Mouse Data Port Keyboard Control Port Keyboard Data Port TTYB Control Port TTYB Data Port TTYA Control Port TTYA Data Port	B
120 0000 -> 12F FFFF	**TOD/NVRAM**	B,H,W
130 0000 -> 13F FFFF	**General Purpose (Generic Port)**	B
140 0000 -> 14F FFFF 140 0002 140 0004 140 0004 140 0005 140 0006 140 0007 140 0007	**Floppy Controller** Digital Output Register (DOR) Main Status Register (MSR, Read Only) Datarate Select Register (DSR, Write Only) FIFO Reserved (Test mode select) Digital Input Register (DIR, Read Only) Configuration Control Register (CCR, Write Only)	B
150 0000 -> 170 0000	**Reserved**	
180 0000	**89C105 Configuration Register**	B
190 0000 -> 19F FFFF 190 0000 191 0000	**Auxiliary I/O Registers** Aux 1 Register (Miscellaneous System Functions) Aux 2 Register (Software Powerdown Control)	B

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 theEPROM 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 (.95µ 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.

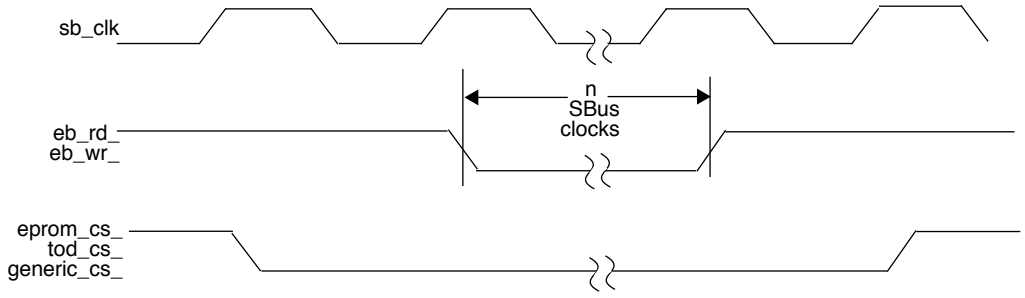

Figure 6-4 EBus Timings: Chip Selects and eb_rd_/eb_wr

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 100 0000 100 0002 100 0004 100 0006 110 0000 110 0002 110 0004 110 0006	**Keyboard, Mouse, and Serial Ports** Mouse Control Port Mouse Data Port Keyboard Control Port Keyboard Data Port TTYB Control Port TTYB Data Port TTYA Control Port TTYA Data Port	B
120 0000 -> 12F FFFF	**TOD/NVRAM**	B,H,W
130 0000 -> 13F FFFF	**General Purpose (Generic port)**	B
140 0000 -> 14F FFFF 140 0002 140 0004 140 0004 140 0005 140 0006 140 0007 140 0007	**Floppy Controller** Digital Output Register (DOR) Main Status Register (MSR, Read Only) Datarate Select Register (DSR, Write Only) FIFO Reserved (Test mode select) Digital Input Register (DIR, Read Only) Configuration Control Register (CCR, Write Only)	B

Devices Residing on the EBus

The following devices reside on theEBus:

- 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 theEBus 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 externalEBus: 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 theEPROM 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 theEBUS 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.

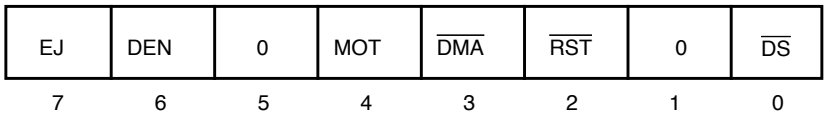

Figure 6-5 DOR Register Field Definitions

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)

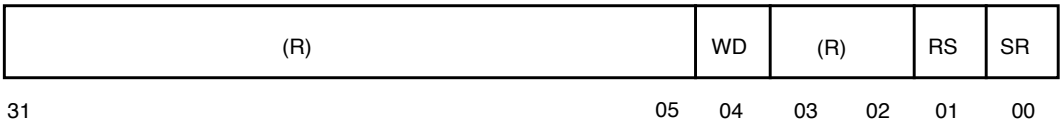

Figure 6-6 System Reset (Control) Register

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

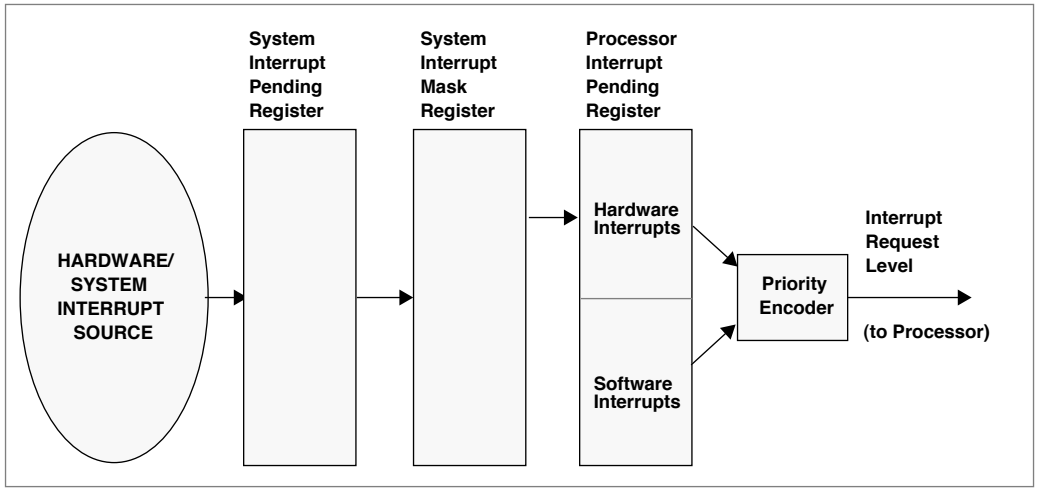

Figure 6-7 Interrupt Control Block Diagram

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.

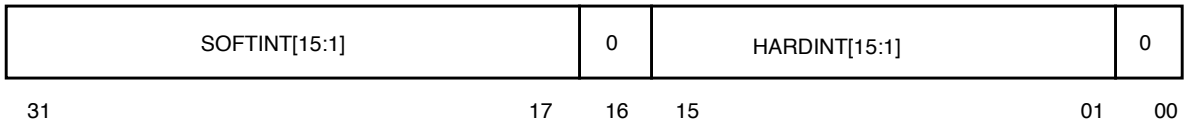

Figure 6-8 Processor Interrupt Pending Register (read only)

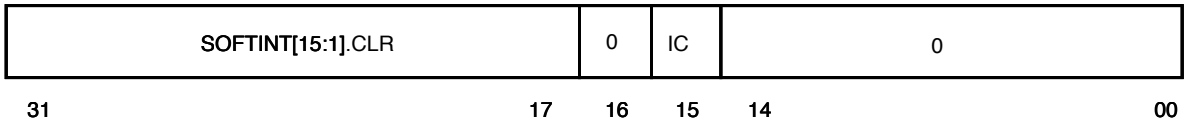

Figure 6-9 Processor Interrupt Clear-Pending Pseudoregister

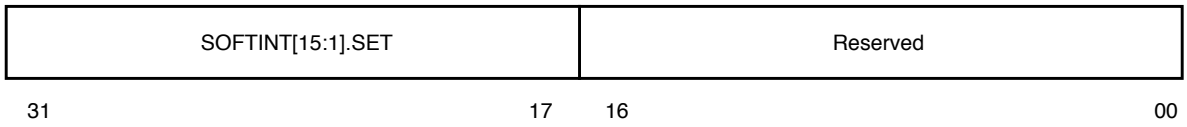

Figure 6-10 Processor Set-Soft-Int Pseudoregister

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.

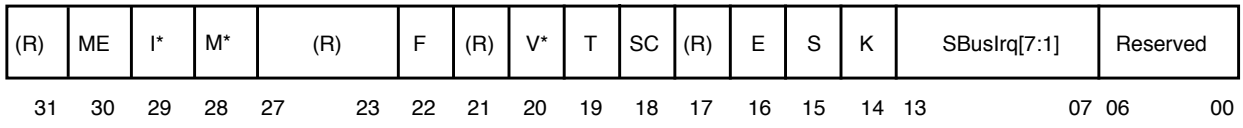

Figure 6-11 System Interrupt Pending Register

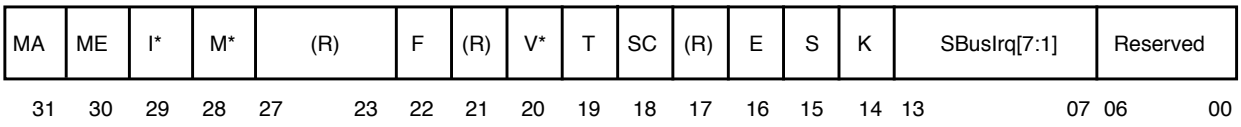

Figure 6-12 System Interrupt Target Mask Register (Read Only), and Mask Set and Mask Clear Registers (Write Only)

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

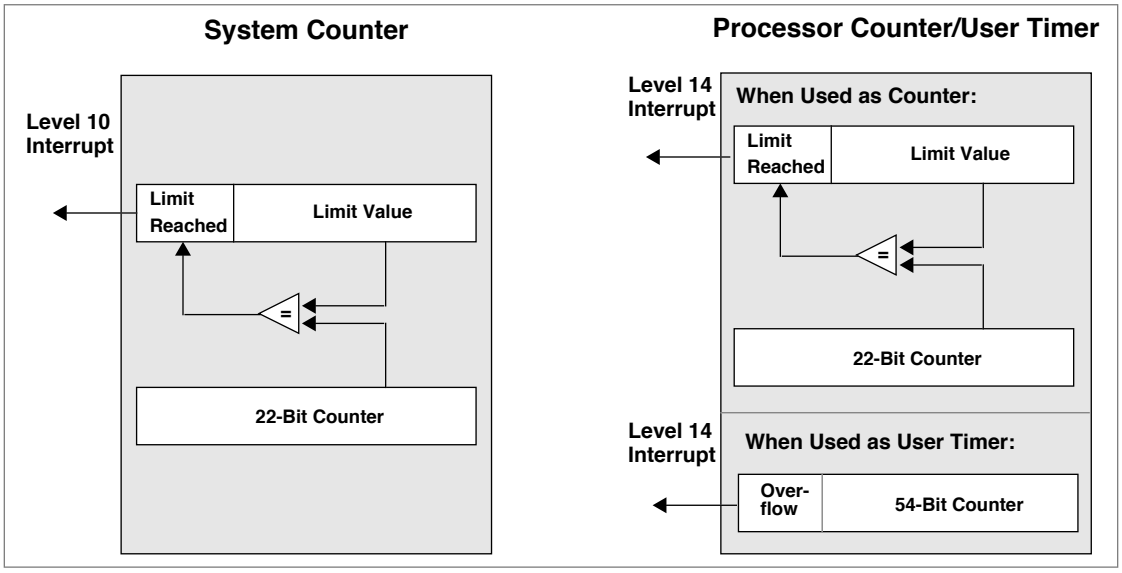

Figure 6-13 Counter-Timer Block Diagram

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:

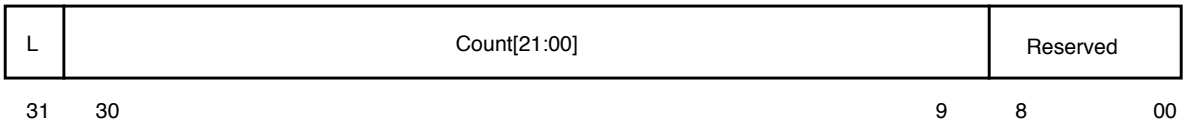

Figure 6-14 System Counter Register Field Definitions

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.

