BAC (Bus Arbiter/Controller)

GENERAL DESCRIPTION

The MULTIBUS II Bus Arbiter/Controller (BAC) is an 84-pin, CMOS component that embodies the Arbitration and system control line functions of the MULTIBUS II Parallel System Bus (iPSB). The BAC provides bus arbitration for systems with multiple masters, is processor independent and conforms to the MULTIBUS II bus architecture specification. Figure 1 presents its pin configuration and Figure 2 presents a block diagram of the BAC.

FEATURES

- o Provides standard MULTIBUS II bus interface
- o Reduces component part count and board area
- o Reduces board design and debug time
- o Provides the arbitration and control logic for MULTIBUS II agent on the Parallel System Bus (iPSB)
- o Permits an iPSB bus agent to behave as requestor and/or replier on the bus
- o Processor independent local bus interface
- o Supports maximum bus speed at 10 MHz
- o Generates and checks parity for System Control lines
- o Supports Burst Transfers
- o Companion to Message Interrupt Controller (MIC)

Note:
An asterisk following a signal name indicates that the signal is active when low.

The following are trademarks of Intel Corporation : Intel, MULTIBUS, iPSB, iLBX, iSSB, iSBX, MULTICHANNEL, and MULTIMODULE.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

PIN CONFIGURATION

63 LOCK*	61 REQU ESTA	60 ADD0*	58 GRANTA	55 PAR3*	54 OTHER READY	51 CLK	48 TIMOUT	46 IBUS ERR*	45 OBREQ*	42 GND
66 LAST INA*	64 PRIO RITY	62 LAST OUT	59 READYB	56 PAR1*	49 PAR2*	50 WRITE*	47 LACHN	44 OBUS ERR	43 IBREQ*	40 ARBIN 5*
67 SELE CTB*	65 WID TH0*			57 PAR0*	53 VCC	52 GND			41 ARB OUT4	39 ARBIN 4*
69 SPACE 1*	68 SPACE 0*								38 ARB OUT3	37 ARBIN 3*
72 WIDTH 1*	71 LAST INB*	73 VCC						33 VCC	35 ARB OUT2	34 ARBIN 2*
75 REQU ESTB	70 DIREC	74 GND			BAC			32 GND	31 ARB OUT1	36 ARBIN 1*
76 AGERR0	77 AGERR1	78 READYA						28 SC8*	29 ARB OUT0	30 ARBIN 0*
79 SELE CTA*	80 AGERR2								26 SC7*	27 SC6*
81 BROAD CAST	83 N.C.		KEY	7 CDIS*	11 VCC	12 GND			23 SC5*	25 SC4*
82 RSEL1	1 RSEL0	2 GRANTB	5 ADD1*	8 RIO4	10 RIO2	14 RIO0	17 SC0*	20 SC2*	22 SC9*	24 SCOEH
84 EINT	3 RRW	4 RSTNC*	6 DBERR*	9 RESET	15 RIO3	13 RIO1	16 SCOEL	18 SC1*	19 SC3*	21 GND

Figure 1. Pin Configuration (Component Side Perspective)
84-Pin PIN GRID ARRAY PACKAGE

Powered by ICminer.com Electronic-Library Service CopyRight 2003

PIN FUNCTION DESCRIPTION

The Bus Arbiter/Controller component's pin functions are categorized as the iPSB Bus interface signals, Primary and Secondary Agent Interface signals, Address Path Control signals, Error signals, and Register signals. Tables 1, 2, and 3 describe the BAC pin functions. Some of the signals perform dual functions depending upon whether the agent is a requestor or replier on the iPSB bus. These dual function signals are described below in Table 2.

Table 1: BAC Interface pin Descriptions
(Refer to Figure 3)

SYMBOL (NAME)	I/O	DESCRIPTION
ARBIN0*-ARBIN5* (Arbitration lines Input)	I	Direct input from ARB0*-ARB5* lines of the iPSB bus. Used in bus arbitration by a requesting agent. Also used to input Arbitration and Slot I.D.s during Reset initialization.
ARBOUT0-ARBOUT4 (Arbitration lines Output)	O	Output of the Arbitration I.D. register through external inverting open collecter drivers to ARB0*-ARB4* lines of iPSB bus when arbitrating. (ARB5* is driven separately if a high priority request is implemented.)
IBREQ* (Bus Request Input)	I	Direct input from the BREQ* line of the iPSB bus. Used in the arbitration protocol.
OBREQ* (Bus Request Output)	O	Output through external inverter and inverting open collector to BREQ* line of iPSB bus. Part of the arbitration protocol. Also used to enable ARB5* in case of a high priority request.
IBUSERR* (Bus Error Input)	I	Input of BUSERR* line of iPSB bus. Used to detect error and reset the BAC internally on its arbitration and transfer states. If relevant, sets a bit in the ERROR Register and causes the EINT signal to go active on the agent interface. Also registered by the BAC and output on the agent interface for use by other logic on the board.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

OBUSERR (Bus Error Output)	O	Output through external inverting open collector driver to the BUSERR* line of iPSB bus. Used to indicate parity errors on the SCx* and ADxx* lines of the iPSB bus. The SCx* parity is computed internal to the BAC. The ADxx* parity is computed externally on a per-byte level and qualified internally by the BAC for validity of byte transfer on a per-cycle basis.
TIMOUT (Timeout)	I	Input of the TIMOUT* line of iPSB bus through an external inverter. Used to detect time-outs and to reset the BAC internally on its arbitration and transfer states. If relevant, sets a bit in the ERROR Register and causes the EINT signal in the agent interface to go active.
RESET (Reset)	I	Input of the RST* line of iPSB bus through an external inverter. Used to reset the BAC internally and enable initialization of the I.D. registers.
RSTNC* (Reset not completed)	I	Input of the RSTNC* line of the iPSB bus and used to keep the BAC component under reset.
SC0*-SC9* (System Control lines)	I/O	Bidirectional lines used to input/drive the SC0*-SC9* lines of the iPSB bus through two external 5-bit transceivers.
SCOEL-SCOEH	O	Direction control for the two external transceivers on the SC0*-SC9* path. The partitioning of the SCx* lines between the two transceivers corresponds to the Owner/Replier roles. SCOEL is active on the BAC that owns the iPSB bus. SCOEH is active during the request phase on a Requestor and during the reply phase on a Replier. The only exception is in case of a Broadcast operation when the Requestor drives SCOEH during the reply phase as well.
LACHN (Latch)	I	Input through external inverter of one of AD0*-AD19* lines of iPSB bus as selected on the backplane and used to validate the Arbitration and Slot I.D.s to be registered under reset initialization.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

CLK (Clock)	I	Input through an external inverting receiver of the BCLK* line of the iPSB bus.

Table 2 BAC Agent Interface Pin Descriptions
(Refer to Figures 6, 7, and 8)

SYMBOL (NAME)	I/O	REQUESTOR/REPLIER	DESCRIPTION
REQUESTA (Request)	I	REQUESTOR	Input from primary agent indicating a request for the iPSB bus. Should be held active until GRANTA output is received by the agent.
REQUESTB (Request)	I	REQUESTOR	Input from secondary agent indicating a request for the iPSB bus. Should be held active until GRANTB output is received by the agent.
PRIORITY (Priority)	I	REQUESTOR	Input from either the primary or secondary agent qualifying REQUESTA/B as high priority. Should be multiplexed externally and held active until the corresponding GRANTA/B output is received by the agent. This signal is also driven around the BAC to the iPSB bus as the ARB5* line. Hence, it should be stable when the BAC is in the midst of arbitration.
GRANTA (Grant)	O	REQUESTOR	Output to the primary agent indicating ownership of the iPSB bus. Used to remove REQUESTA (and PRIORITY) input. Active for at least two clock cycles and withdrawn when request phase of transfer cycle occurs (i.e., SC0* low). May be used to output enable on-board address buffers.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

GARANTB (Grant)	O	REQUESTOR	Output to the secondary agent indicating ownership of the iPSB bus. Used to remove REQUESTB (and PRIORITY) input. Active for at least two clock cycles and withdrawn when request phase of transfer cycle occurs (i.e., SC0* low). May be used to output enable on-board address buffers.
READYA (Ready)	I	REQUESTOR	Input from primary requesting agent indicating readiness as below: - Of request phase information (WIDTH0*-WIDTH1*, SPACE0*-SPACE1*, WRITE*). Address validity is implied. - Of reply phase information (LASTINA*) During the reply phase this input indicates the validity of write data or readiness to read data.
		REPLIER	Input from primary replying agent indicating readiness of reply phase information (AGERR0-2). Input indicates validity of read data or readiness to consume write data.
READYB (Ready)	I	REQUESTOR	Input from secondary requesting agent indicating readiness as below: - Of request phase information, the BAC 84110 assumes a write transfer width of 32 in message space. BROADCAST must be specified if required.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