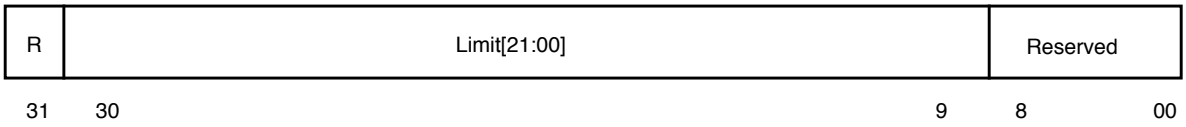

Figure 6-15 Counter/Timer Limit Register Field Definitions

Field Definitions:

R, Reserved	Reserved, read as 0.
Limit	Limit value to count to before setting interrupt and resetting counter.

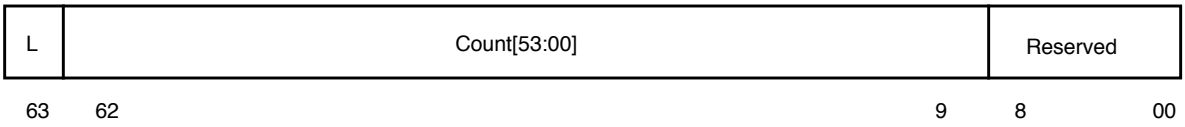

Figure 6-16 User Timer Count Register Field Definitions

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.

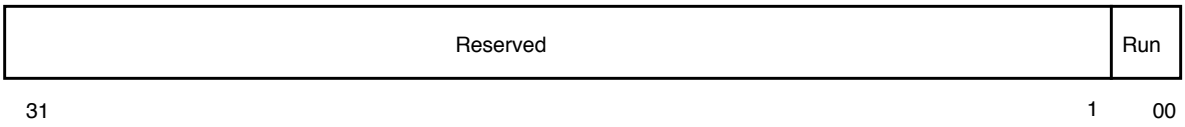

Figure 6-17 User Timer Start/Stop Register Field Definitions

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.

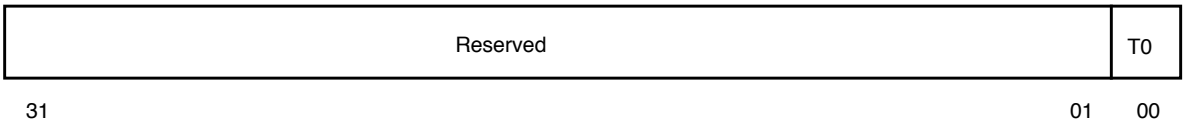

Figure 6-18 Counter/Timer Configuration Register Field Definitions

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.

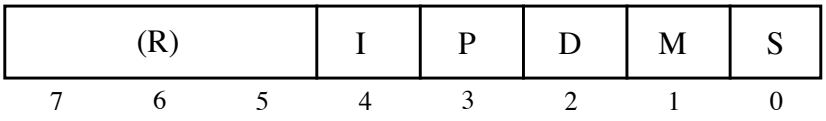

Figure 6-19 The 89C105 Configuration Register Field Definitions

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

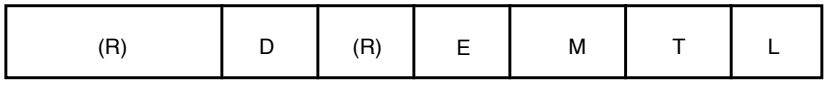

Figure 6-21 Auxiliary I/O Register 1 (0x190 0000) Field Definitions

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

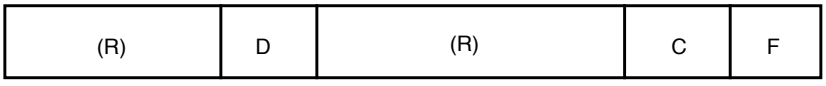

Figure 6-22 Auxiliary I/O Register 2 Field Definitions

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.

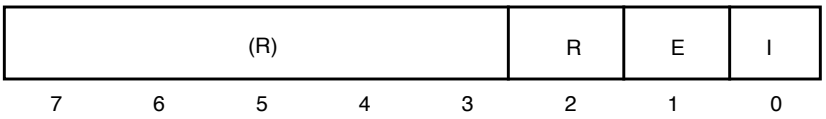

Figure 6-23 Modem Register Field Definitions

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 EEE 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 TRSTBJTAG 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.

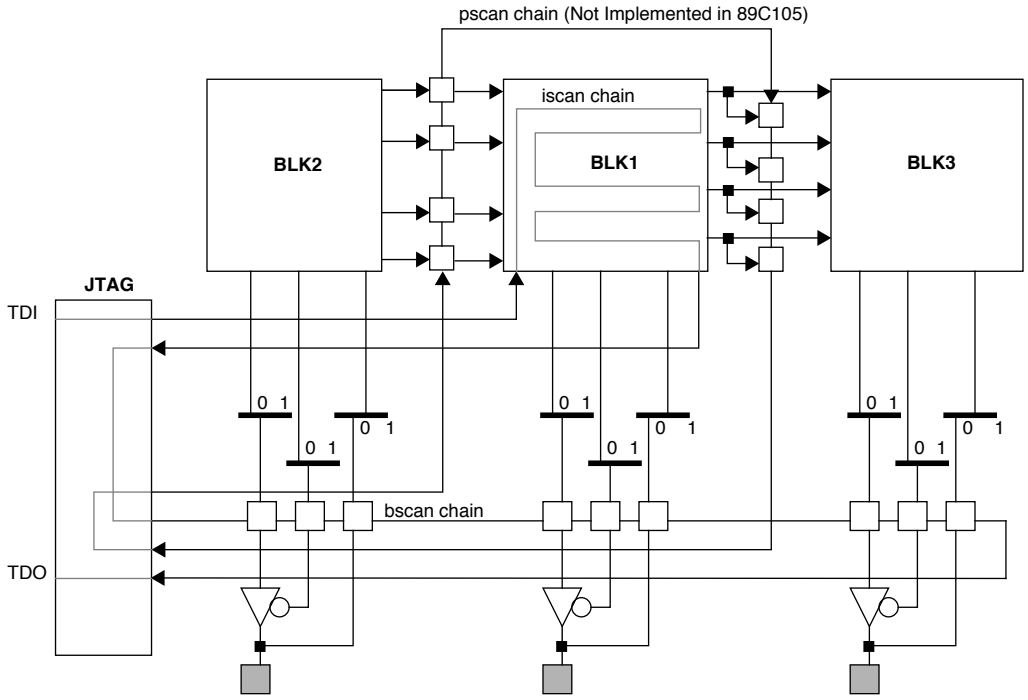

Figure 6-24 TBLK1 (Internal Scan) Diagnostic Mode

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

.

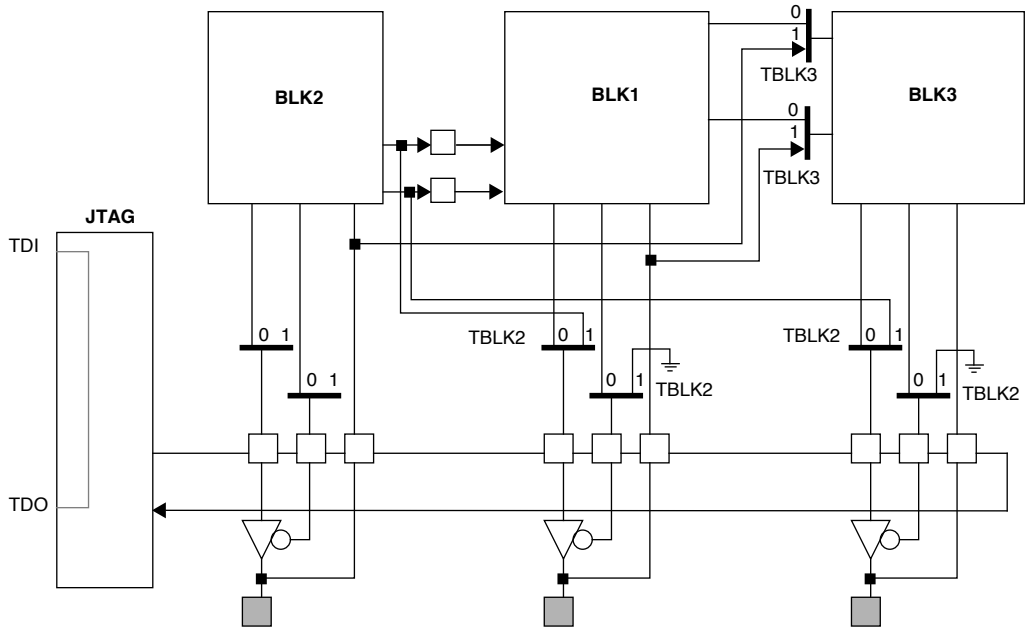

Figure 6-25 TBLK2/TBLK3 Diagnostic Mode

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 theJTAG controller. It has been partitioned into two main functional areas: Scan Datapath and Scan Control Logic:

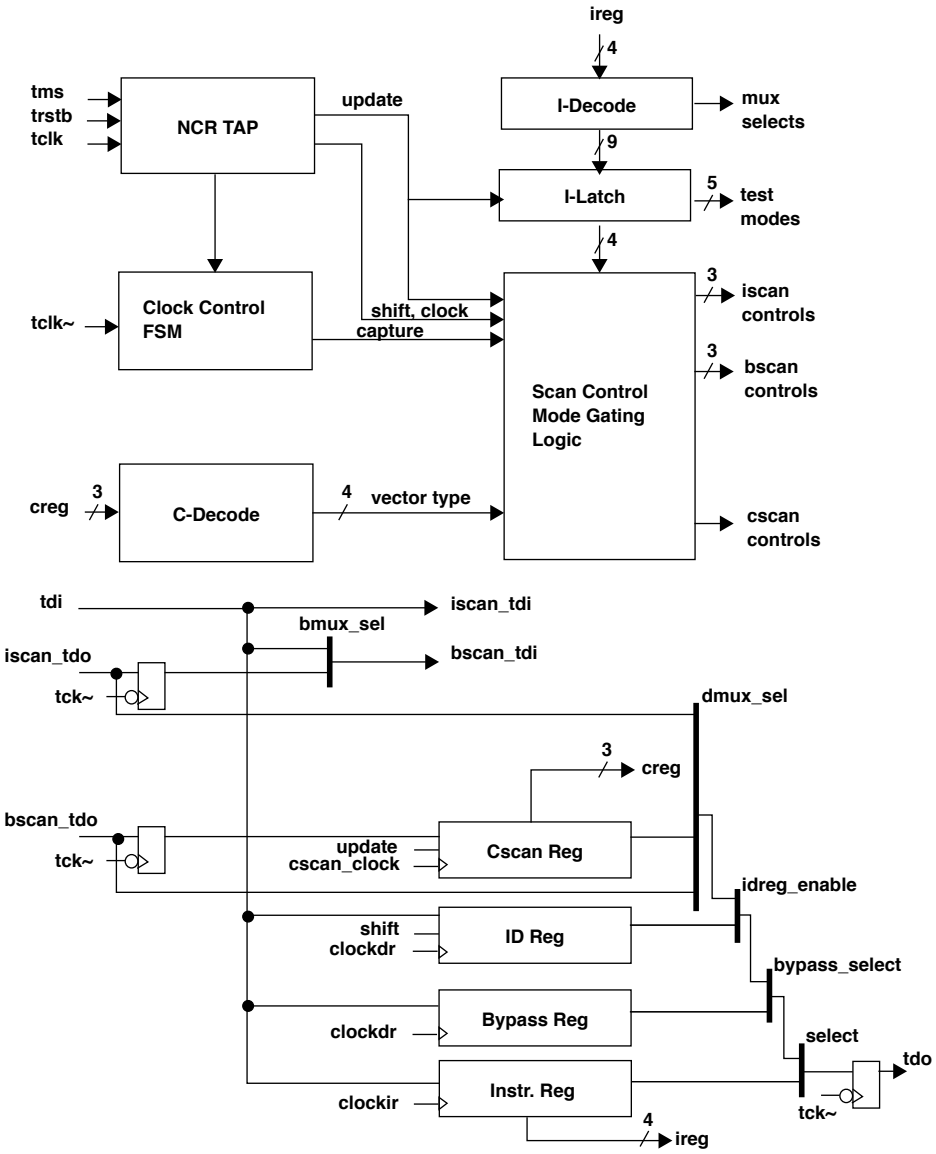

Figure 6-26 JTAG Controller Block Diagram

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

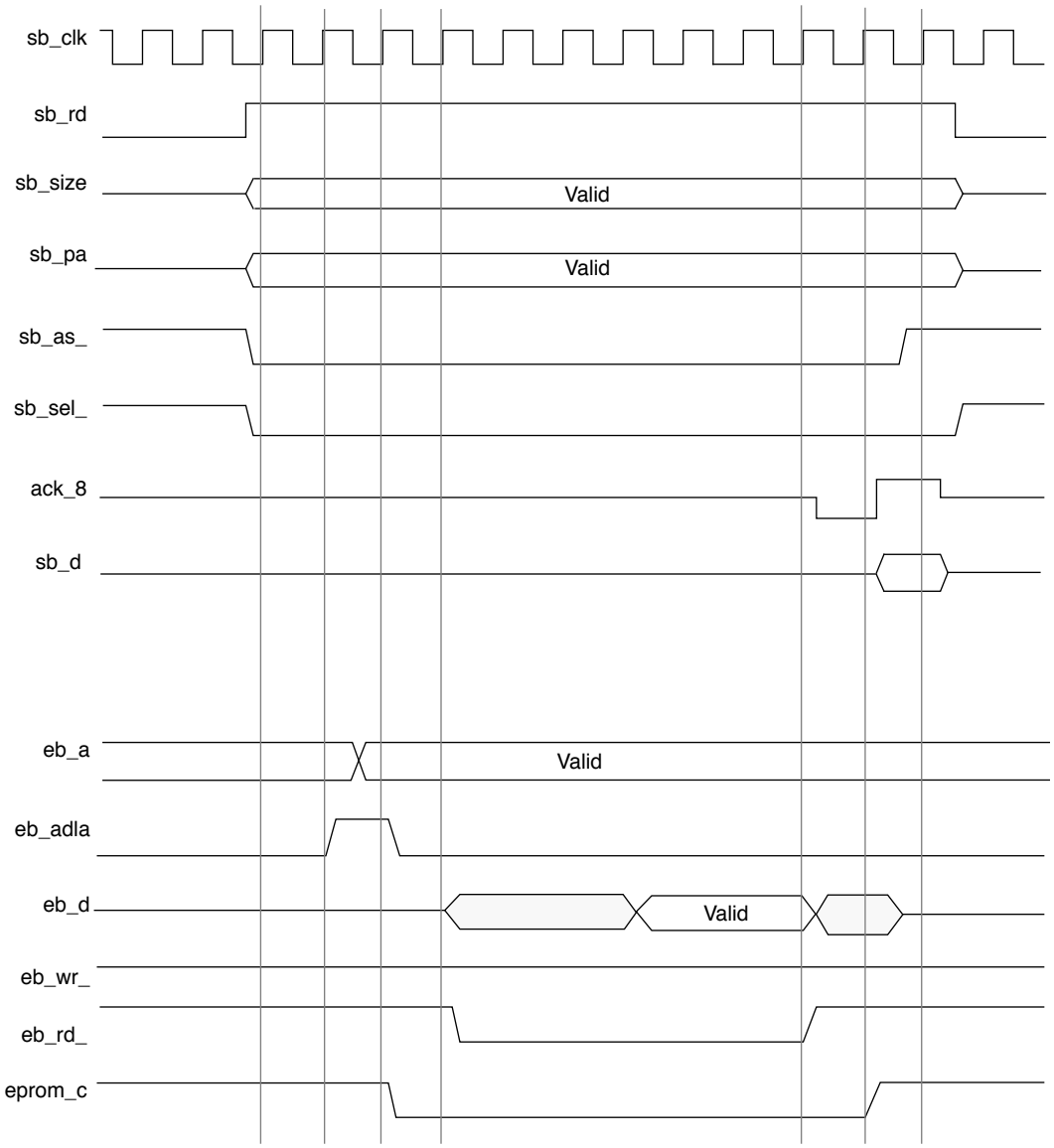

Figure 6-27 EPROM Read

TOD/NVRAM Read

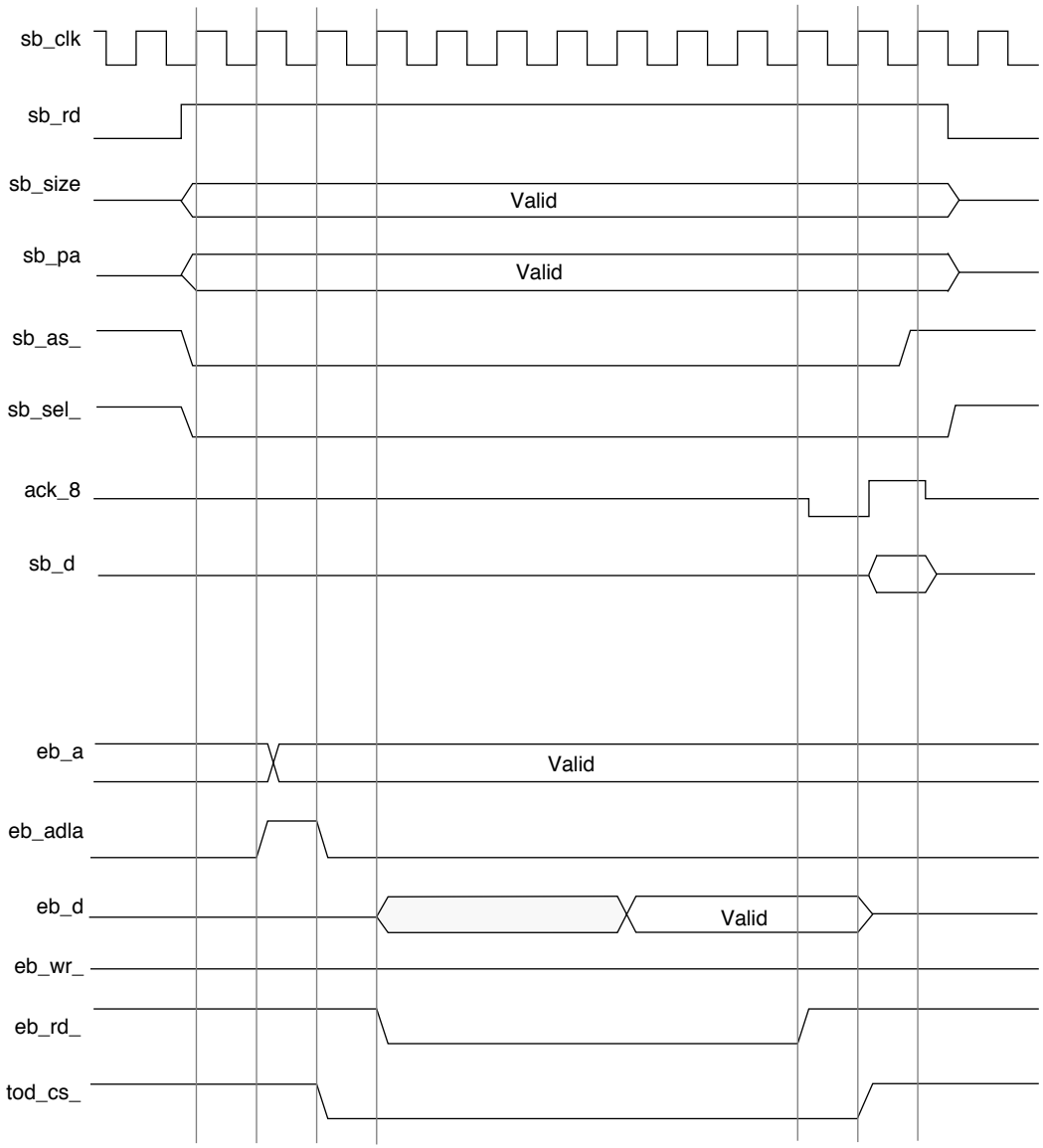

Figure 6-28 TOD/NVRAM Read

TOD/NVRAM Write, Showing Buffered EBus Cycle

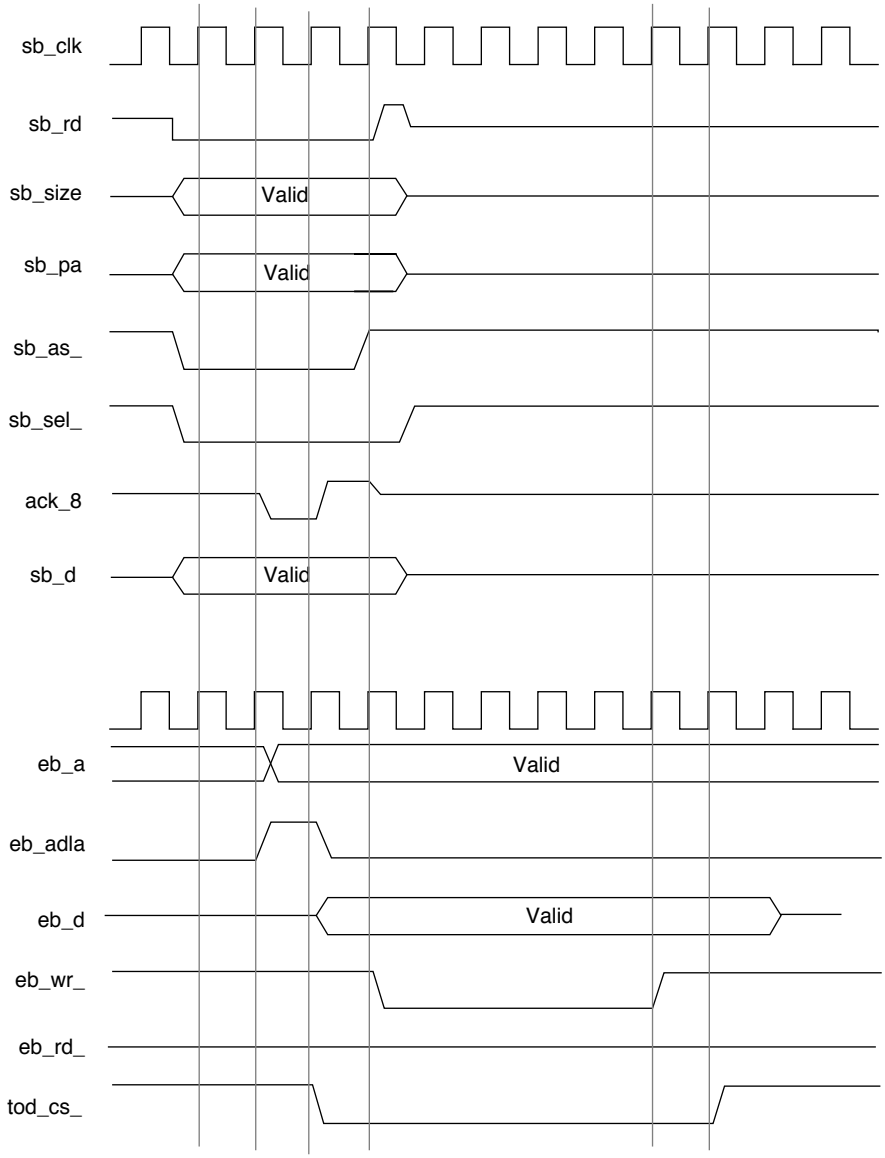

Figure 6-29 TOD/NVRAM Write, Showing Buffered EBus Cycle

TOD/NVRAM Write/Write, Showing Buffered EBus Cycle