READYB (Ready)	I	REQUESTOR (Cont'd)	- Of reply phase information, LASTINB* must be hardwired low for a MIC interface but may be used in a more general message implementation. Data must be valid on the bus by the next cycle.
		REPLIER	Input from secondary replying agent indicating readiness of reply phase information (AGERR0-2). Input indicates validity of read data or readiness to consume write data.
WIDTH1*- WIDTH0* (Width)	I	REQUESTOR	Input from primary requesting agent indicating width of requested transfer as one of 8-, 16-, 24-, or 32-bits and qualifying a request phase READYA. Encoding is as follows: 8-bit: WIDTH1* = H, WIDTH0* = H 16-bit: WIDTH1* = H, WIDTH0* = L 24-bit: WIDTH1* = L, WIDTH0* = H 32-bit: WIDTH1* = L, WIDTH0* = L
SPACE1*- SPACE0* (Space)	I	REQUESTOR	Input from primary requesting agent indicating the address space of the requested transfer as one of memory, I/O, message, or interconnect and qualifying a request phase READYA. Encoding scheme is as follows: Memory: SPACE1* = H, SPACE0* = H I/O: SPACE1* = H, SPACE0* = L Message: SPACE1* = L, SPACE0* = H Interconnect: SPACE1* = L, SPACE0* = L
WRITE* (Write)	I	REQUESTOR	Input from primary requesting agent indicating requested transfer to be either read or write and qualifying a request phase READYA. Asserted for a write.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

BROADCAST (Broadcast)	I	REQUESTOR	Input from secondary requesting agent indicating the requested transfer is a Broadcast and qualifying a request phase READYB. Must be active during entire transfer cycle.
	I	REPLIER	Input from secondary agent indicating that selection as a replier is in the broadcast mode and qualifying SELECTB*. Must be valid starting from when SELECTB* goes active to the last data handshake.
LASTINA* (Last Input)	I	REQUESTOR	Input from primary requesting agent indicating the current READYA should correspond to an EOC on the iPSB bus.
LASTINB* (Last Input)	I	REQUESTOR	Input from secondary requesting agent indicating the current READYB should correspond to an EOC on the iPSB bus. For a MIC component implementation, this input is hardwired active low.
OTHERREADY (Other Ready)	O	REQUESTOR	Output to either primary or secondary requesting agent indicating readiness in the previous cycle of other agent involved in a transfer cycle (i.e., SC4* active). This signal is generated internally by the BAC to a requesting agent in case of a Broadcast transfer.
	O	REPLIER	Output to either primary or secondary replying agent indicating readiness in the previous cycle of other agent involved in a transfer cycle (i.e., SC3* active).

Powered by ICminer.com Electronic-Library Service CopyRight 2003

LOCK* (Lock)	I	REQUESTOR	Input from primary requesting agent indicating that the bus is to be locked.
SELECTA* (Select)	I	REPLIER	Input from primary replying agent indicating that the BAC should execute as a replier the iPSB bus. Selection is based on the space and address information received by the on-board selection logic from the bus during a request phase.
SELECTB* (Select)	I	REPLIER	Input from secondary agent (MIC) indicating that the BAC should execute as a replier the iPSB bus. Selection is based on a destination address match in message space. If selection is in the Broadcast mode, the BROADCAST input should be simultaneously activated.
AGERR0- AGERR2 (Agent Error)	I	REPLIER	Inputs shared between primary or secondary replying agent indicating the nature of agent error and qualifying a reply phase READYA or READYB respectively. This error is combined with any iPSB protocol violations detected internally by the BAC before being driven onto lines SC5*-SC7* of the iPSB bus. Encoding scheme is as follows: (L=Low, H=High)

AGERR	2	1	0
Reserved	H	H	H
Reserved	H	H	L
Agent Data Error	H	L	H
NACK Error	H	L	L
Transfer Not Understood	L	H	H
Continuation Error	L	H	L
Width Error	L	L	H
No Error	L	L	L

Powered by ICminer.com Electronic-Library Service CopyRight 2003

LASTOUT (Last Output)	O	REPLIER	Output to either primary or secondary replying agent indicating last data transfer (i.e., SC2* active in previous cycle).

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Table 3 Address Path Control, Error Control, and Register I/O Control Pin Functions.

CATEGORY	SYMBOL (NAME)	I/O	DESCRIPTION
ADxx* Path Controls	PAR0*-PAR3* (Parity)	I	Parity inputs from on-board parity checkers for bytes 0-3 of the ADxx* bus. Low for invalid parity. Qualified internally by the BAC to drive OBUSERR.
	ADD0*-ADD1* (Address)	I	Least significant two bits of the ADxx* bus as seen during a request phase on the iPSB bus.
	DIREC (Direction)	O	Transmit/Receive direction control for transceivers in the address/data and parity paths. Valid for all phases on Requester or Replier.
Error Controls	DBERR* (Bus-Error)	O	For use by the agent interface logic to sense a BUSERR*.
	EINT (Error Interrupt)	O	Interrupt signal to the primary agent indicating a relevant Bus Error, Timeout or Agent Error.
Register I/O controls	RIO0-RIO4 (Register I/O)	I/O	Bidirectional data lines used to read/write to the Arbitration, Slot or Error Port as needed.
	RSEL0-RSEL1 (Register Selection)	I	Input to select between three internal registers.
	RRW (Register Read/Write)	I	Read/Write selection inputs for registers. High for a write.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Miscellaneous	CDIS*	I	Input to the BAC component. This signal allows board testers to disable the outputs, allowing external logic to drive the output pins. The inputs are not affected by this signal.

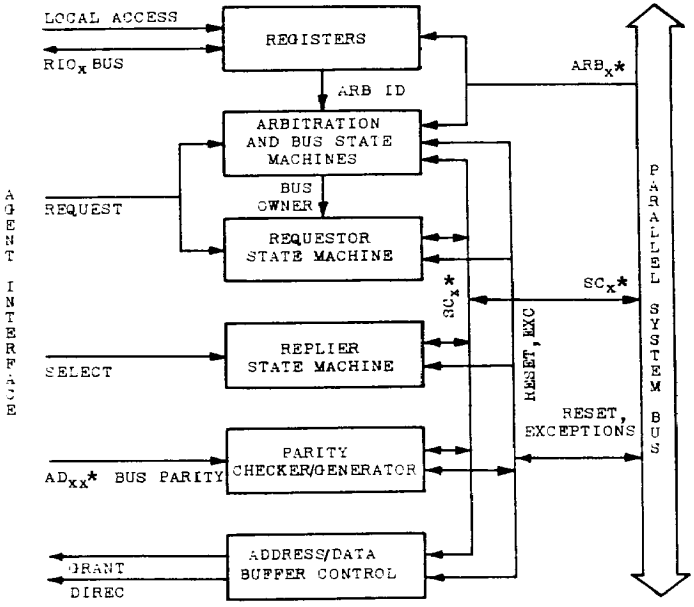


Figure 2 BAC Component Diagram

Powered by ICminer.com Electronic-Library Service CopyRight 2003

FUNCTIONAL DESCRIPTION

INTRODUCTION

The Bus Arbiter/Controller (BAC) provides a generalized interface to the Parallel System Bus (iPSB) of the MULTIBUS II architecture. It performs the functions of arbitrating for ownership of iPSB bus and conducting bus transfer and exception cycles as a requestor or replier. In addition the BAC supports parity generation on the SCx* lines, parity checking functions on the SCx* and ADxx* lines, and error reporting to the host CPU. For those designs migrating to full message passing, the BAC provides a direct interface to the Message Interrupt Controller (MIC) component.

The BAC has four interfaces: the iPSB bus interface, the Host Interface (primary and secondary agent) in the on-board local environment and the Register Interface. The iPSB bus interface, as stated above, supports arbitration, transfer cycles as a requestor/replier and handling of exception cycles on the iPSB bus. Arbitration and Slot IDs received over the iPSB bus from the Central Services Module (CSM) under reset initialization are registered on-chip using the ARBINx*, RESET and LACHN inputs. Arbitration is supported in both normal and high priority modes by interfacing with the ARBx* and BREQ* lines of the iPSB bus. Transfer cycles are handled by receiving and driving the SCx* lines. Exception cycle support is through the interface with the BUSERR* and TIMOUT* lines. The inverted BCLK* is used for internal clocking of all synchronous events.

The host interface consists of a primary and secondary agent interface. The primary agent interface is targeted towards any general purpose processor type host and supports requestor and replier functions. As a requestor, the primary interface support consists of arbitration in normal or high priority mode, transfers in all four address spaces (memory, I/O, message, and interconnect), transfer widths of 8-, 16-, 24-, and 32-bits, and transfer options (read/ write, block or locked). As a replier, the primary agent interface supports all the above options and reporting of agent errors in the replier mode.

The host secondary agent interface (used by the Message Interrupt Controller) support consists of requestor and replier functions and arbitration in normal or high priority modes. Some features of this interface are not used by the MIC component. As a requestor, the interface supports the message address space, 32-bit wide transfers, write only operations, block transfer options, and broadcast capability. The replier mode supports all the above and the reporting of agent errors. Figure 3 illustrates the BAC component interfaces as implemented in a single board computer environment.

The high priority and agent error reporting inputs are shared with the primary agent and need to be multiplexed externally as appropriate. Of these, the MIC does not use the high priority input, the burst tansfer capability or the broadcast feature. However, these features may be used in an upgraded level of message space support.

The register interface support consists of three internal 5-bit registers. Table 4 lists these registers and their functions. These functions support the on-chip registration of the arbitration and slot I.D. numbers supplied by

Powered by ICminer.com Electronic-Library Service CopyRight 2003

the CSM during initialization. The BAC also provides a third register which holds the latest error information received from the iPSB bus.

Table 4 BAC Register-Description

NAME	FUNCTION
SLOTID	Holds the slot I.D. of the board
ARBID	Holds the arbitration I.D. of the board.
ERROR	Indicates nature of bus error, timeout, or reported agent errors.

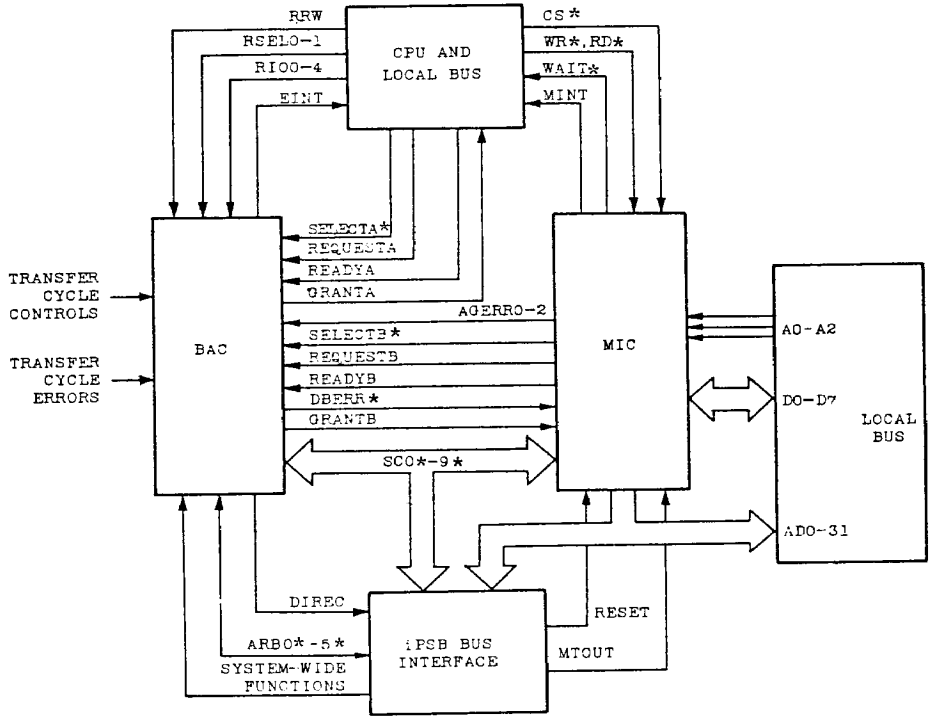


Figure 3 BAC Component Interface In Single Board Computer Environment

Powered by ICminer.com Electronic-Library Service CopyRight 2003

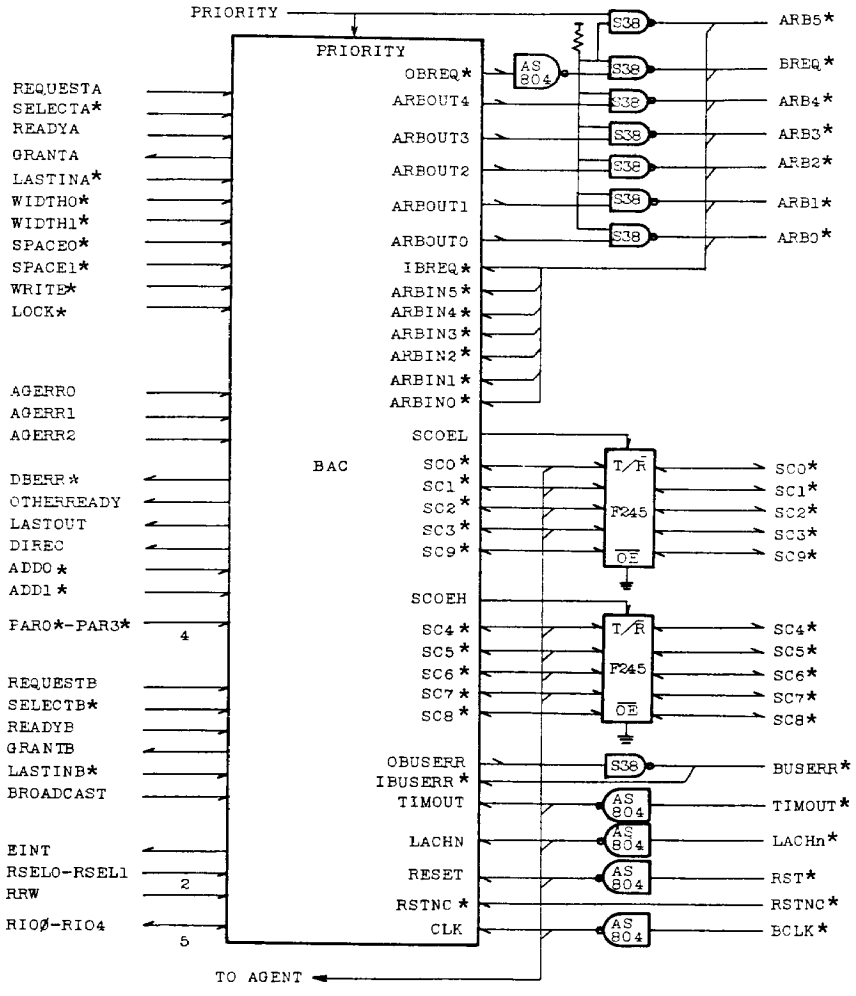


Figure 4 Recommended BAC Interface To iPSB Bus

Powered by ICminer.com Electronic-Library Service CopyRight 2003

MODES OF OPERATION

An agent may interface with the BAC as a requestor or replier. All inputs and outputs to the Bus Arbiter/Controller are synchronous with BCLK* with the exception of the Register Mode Signals (RIO0-RIO4, RRW, LOCK*, RSEL0-RSEL1), EINT and the Arbitration lines (ARBIN0*-ARBIN5*). The synchronized I/O results in a generalized interface suitable to most kinds of devices. For each agent that is asynchronous with BCLK* and depending on the modes supported by each agent, up to three (3) input signals (REQUESTA/B, SELECTA*/B*, READYA/B) may have to be synchronized externally. Synchronous agents, like Message Interrupt Controller, do not require additional synchronization overhead. The following section defines the signal level protocol and presents the relative handshake timings of each of the various operation protocols:

- Initialization Procedure
- Arbitration Protocol
- Requestor Protocol
- Replier Protocol
- Address/Data Bus Control
- Parity Checking
- Host Error Reporting
- Register Access Protocol

INITIALIZATION PROCEDURE

The Bus Arbiter/Controller component is initialized by the Central Services Module over the iPSB bus. During RST* active, the CSM sends each agent its ARBITRATION I.D. and SLOT I.D. over the ARB0*-ARB4* lines. The BAC receives the I.D.s via the ARB0*-ARB4* lines and registers them internally. The ARBIN5* is used to select between the two registers (Arbitration and Slot) and the LACHN input is used to validate the I.D. This initialization occrus for every slot on the iPSB bus backplane in conformity with the MULTIBUS II bus specification during every cold and warm reset. The RESET input to the BAC is used to initialize all internal state information.

ARBITRATION PROTOCOL

The BAC component arbitrates for iPSB bus ownership in response to a request from the primary or secondary agents (REQUESTA/B) input from the Host interface. The PRIORITY input may be activated by the requesting agent to qualify the Request as a High Priority. This input is shared between the primary and secondary interfaces. In response to a request, the BAC activates OBREQ* and places its Arbitration I.D. onto the ARBOUT0-ARBOUT4 lines. The ARBOUT5 line is absent in the BAC component and may be optionally generated externally as shown in Figure 3. When the BAC component becomes the iPSB bus owner (i.e., successfully completes an arbitration cycle) it activates GRANTA/B at the agent interface. On receiving the GRANTA/B, the agent should wait for one cycle before disasserting the REQUESTA/B and PRIORITY inputs. In case of an exception error occuring on the first cycle of iPSB bus ownership (causing the GRNATA/B to be withdrawn due to loss of bus ownership), this delay ensures the BAC component continues to arbitrate for access of the iPSB

Powered by ICminer.com Electronic-Library Service CopyRight 2003

bus again. This procedure is consistent with the error signalling protocol discussed later in this data sheet. The local agent (primary or secondary) has three bus cycles to complete the request phase. The BAC waits for a READYA/B input to validate the request phase information at its inputs before doing a request phase of a transfer cycle.