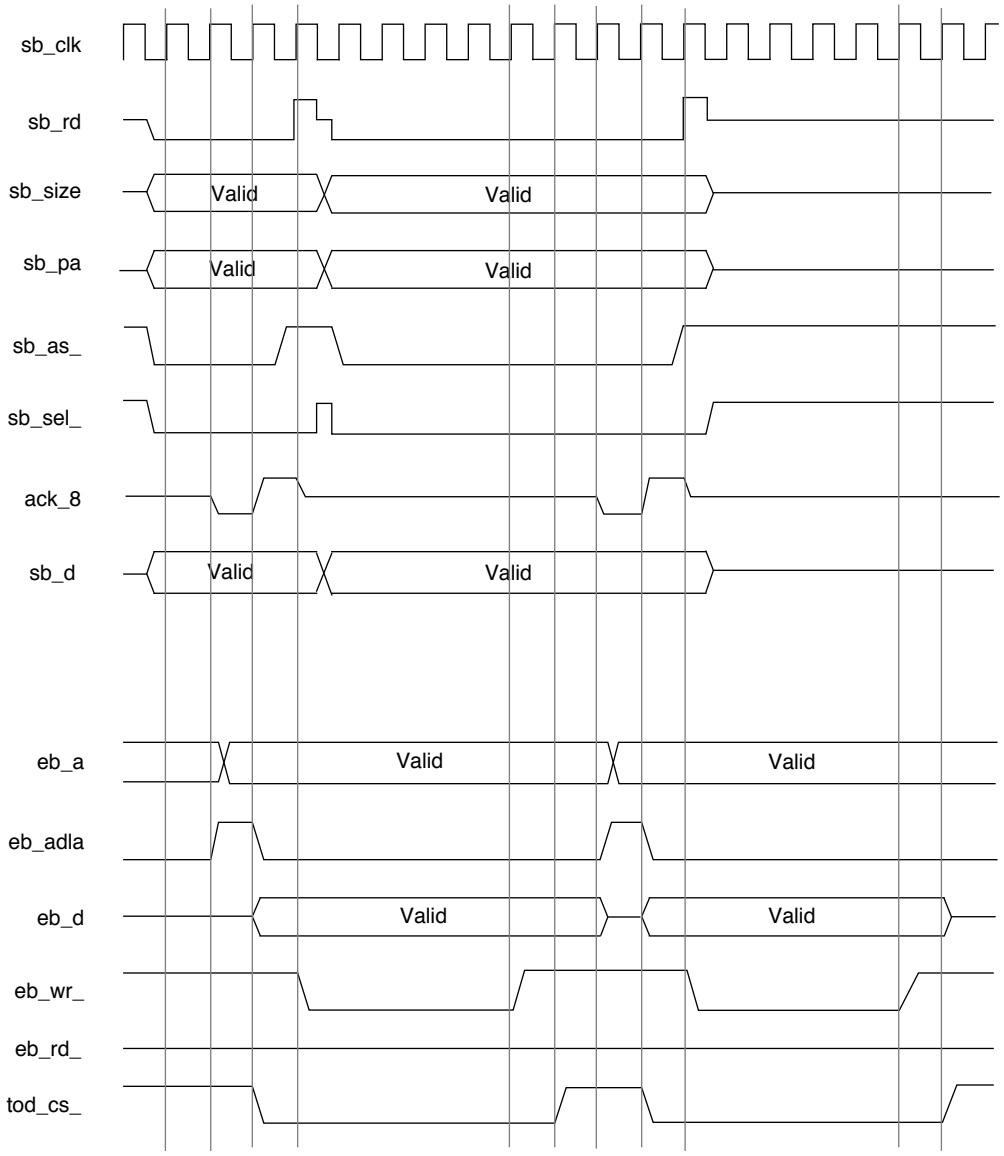

Figure 6-30 TOD/NVRAM Write/Write, Showing Buffered EBus Cycle

Generic Port Read

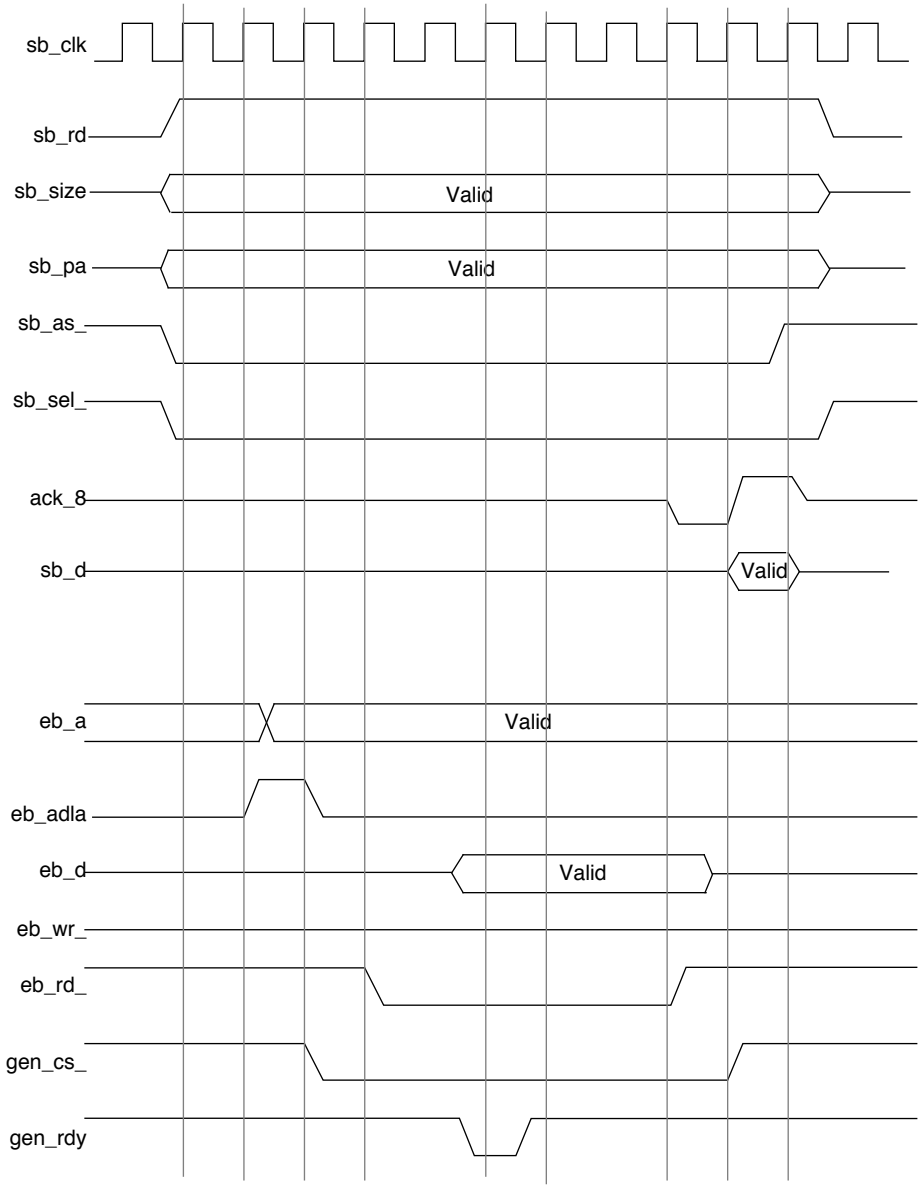

Figure 6-31 Generic Port Read

Generic Port Write

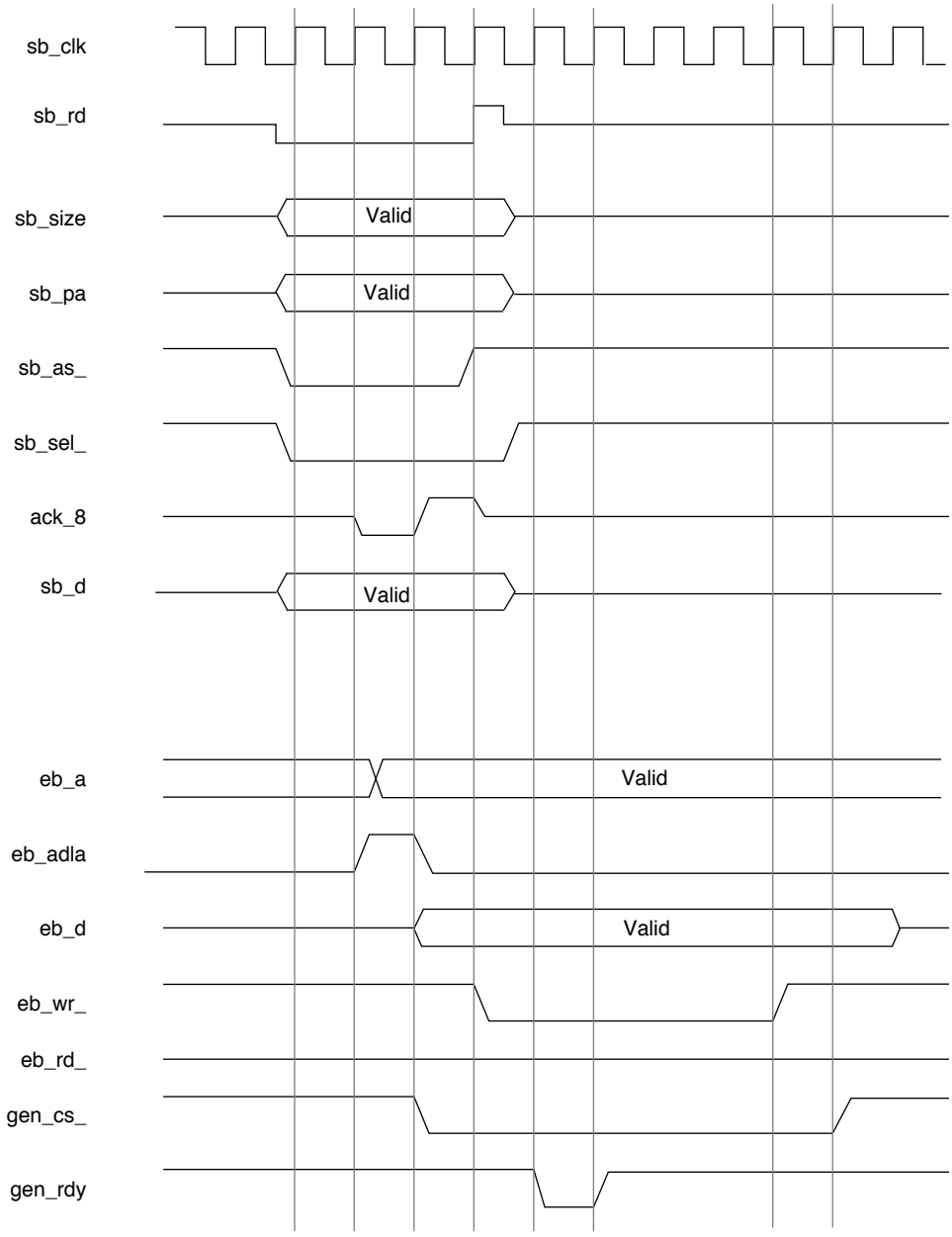

Figure 6-32 Generic Port Write

Generic Port Time-Out

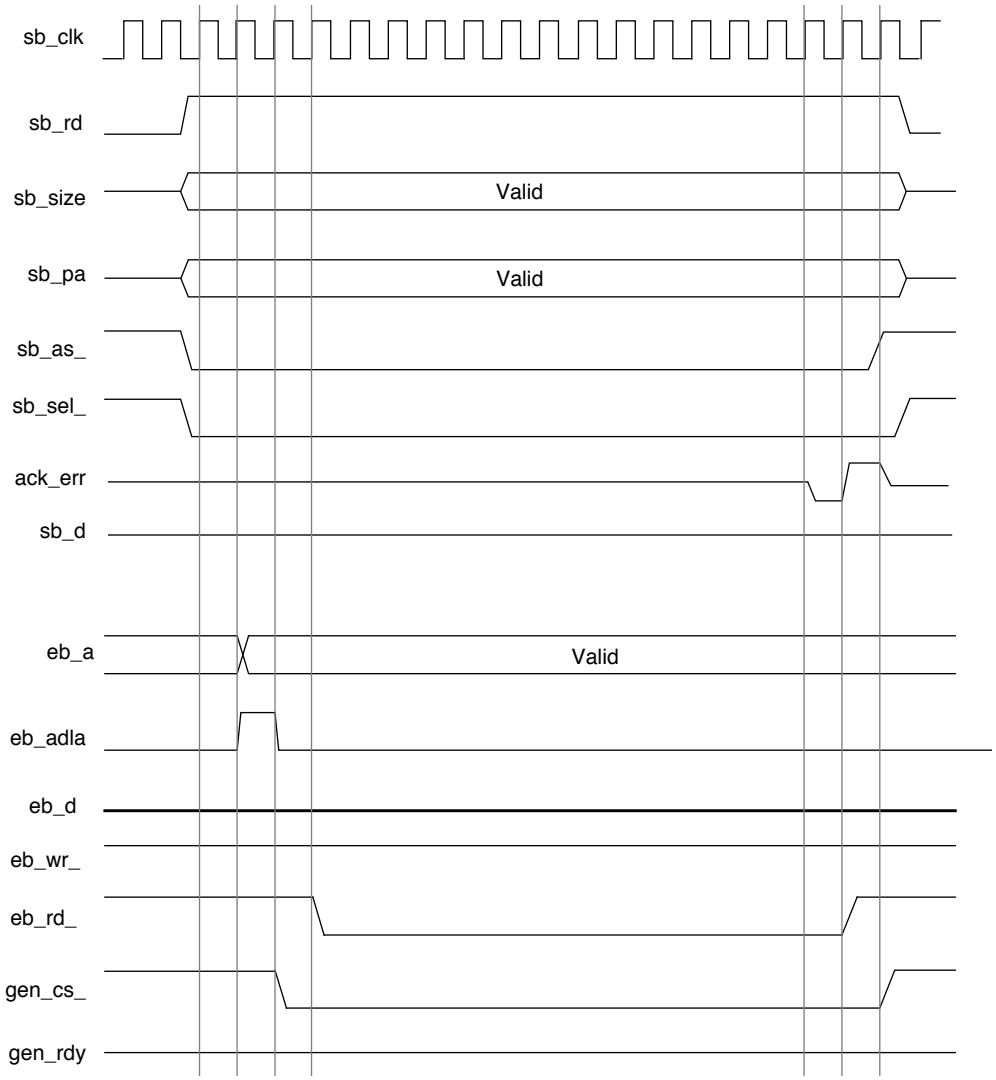

Figure 6-33 Generic Port Time-Out

Serial Ports/Keyboard/Mouse Write/Read Showing Hardware Holdoff

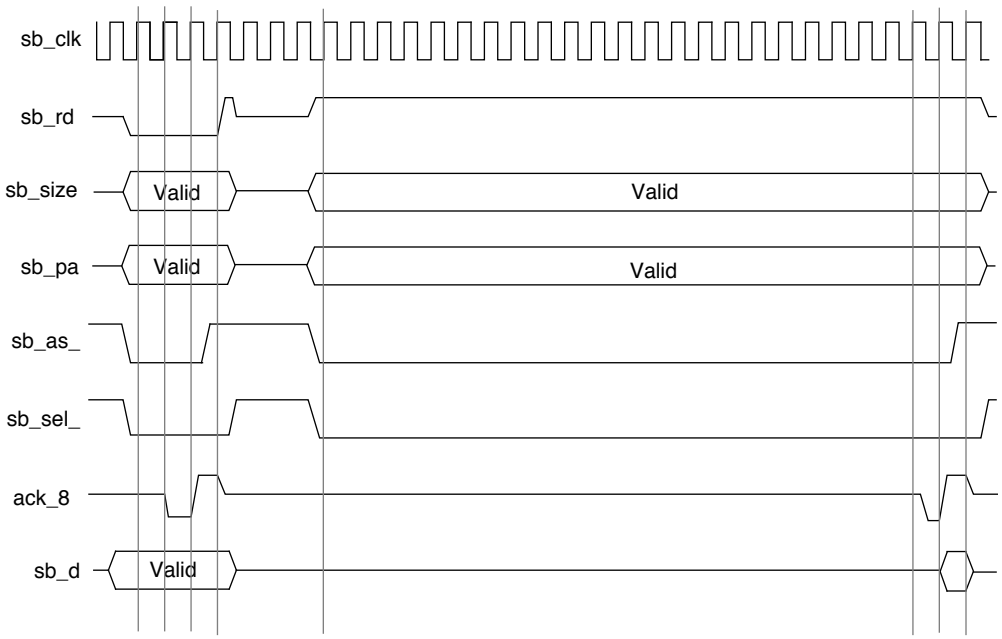