It is possible that the BAC component may be parked on the iPSB bus at the time a REQUESTA/B is received. If the BAC component can retain the iPSB bus ownership at this point, the GRANTA/B will be issued in the next clock cycle without undergoing arbitration. Figure 5A below illustrates the timing of the relevant arbitration signals.

The BAC component does not guarantee a maximum latency from a REQUESTA/B to the start of arbitration or the duration of arbitration itself. For further detail on the arbitration protocol, please refer to the MULTIBUS II specification.

In the event that both REQUESTA and REQUESTB are active simultaneously at the time of acquiring ownership, the BAC component issues a GRANTB. This gives message space operations (at the secondary interface) a lower latency.

At the primary agent interface, it is possible to lock out the secondary agent as well as all other bus agents once access has been gained. The LOCK* input from the primary agent undergoes a maximum two cycle internal delay before appearing on the iPSB bus as SC1* active. The BAC asserts SC1* by the reply phase of a transfer cycle. However, SC1* stays active and the BAC component continues to retain bus ownership until two cycles after LOCK* is disasserted. During the period that SC1* is active, the BAC component internally disables the REQUESTB input.

Any interface to the BAC component that intends to use the lock feature must account for the delay between asserting or disasserting LOCK* and the corresponding action taken by the BAC component on the iPSB bus. Figure 5B below illustrates a transfer cycle that has a single cycle reply phase. LOCK* is asserted at the time of the request phase READYA in order that SC1* gets asserted during the reply phase. In a board design, awareness of the criticality of the interface for this feature and assurance that the LOCK* input is valid ahead of the requirement on the iPSB bus is necessary.

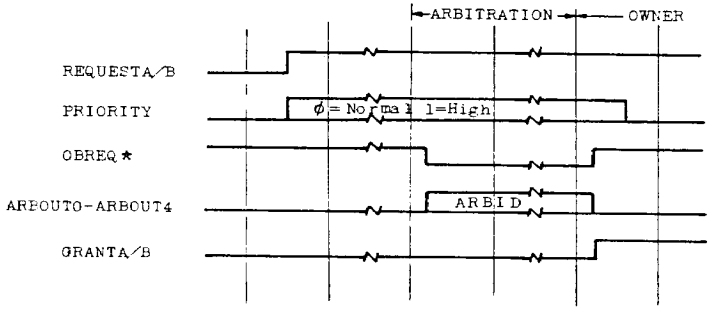


Figure 5A Arbitration Signal Protocol

Powered by ICminer.com Electronic-Library Service CopyRight 2003

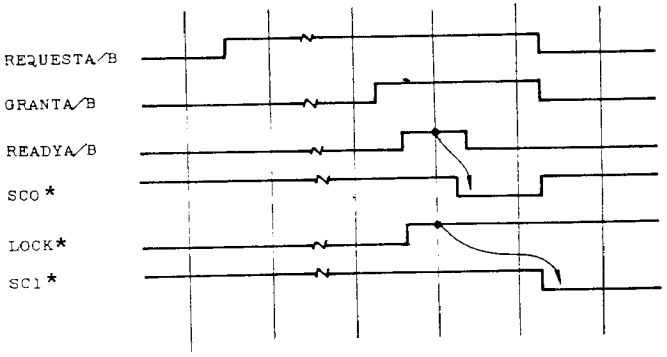


Figure 5B Lock Signal Protocol

REQUESTOR PROTOCOL (Refer to Figures 6A through 6F)

In response to a REQUESTA/B, the BAC component activates a GRANTA/B output upon acquiring ownership of the iPSB bus. The GRANTA/B may be used to enable the Host address buffers. Simultaneously, the BAC component activates DIREC which turns the ADxx* bus transceivers around and sends the address onto the iPSB bus. In response to the GRANTA/B, the primary or secondary agent must estabilsh the address on the ADxx* bus and all the required request phase information at the BAC component inputs (SPACE0*, SPACE1*, WIDTH1*, WIDTH0*, WRITE*, and LOCK*). For the secondary agent, the only relevant input is BROADCAST since the BAC component assumes all the other request phase information. The agent then pulses the READYA/B input for one cycle. Depending on the delay between the GRANTA/B and the READYA/B, one of two situations may occur as shown in Figures 6A and 6B.

Case 1: The situation shown in Figure 6A occurs when the READYA/B is seen active on the clock edge after issuing a GRANTA/B. It is possible to have such an implementation by establishing the READYA/B at the same time as the REQUESTA/B or by feeding the GRANTA/B back through combinational logic as the READYA/B. Upon receiving the ready indication, the BAC component drives the request phase information on to the iPSB bus and asserts SC0*. The GRANTA/B is disasserted immediately after the request phase occurs on the iPSB bus.

CASE 2: The situation in Figure 6B occurs when READYA/B does not go active in the cycle after GRANTA/B but is active one cycle later. This causes the GRANTA/B to be kept active for three cycles, with the request phase occuring on the third cycle. This option permits the agent one extra cycle to establish all the required request phase information and then pulse the READYA/B.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

During the request phase of the transfer cycle, the AD0*, AD1* bits of the ADxx* bus are received by the BAC component from the ADD0* and ADD1* input pins as if they were on the iPSB bus. This information is subsequently used by the ADxx* parity checking logic. The BAC component also reads in the PAR0*-PAR3* inputs during the request phase and drives OBUSERR, if necessary.

It should be noted from Figures 6A and 6B, that the BAC component is capable of going from the request phase to the reply phase with no idle cycles. The same READYA/B input to the BAC component is used to signal the requestor half of the handshake on the iPSB bus (i.e. activate SC3*). Thus, if the READYAB input is active during a request phase (i.e. when SC0* is active), this will be interpreted as a reply phase ready. If the READYA/B is not active during the request phase, the BAC component waits for it to go active before asserting SC3*.

Each handshake concludes on seeing SC4* go active while SC3* is active as shown in Figure 6C. The end of transfer is indicated by asserting LASTINA*/B* low while asserting READYA/B high. This assertion results in the BAC component activating SC2* and SC3* simultaneously. The BAC component responds to any errors reported by the replying agent on lines SC5*-SC7* by activating SC2* unconditionally on behalf of the host and terminating the reply phase as shown in Figures 6E and 6F.

By keeping the READYA/B active during a burst transfer, burst (back-to-back) transfers are possible. Many agents cannot handle data at this rate and hence must pulse READYA/B for one cycle each time.

It is the responsibility of the host to ensure that write data is valid or read data consumable when issuing a READYA/B.

The BAC component's OTHERREADY output signal is used to indicate readiness of the other agent in a transfer cycle. As a Requestor, this output corresponds to sensing SC4* active in the previous cycle. This signal may be used typically while performing Reads in order to consume the read data and then activate the READYA/B, so that the BAC component may complete the handshake.

For a broadcast operation, the secondary agent needs to assert BROADCAST along with all the request phase information and keep it asserted until the end of transfer. Figure 6D shows the signal protocol for a broadcast operation. The requesting BAC component drives both nibbles of SCx* lines (by keeping both SCOEH and SCOEL active) during the entire transfer cycle. The burst transfer feature is a valid option and for each data transfer, the requesting BAC component asserts SC3*, waits for 7 bus clocks and handshakes with SC4*. On the last transfer, SC2* is asserted along with SC3*.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

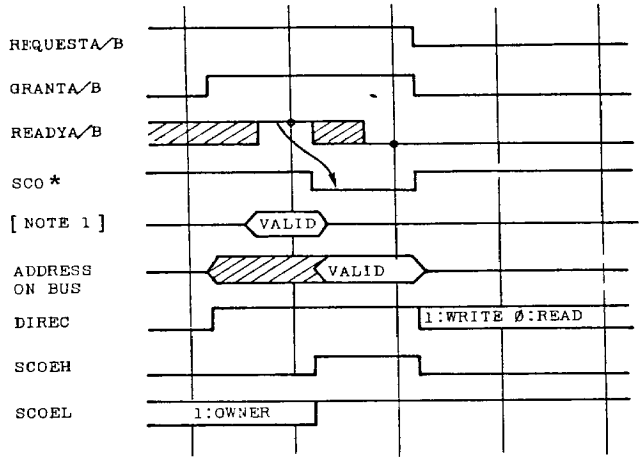
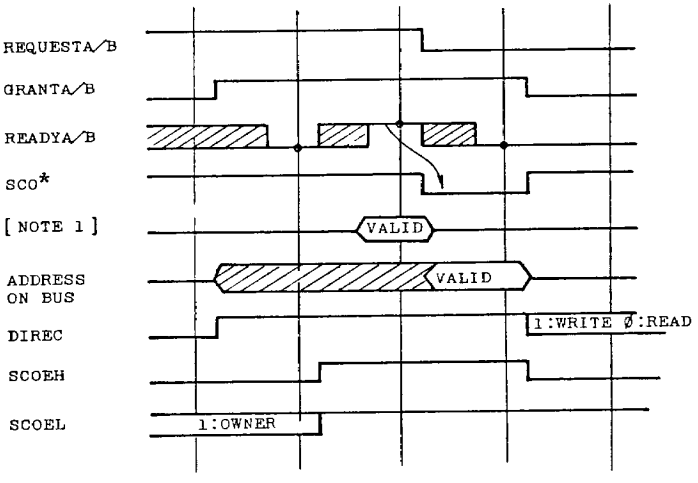


Figure 6A Request Phase (2 CYCLE)



NOTE 1 : REQUEST PHASE INFORMATION AS INPUTS TO BAC
- FROM AGENT A : WIDTHO*-1*,SPACEO*-1*,WRITE*
- FROM AGENT B : BROADCAST

Figure 6B Request Phase

Powered by ICminer.com Electronic-Library Service CopyRight 2003

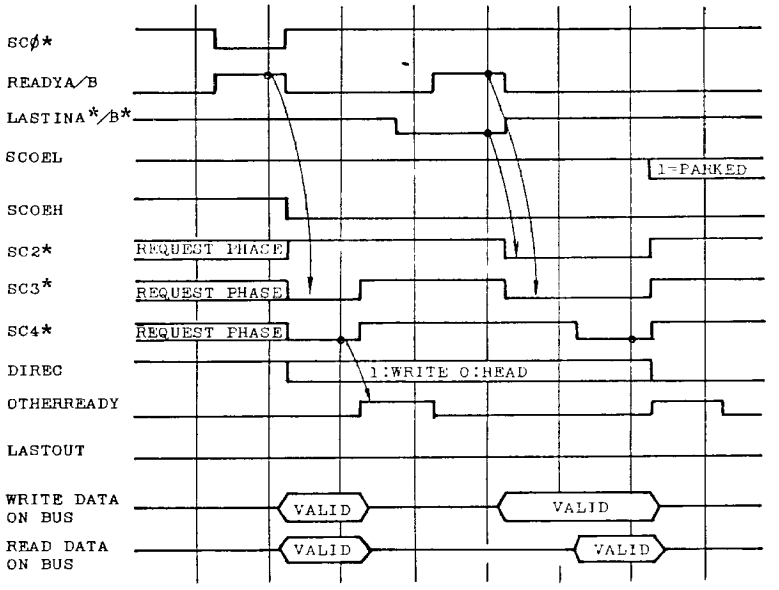


Figure 6C Requestor Handshake

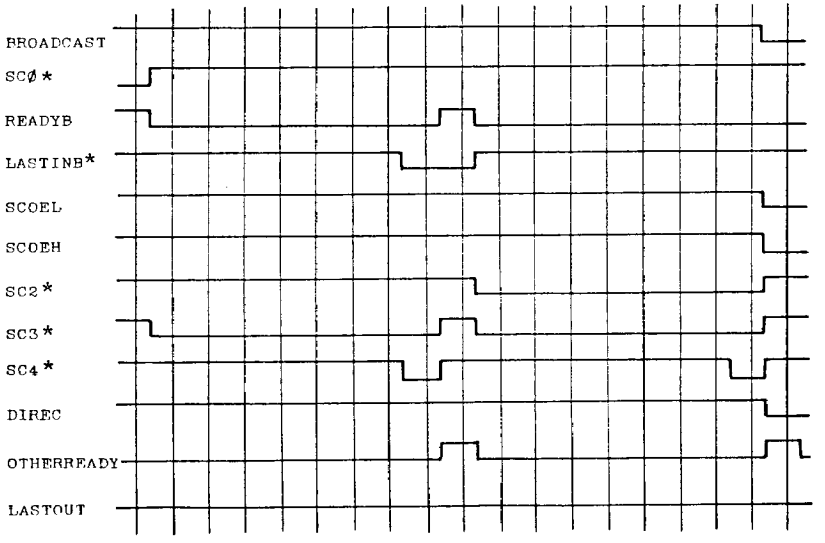


Figure 6D Requestor Handshake (Broadcast)

Powered by ICminer.com Electronic-Library Service CopyRight 2003

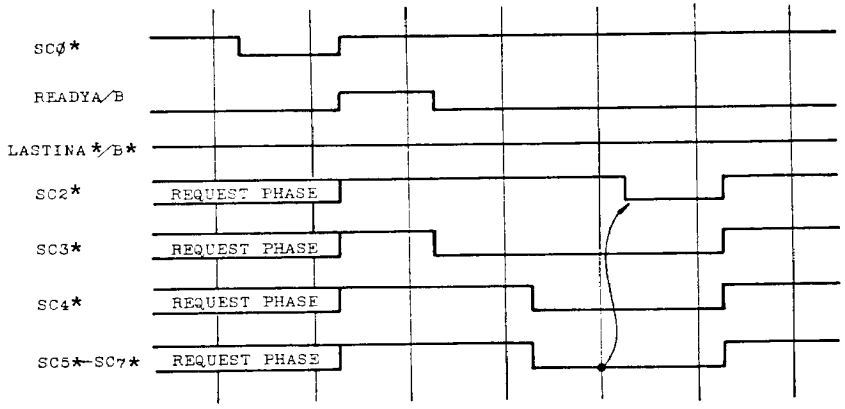


Figure 6E Agent Error END-OF-CYCLE

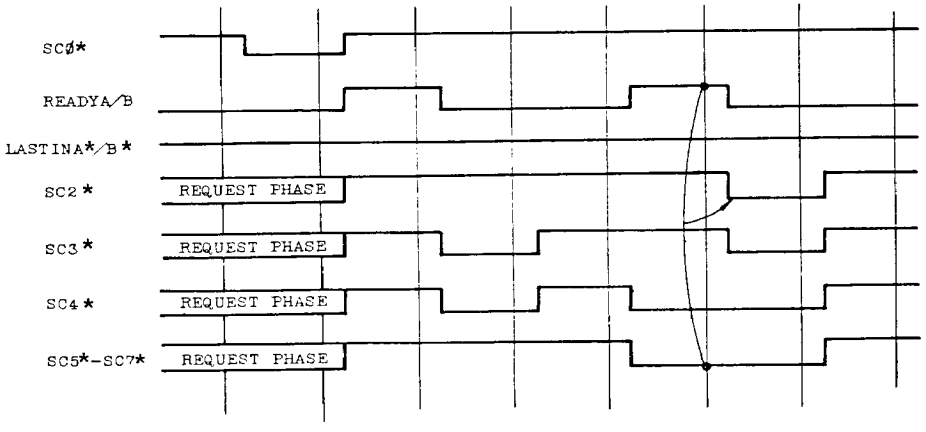


Figure 6F Agent Error END-OF-CYCLE

Powered by ICminer.com Electronic-Library Service CopyRight 2003

REPLIER PROTOCOL (Refer to Figures 7A and 7B)

All agents, except the requesting agent, are potential repliers and actively decode request phase information. During every request phase occuring on the iPSB bus, the BAC component on each module latches in all width, space, read/write and ADD0*/1* information from the iPSB bus. Selection as a replier is an on-board function and is performed (independent of the BAC component) by decoding the ADxx* lines and SC4*, SC5* (space) as seen when SC0* goes active.

The BAC component is informed of its role as a replying agent by the SELECTA*/B* input going active as shown in Figure 7A. Upon being selected as a replier, the BAC component activates SCOEH to drive the higher nibble of the SCx* lines. If the transfer operation is a read, the BAC component simultaneously activates the DIREC output as well.

The SELECTA*/B* input should be kept active for a least two cycles. On the second cycle or later, the BAC component expects a READYA/B from the host. This indicates that the host is ready with "write data" or is ready to consume "read data" on the next cycle. Based on the READYA/B input the BAC component activates SC4* and looks for an SC3* from the replier to conclude this data transfer handshake. If during the handshake, the requestor had asserted SC2* as well, then an end-of-transfer is understood. If an SC2* was not received, the replying BAC component expects further READYA/B inputs from its host.

As in the case of the Requestor, the Replier needs to pulse the READYA/B input to the BAC component for one clock at a time unless the replying host can handle burst (back-to-back) transfers.

Any Agent Errors are input to the BAC component at its AGERR0-2 pins and held active until an end-of-transfer (SC2* active) is seen on the iPSB bus.

The BAC independently checks the request phase information on the iPSB bus for any violations of the MULTIBUS II protocol. This includes the following protocol violations:

- Message requests of 8 or 24 width
- Message read requests
- Interconnect accesses other than 8 bits wide
- Interconnect accesses with either AD1* or AD0* low
- Memory or I/O access of 16 width with AD1* and AD0* both low
- Memory or I/O access of 24 width with AD1* low
- Memory or I/O access of 32 width with either AD1* or AD0* low
- Sequential transfers in interconnect space
- Sequential non-aligned transfers in memory or I/O space

Agnet reported errors are overridden by internally diagnosed errors to become "not understood" errors. The resultant errors appear on the SC5*-SC7* lines (as described in Table 5) when a READYA/B input is received to validate the agent reported errors.