Figure 6-34 Serial Ports/Keyboard/Mouse Write/Read, Showing Hardware Holdoff

Floppy Read/Write

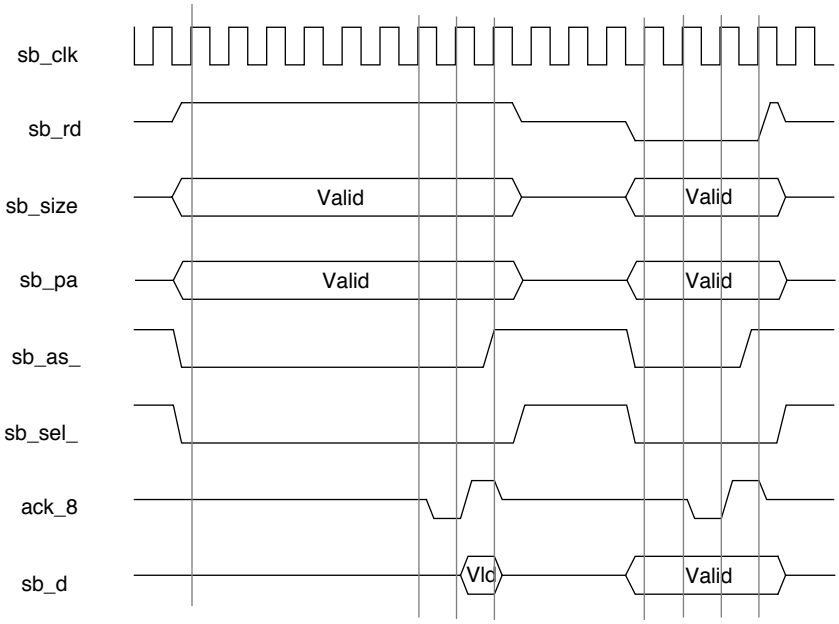

Figure 6-35 Floppy Read/Write

DMR/AUXIO/Configuration Register Read/Write

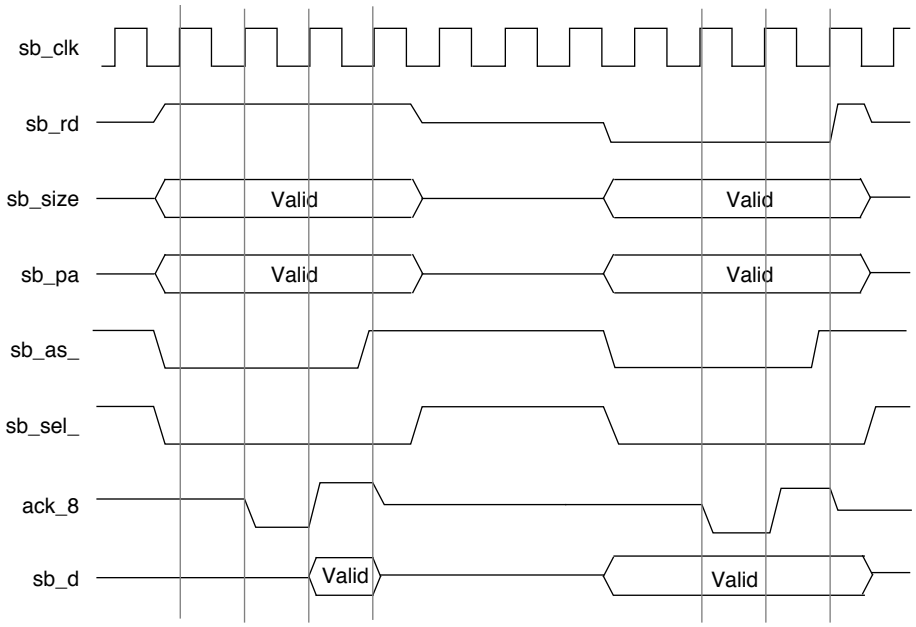

Figure 6-36 DMR/Aux IO/Configuration Register Read/Write

Counter/Timer Read

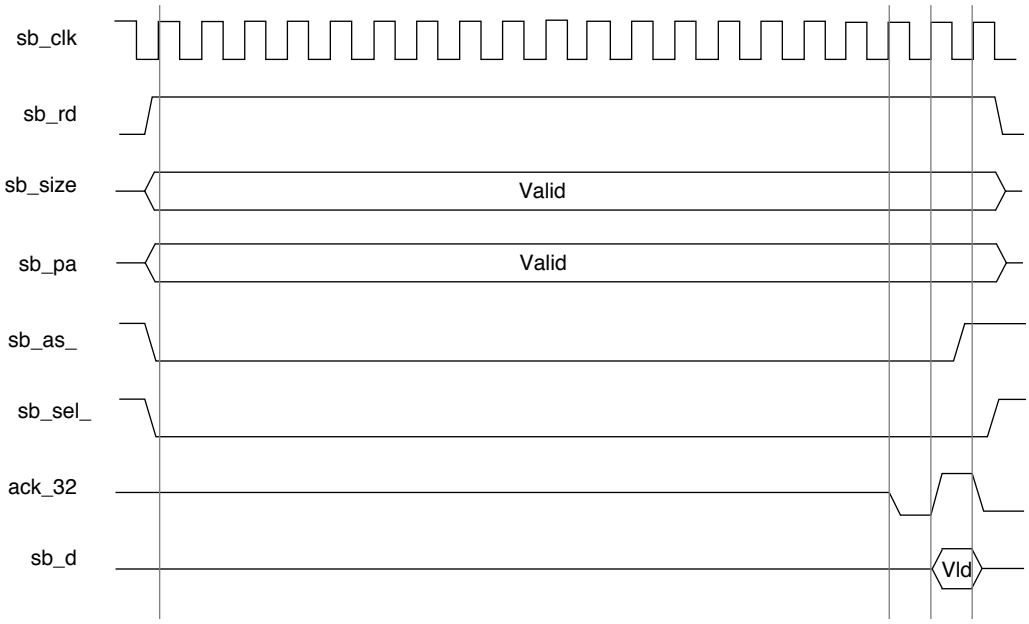

Figure 6-37 Counter/Timer Read

Counter/Timer Two-Word Burst Read

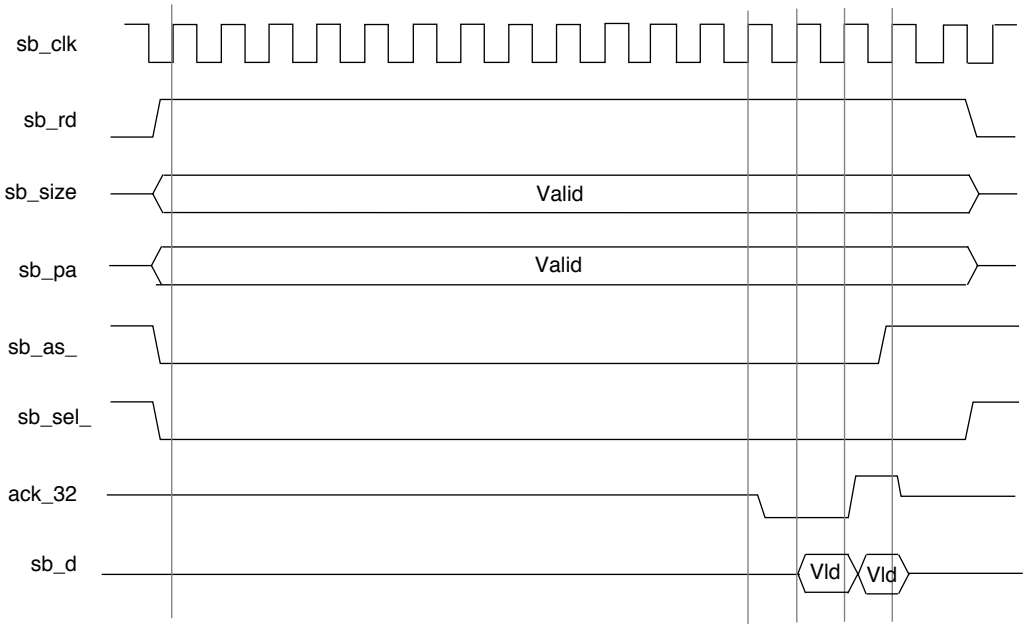