As a replier, the BAC component outputs two signals, OTHERREADY and LASTOUT to the host interface. These are delayed versions of SC3* and SC2* respectively for use by the on-board logic.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

The protocol for a broadcast transfer as a replier consists of asserting the BROADCAST input at the sames time as the SELECTB* input is asserted. The BROADCAST input must be held active until the transfer cycle is completed. This protocol is shown in Figure 7B. In this case, the BAC component does not activate SCOEL since all the SCx* lines are driven by the Requestor BAC component. The READYA/B and AGERR0-2 inputs are ignored since the handshake sequence is controlled by the Requestor BAC component. The BAC component outputs OTHERREADY and LASTOUT are defined as in the case of a regular transfer.

Table 5 Agent Error Types and Related SC Lines

AGENT ERROR	SC7*	SC6*	SC5*
Reserved	Low	Low	Low
Reserved	Low	Low	High
Data Error	Low	High	Low
NACK	Low	High	High
Transfer-Not-Understood	High	Low	Low
Continuation	High	Low	High
Transfer-Width	High	High	Low
No Error	High	High	High

ADDRESS/DATA BUS CONTROL

The BAC component does not receive or drive the ADxx* bus (except for the AD0* and AD1* inputs used for parity qualification described earlier). The DIREC output of the BAC component is used by the host to change the direction of the ADxx* bus transceivers around. These transceivers are normally turned inward for reading the AD lines (DIREC inactive). The DIREC output goes active during the Request phase on a Requestor and during the Reply phase for writes on a Requestor and for reads on a Replier. (Refer to Figures 6C, 6D, 7A, and 7B.)

Powered by ICminer.com Electronic-Library Service CopyRight 2003

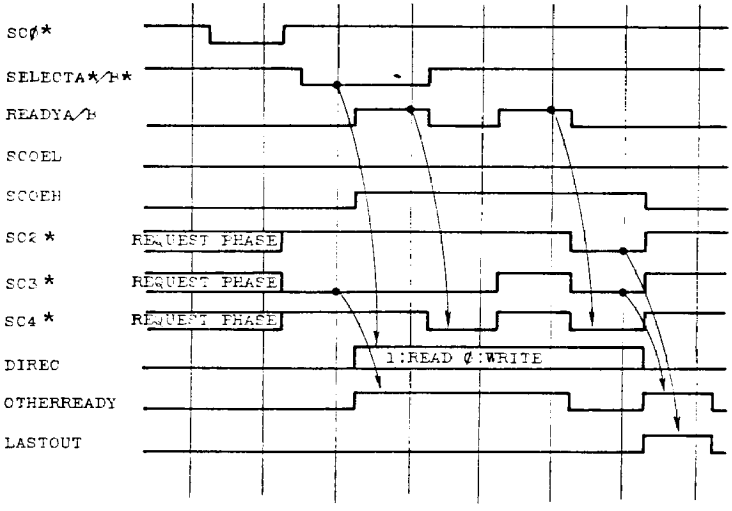


Figure 7A Replier Handshake

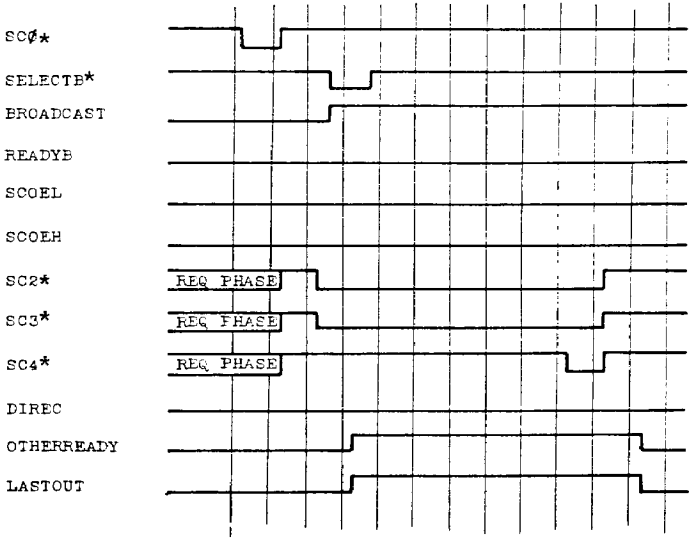


Figure 7B Replier Handshake (Broadcast)

Powered by ICminer.com Electronic-Library Service CopyRight 2003

PARITY CHECKING AND PROTOCOL VIOLATIONS

The BAC component checks parity on the SCx* bus on every bus clock cycle and drives OBUSERR in case of an error. In addition, it qualitifes the byte-level parity inputs received from the on-board ADxx* bus parity checkers. The parity inputs to the BAC component (on PAR0*-PAR3* lines) are internally qualified as follows to drive OBUSERR in the next cycle:

1. For request phase (SC0* active), all BAC components in the system check parity as follows:

   PAR0*-PAR3* are assumed valid if the access was to memory space.

   PAR0*-PAR1* if the access was to any other space.

2. During the reply phase, parity checking is based on the width, space and AD0*, AD1* signalled in the request phase. This information uniquely identifies the valid bytes of data transfer and hence the valid PAR0*-PAR3* inputs as detailed in the MULTIBUS II specification. The BAC component only checks parity when transfer handshakes occur without agent errors being reported. That is, SC3* and SC4* must both be active in that cycle and SC5*, SC6* and SC7* must be inactive at the time.

   The responsibility of checking parity is decided as follows. If the trnasfer is a Read, the Requestor BAC component does the checking. If the transfer is a Write, the Replying BAC component(s) does(do) the checking.

The BAC component checks for iPSB protocol violations in the reply phase of every transfer cycle. A bus error is generated in either of the following illegal situations:

a. If the SC2* input is active while the SC3* input is inactive during the reply phase.

b. If any of the SC5*-SC7* inputs are active while the SC4* input is inactive during the reply phase.

HOST ERROR REPORTING (Refer to Figure 7C)

The BAC component has an error reporting protocol that supports the MULTIBUS II architecture. It includes Bus Errors, Timeouts and Agent reported Errors. The role of the primary and secondary agent interfaces is extended to this protocol.

As discussed earlier, the secondary interface is dedicated to message support for a MIC-like device. Such a device would have an error reporting path to the host CPU that is limited to its activity on the iPSB bus. Message related errors are typically recoverable in software and an error reporting path dedicated to message traffic is therefore justifiable.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

The BAC component has a separate error indication line to the host CPU called EINT. Errors reported on EINT are linked with ownership of the iPSB bus. The window (X) for reporting errors on EINT is delayed one cycle from the duration of bus ownership. Further, to prevent duplication of error reporting from the BAC component and from the message device to the host CPU, the BAC component ignores all errors that can be associated with message transfers. The window (Y) for ignoring errors starts on the cycle after a message request phase (i.e. SCO* active) and terminates on the cycle after the EOC handshake (both cycles inclusive).

In summary, from the window S during which errors are reported on EINT, all errors occuring in window(s) Y are blocked out since these are reported through the message device.

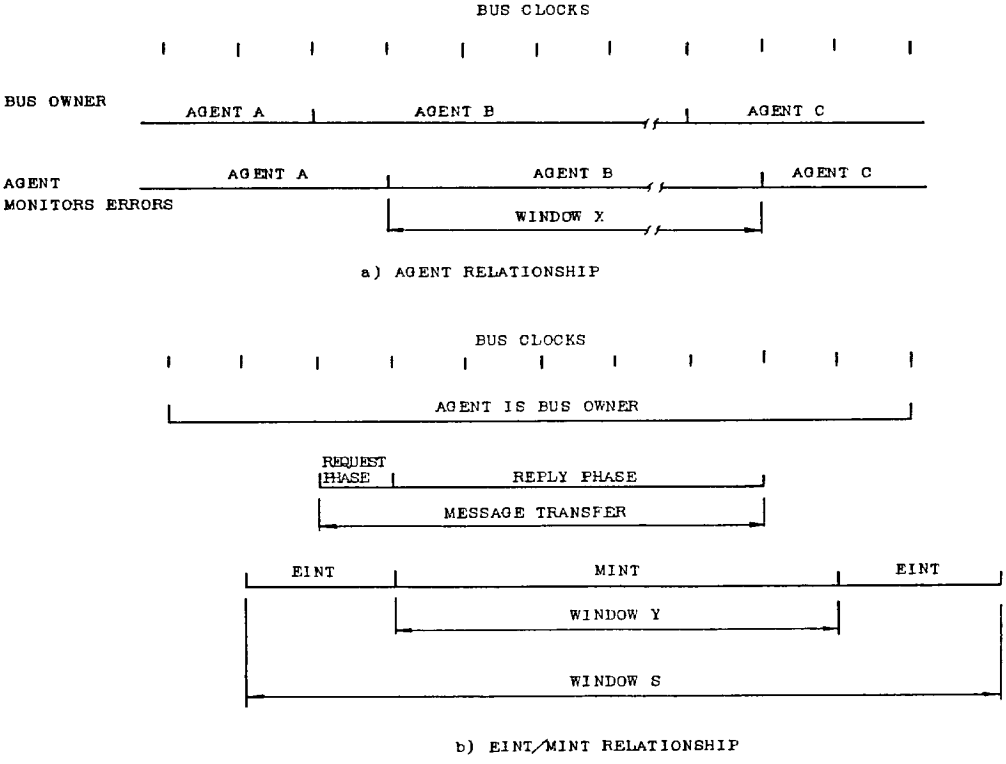


Figure 7C Error Detection Windows on iPSB Bus

Powered by ICminer.com Electronic-Library Service CopyRight 2003

The nature of an error is identified in an Error Register internal to the BAC component that can be read by the host CPU. The EINT signal can be cleared by writing to this register. The register access protocol is discussed in the Register Access Protocol section of this data sheet.