Figure 6-38 Counter/Timer Two-Word Burst Read

Counter/Timer Write

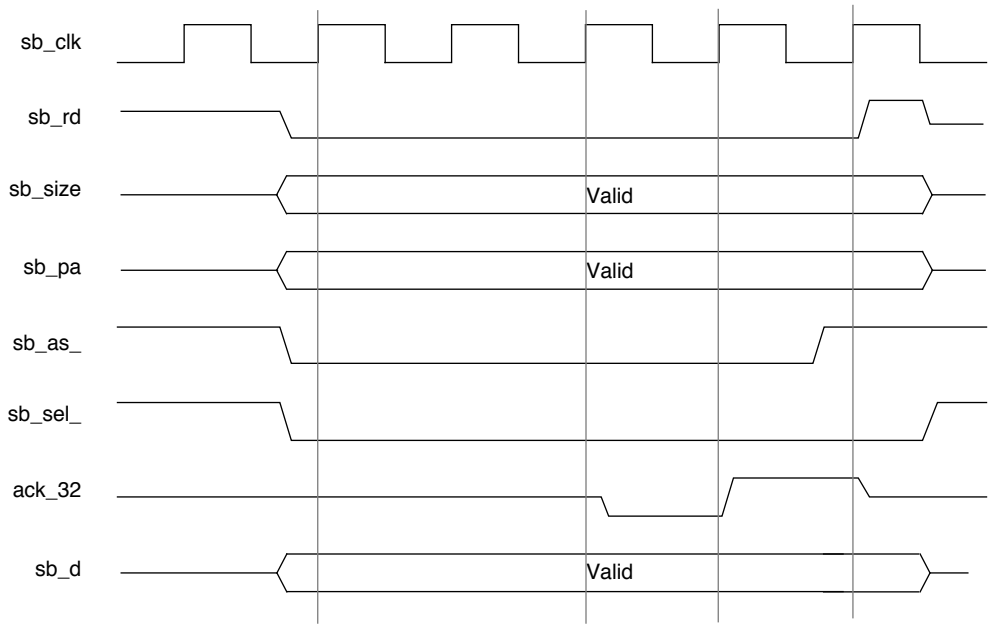

Figure 6-39 Counter/Timer Write

Counter/Timer Two-Word Burst Write

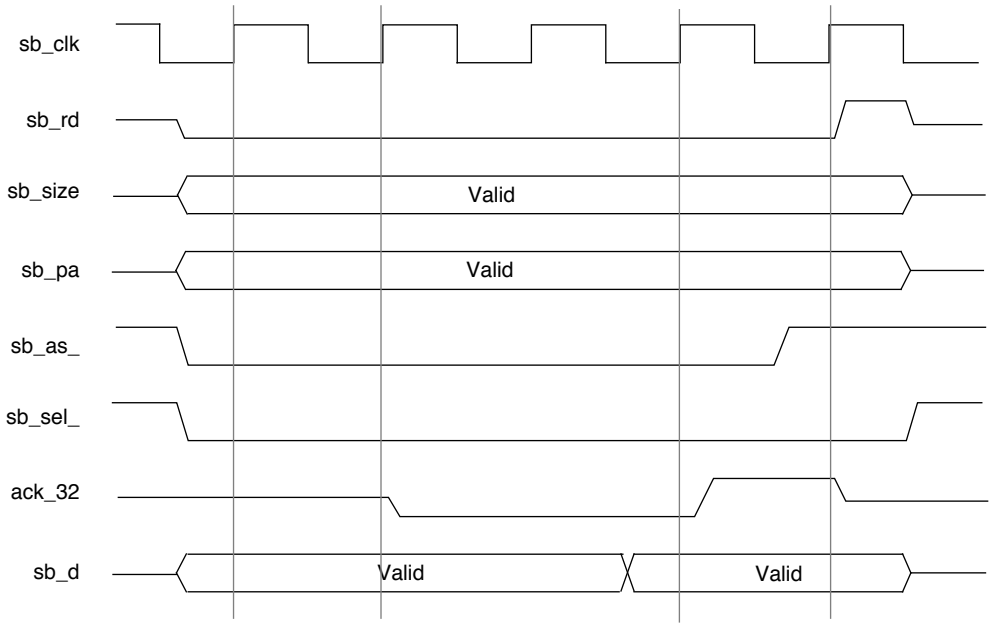

Figure 6-40 Counter/Timer Two-Word Burst Write

Interrupt Controller/SCSR Read/Write

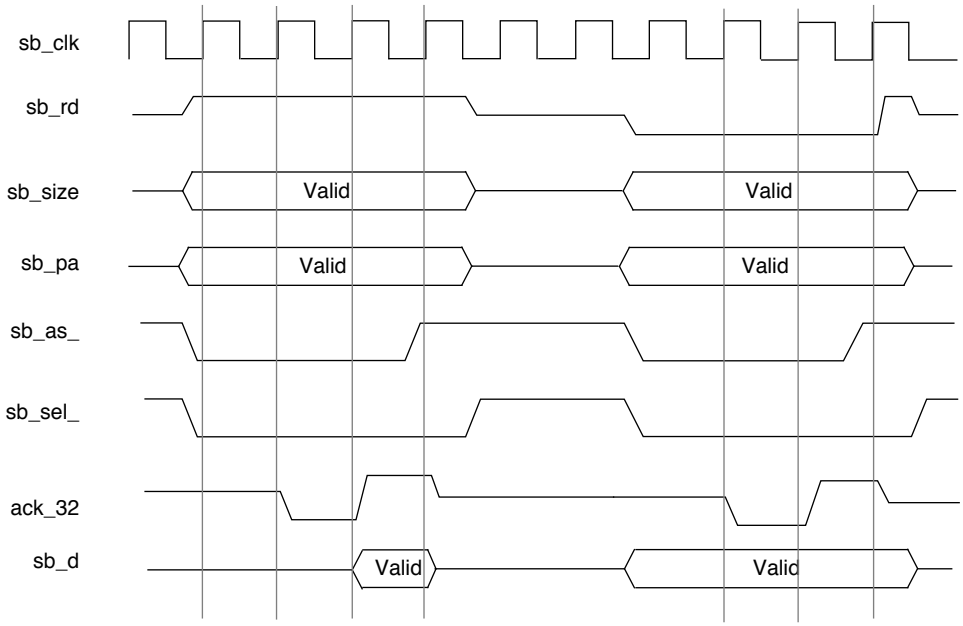

Figure 6-41 Interrupt Controller/SCSR Read/Write

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 againstESD 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

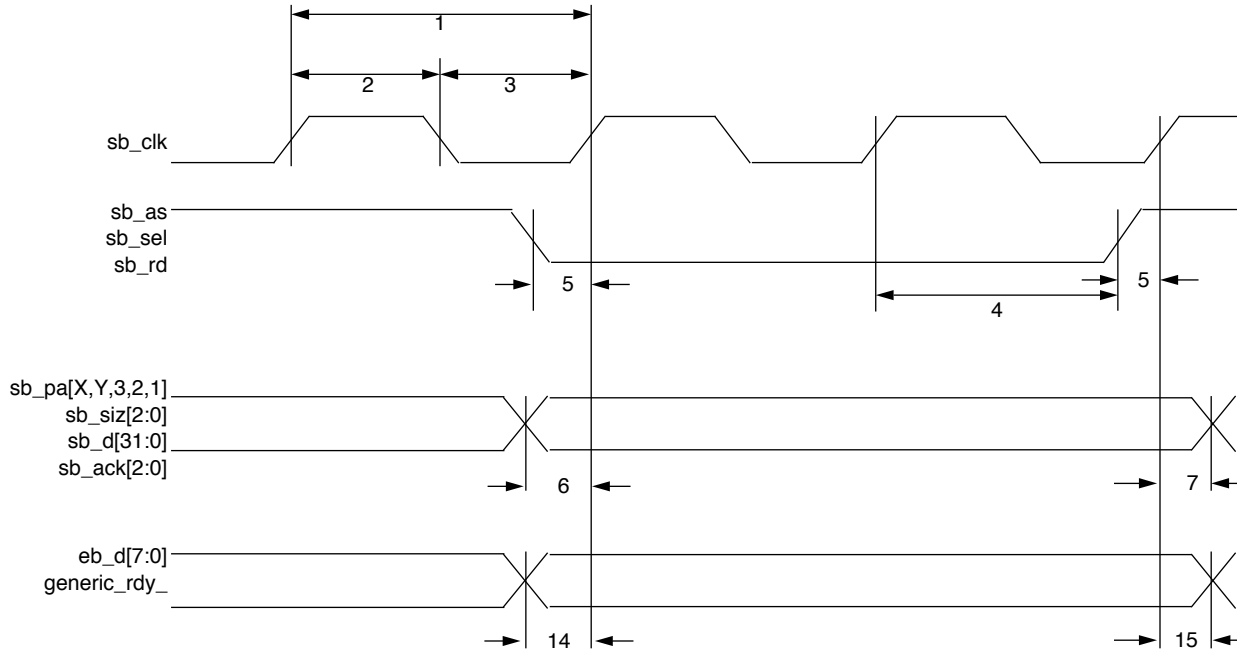

Figure 6-42 SBus/EBus Input Signals

Counter/Timer Input Clock

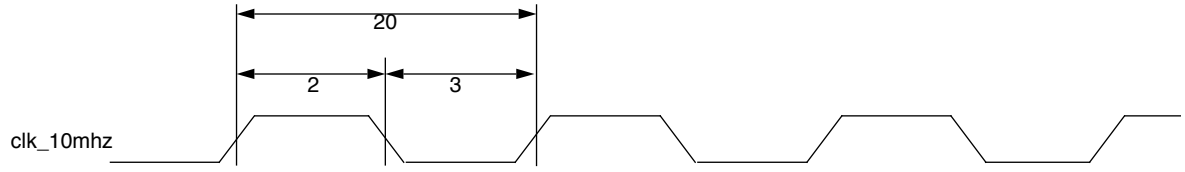

Figure 6-43 Counter/Timer Input Clock

SBus/EBus Output Signals

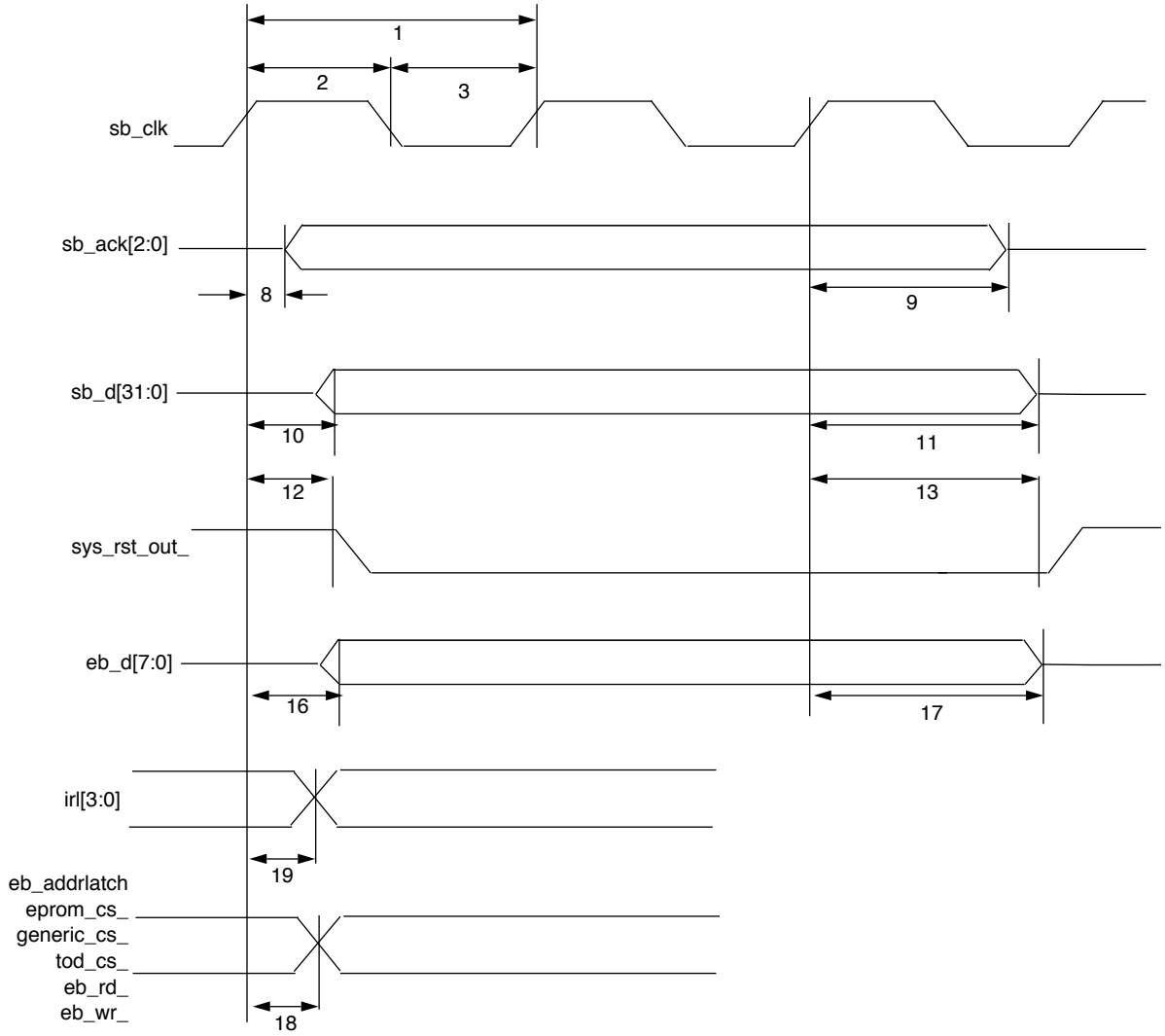

Figure 6-44 SBus/EBus Output Signals

Floppy Controller Clock Inputs

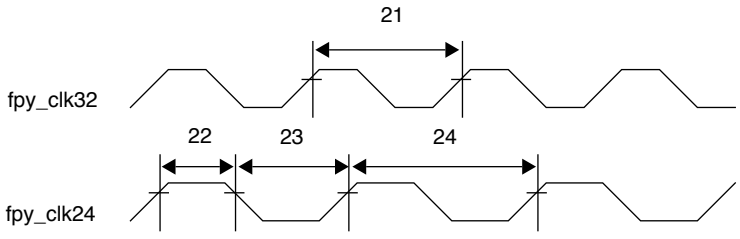

Figure 6-45 Floppy Controller Clock Inputs

Floppy Drive Interface Timing

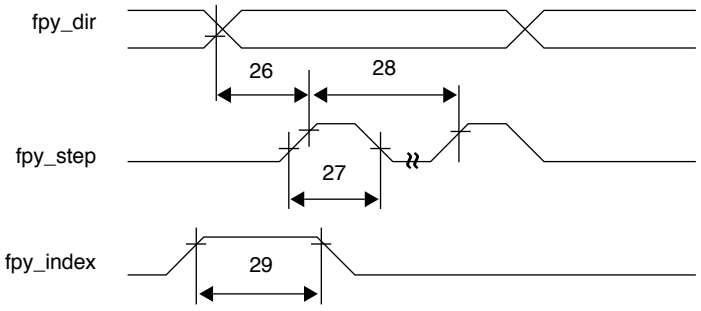

Figure 6-46 Floppy Drive Interface Timing

Floppy Read Data

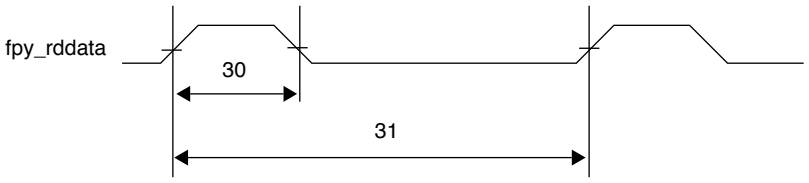

Figure 6-47 Floppy Read Data

Floppy Write Data

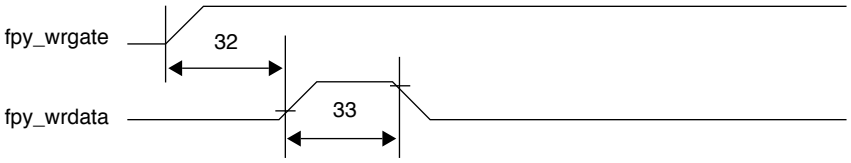

Figure 6-48 Floppy Write Data