It is possible for multiple errors to occur while the first error is being serviced. The only error that must not be lost to the host CPU is a Bus Error. The BAC component guarantees that any Bus Error that occur while a previous error is pending in the Error Register will be registered separately and made known to the host CPU when the current error is cleared. In such a situation, the EINT line will pulse inactive for a minimum of 2 bus clock cycles less 20 nsec.

The host CPU recovery mechanism needs to reside entirely on-board. Any accesses to resources on the iPSB bus to aid in the recovery may cause further agent errors or exceptions that may make the recovery impossible.

REGISTER ACCESS PROTOCOL

The Bus Arbiter/Controller has three (3) 5-bit registers that may be accessed by the agent interface over a 5-bit bus (RIO0-RIO4). These registers are the SLOTID register, the ARBID register, and the ERROR register. Table 4 described these registers. The SLOTID and ARBID registers are initialized by the CSM during the iPSB bus system cold or warm reset period. All three registers can be read by the Host CPU via the RIO0-4 bus of the BAC component by suitably setting the RSEL0-RSEL1 and RRW inputs. The ARBID register contains the priority number used in arbitration and may be written into via the RIO bus as well. The SLOTID value is the board's physical slot location and cannot be changed except through another CSM initiated reset.

The Error register value reflects the cause of the EINT signal. This register contains a snapshot of the value of the TIMOUT* line and any Agent Errors received on a specific cycle along with the value of the BUSERR* line on the following cycle. This snapshot enables the host CPU to decide the validity or otherwise of the agent error bits based on the Bus Error information. If the Bus Error bit is active, the agent error bits should be treated as invalid. Writing to the ERROR register serves to clear the EINT output.

The templates for these registers, as they appear on the RIO bus, are shown in Table 6. Note that the register values as they appear on the RIO bus are positive high true and inverted from the iPSB values. Figure 8 specifies the valid control sequences for reading or writing to these registers and Table 7 lists the related timing information.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Table 6 Template for BAC Registers

Register	RIO4	RIO3	RIO2	RIO1	RIO0
ARBID	ARB4	ARB3	ARB2	ARB1	ARB0
SLOTID	ARB4	ARB3	ARB2	ARB1	ARB0
ERROR	BUSERR	TIMOUT	SC7	SC6	SC5

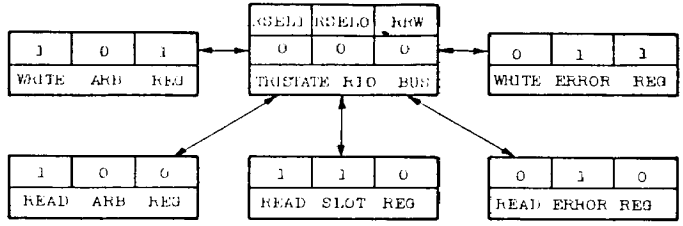
When the RSEL0, RSEL1 and RRW inputs are all low, the BAC component puts the RIO bus in tristate mode, allowing it to be used by external on-board logic. To read any register, the RSEL0 and/or the RSEL1 inputs need to be asserted as shown in Figure 8 while keeping RRW low. Table 7 shows the duration when read data is valid on the RIO bus.

Writing to the ARBID or ERROR registers requires RRW to be asserted high. The RSEL0 input is asserted high to select the ERROR register. No data needs to be setup since writing to the ERROR register is only a command write that causes the EINT output to be disasserted. The actual register clear occurs when the earlier of RRW and RSEL0 is disasserted. The sequence of asserting or disasserting these control inputs is not important. However, a minimum window of 30 nsec. is required when both inputs are asserted.

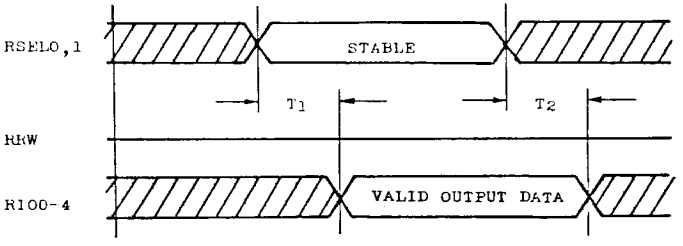
To write to the ARBID register, the RRW and RSEL1 inputs are asserted high. The required ARBID value is setup on the RIO bus. The actual write occurs when the earlier of RRW and RSEL1 is disasserted. The actual sequence of disasserting these command signals is not important. However a minimum window of 30 nsec. is required during which both inputs are active. With respect to this window, the RIO bus inputs must be valid for a minimum duration before and after command deactivation. If the RSEL1 active period extends outside the RRW active window, due care must be taken in driving the RIO bus externally to prevent contention with the BAC component. This is because activating RSEL1 alone corresponds to reading the ARBID register.

If the BAC component is currently using the ARBID in arbitration at the time the host CPU tries to write to it, the value written is stored in a temporary latch within the BAC component. In such a case, the actual update of the ARBID occurs later and is transparent to the host CPU. This may be verified by reading the ARBID register later.

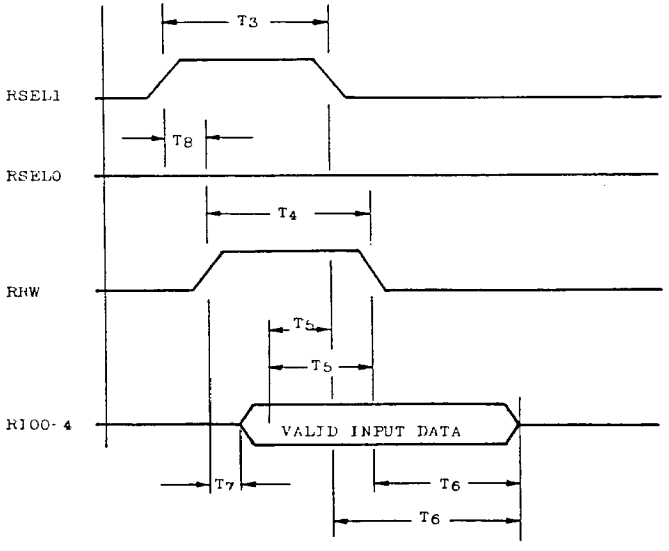
Powered by ICminer.com Electronic-Library Service CopyRight 2003



VALID CONTROL TRANSITIONS



REGISTER READ



REGISTER WRITE (ARB REG)

Figure 8 BAC Register Access Protocol

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Table 7 Register Interface Read/Write A.C. Timing Parameters

PARAMETER	MIN(ns)	MAX(ns)	DEFINITION
T_1	-	50	RIO outputs valid after RSEL0,1 asserted (RRW low)
T_2	0	-	RIO outputs valid after RSEL0,1 diasserted (RRW high)
T_3 Note1	50	-	RSEL0,1 stable high duration
T_4 Note1	50	-	RRW stable high duration
T_5 Note2	30	-	Input data valid before RSEL1, RRW going low
T_6 Note2	5	25/-	Input data valid after RSEL1, RRW going low
T_7 Note3	25/0	-	Input data enable after RRW high
T_8 Note3	-	-	RRW high delay from RSEL0,1 assert

Note1: T3 and T4 should have a minimum overlap of 30 ns.

Note2: Data validity is with reference to earlier of RSEL1 and RRW going low. Writing to the ERROR register is a command only and has no associated input data. If RSEL1 remains high for more than 5 ns. after RRW goes low, then T6 max. is 25 ns. to prevent bus contention.

Note3: T7 min. of 25 ns. is intended to prevent RIO bus contention and is relevant only if T8 exceeds 5 ns. else min. T7 is 0.

BUS ERROR JAMMING PROTOCOL

The occurrence of an exception condition (Bus error or Timout) leads to termination of transfer cycles in progress, allowing all agents up to 3 bus cycles to recover. At the end of this period, a new request phase may occur.