Serial/Keyboard/Mouse Clock Input

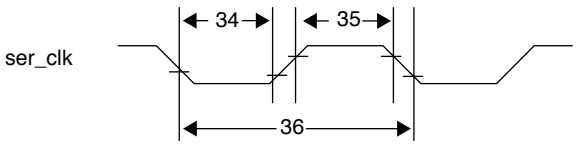

Figure 6-49 Serial/Keyboard/Mouse Clock Input

Serial Pulse Widths

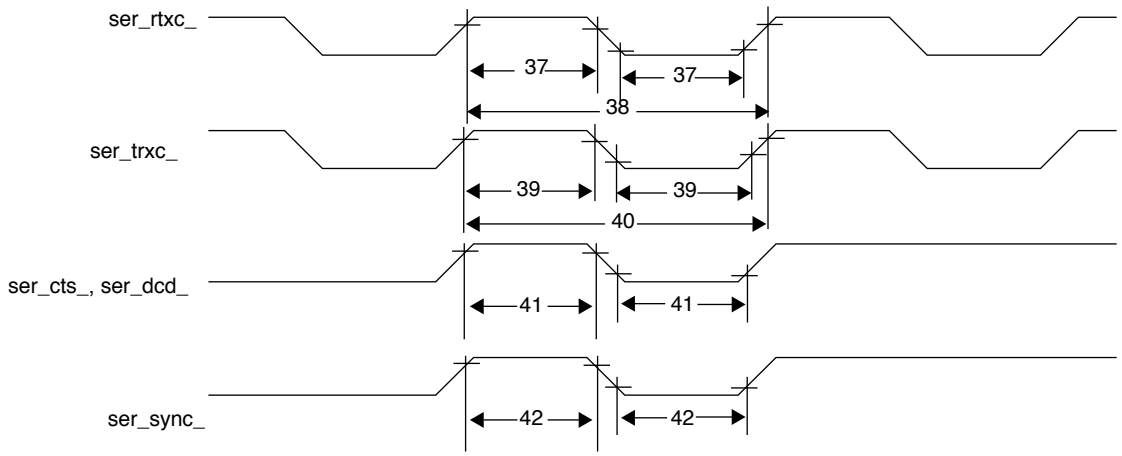

Figure 6-50 Serial Pulse Widths

Serial Data Timing

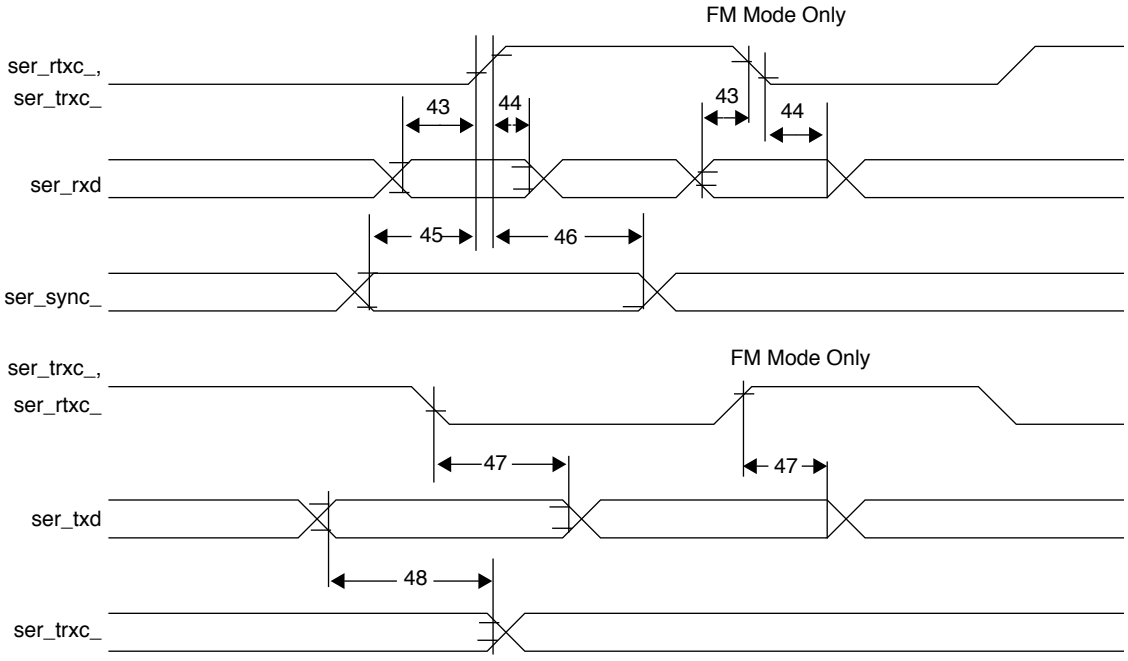

Figure 6-51 Serial Data Timing

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

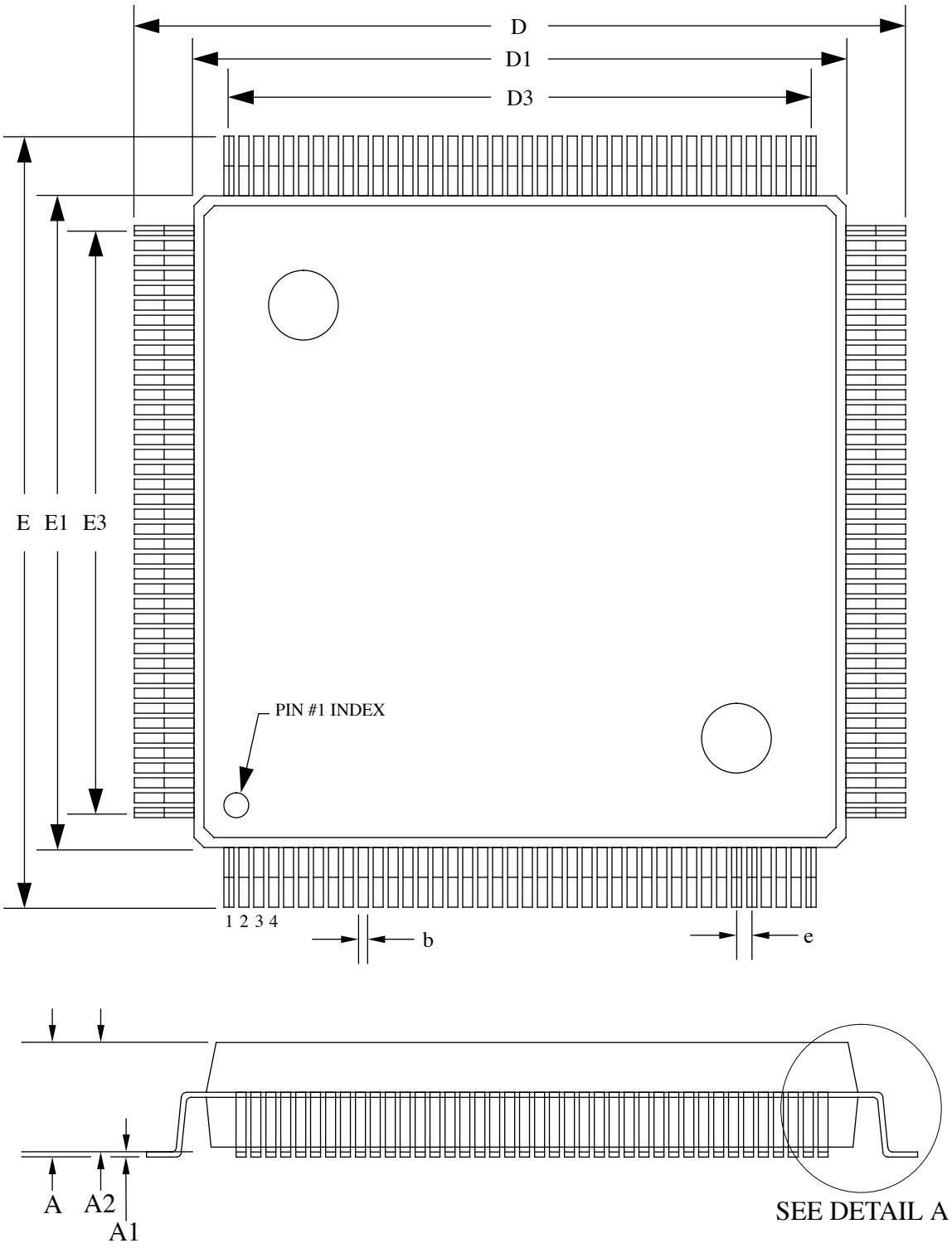

Figure 6-52 Mechanical Packaging Specification

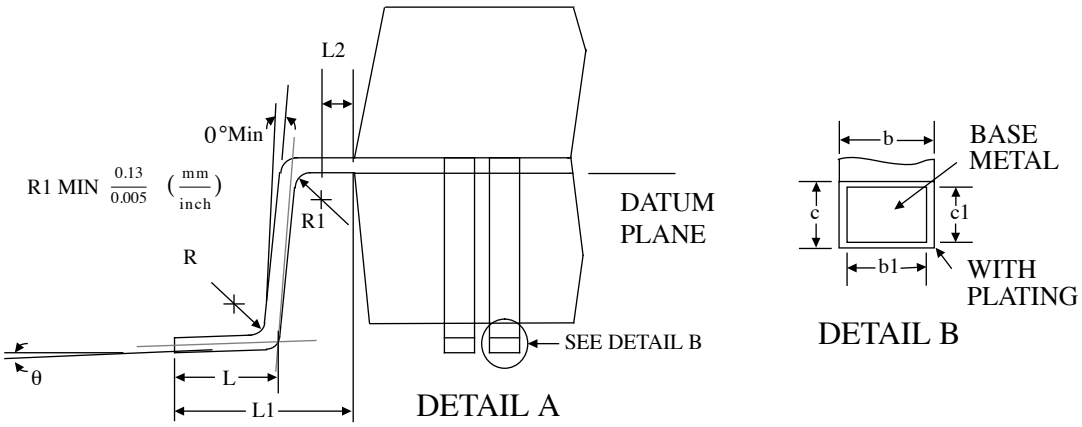

Figure 6-53 Mechanical Packaging Specifications (Detail A and B)

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 distrbution 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.