It is conceivable that repliers on the iPSB bus, typically running asynchronously to the bus, have not recovered in this duration of 3 bus cycles and as such would be unable to decode the new request phase.

The BAC component has a feature whereby a replying agent on the primary interface may gain additional recovery time from exceptions by jamming the BUSERR* line until recovery is complete. This results in the exception cycle being extended and all arbitration and transfer activity being disabled.

At the time of an exception signal being received, the BAC component could be in one of two conditions:

Powered by ICminer.com Electronic-Library Service CopyRight 2003

1. The BAC component could already have been selected as a replier. In this case, the handshake signals could be in one of the following situations:

   a. An EOC handshake occurs (i.e. SC2* and SC4* active).
   b. A non-EOC handshake occurs (i.e. SC3* and SC4* active, SC2* inactive.)
   c. The host CPU has signalled a READYA in a previous cycle and is waiting for a handshake (i.e. SC4* active and SC3* inactive).
   d. The host CPU is not yet ready (SC4* inactive and READYA is inactive).

2. The BAC component has not yet been selected. In this case, one of the following situations may occur:

   a. The BAC component is just being selected. (Selection corresponds to SELECTA* going active).
   b. The BAC component gets selected on the cycle after the exception.
   c. The BAC component never gets selected.

The BAC component interface requires that the on-board logic that generates the READYs and monitors the handshake signals should also interface to the TIMOUT* line (registered externally) and the DBERR* line from the BAC component. Any exception occurring while active as a Replier should be treated as equivalent to LASTOUT and be matched with a READYA. If such a READYA has not already been issued to the BAC component before the exception occurring, it must be issued on the cycle of the exception or one cycle later, failing which the BAC component activates the OBUSERR output until the READYA is received.

Figures 9A and 9B illustrate the possible scenarios discussed above. The EXC* signal is a virtual signal. It is normally an "OR" of IBUSERR* and TIMOUT and received by the BAC component. Where a JAM is indicated, the EXC* signal corresponds to the OBUSERR output of the BAC component.

In case 1a of Figure 9A the exception does not affect the recovery of the host CPU. Therefore, the BAC component does not expect a READYA. In case 1b, the host CPU may be performing the next data access cycle before the delayed exception is noticed. It may, hence, be necessary to jam the bus if the host CPU cannot recover in time. In case 1C, the host CPU would receive the delayed exception and interpret it as the EOC handshake, thus recovering safely. The situation in case 1d, Figure 9B, is similar to 1b and would cause the host CPU to receive the exception in the midst of a local data access cycle.

In cases 2a and 2b (Figure 9B), the host CPU is in the midst of an on-board access when it receives the exception and a corresponding READYA is called for. In case 2c, the host CPU either does not intend to select itself or else receives the exception before it is committed as a Replier. In both situations, the BAC component is unaffected.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

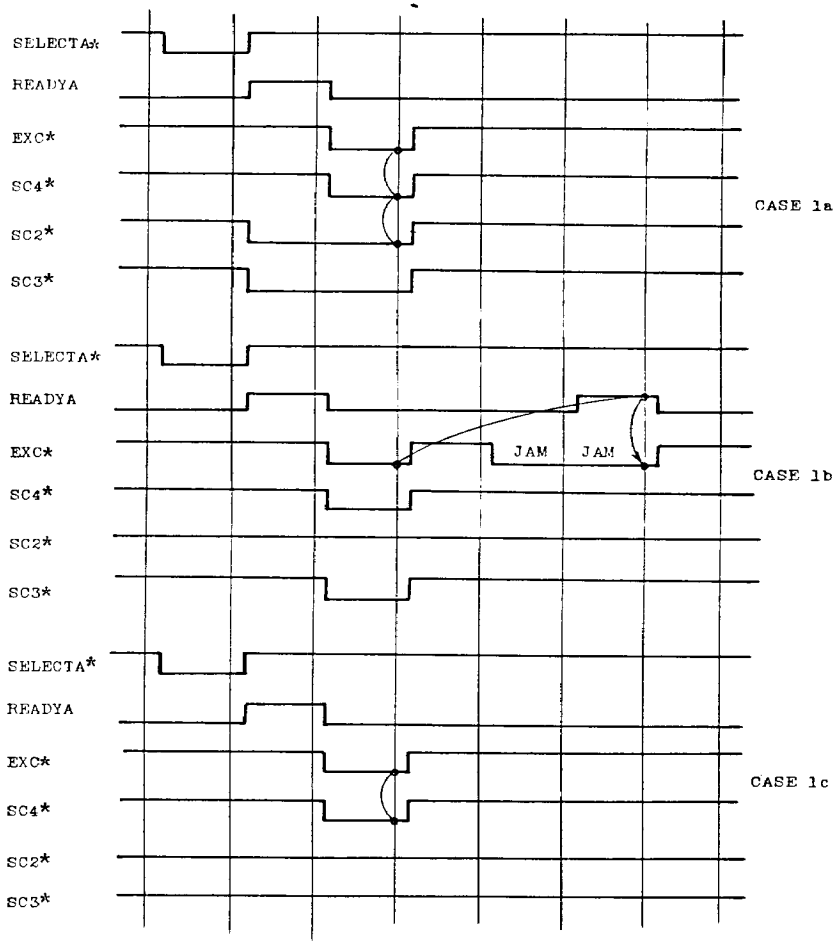


Figure 9A Bus Error Jam Protocol

Powered by ICminer.com Electronic-Library Service CopyRight 2003

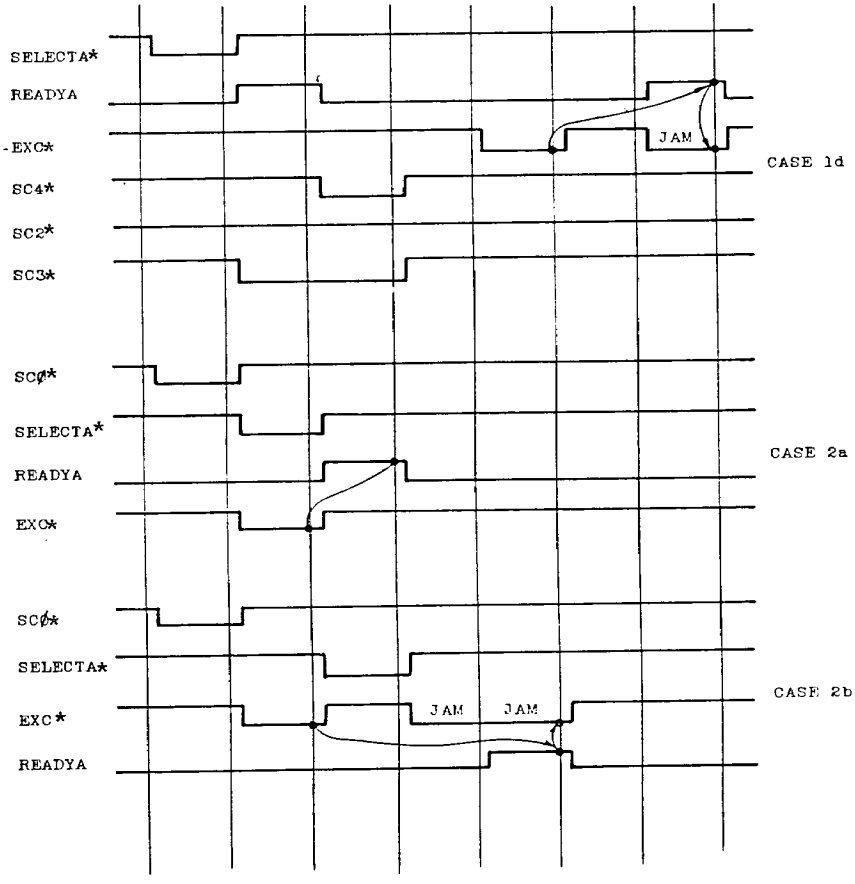


Figure 9B Bus Error Jam Protocol (CONT.)

Powered by ICminer.com Electronic-Library Service CopyRight 2003

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
VCC	VCC Supply Voltage with respect to Vss	-0.3V to 7V
VIN	Input Voltage	-0.3V to VCC+0.3V
TSTG	Storage Temperature	-65°C to 150°C
TOPR	Operating Temperature	0°C to 70°C

D.C. CHARACTERISTICS (BAC: TA = 0°C to 70°C, VCC = 5V ± 5%)

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
VIL	LOW level input voltage				0.8	V
VIH	High level input voltage		2.0			V
VOL	Low level output voltage				0.4	V
VOH	High level output voltage		2.4			V
ICC	Power Supply Current				100	mA
IIL Note1	Low level input current				20	uA
IIH Note1	High level input current				10	uA
IOZ Note1	Tri-state leakage current				10	uA
IOL Note2	Low level output current	VOL=0.4V	2.0			mA
IOH Note2	High level output current	VOH=2.4V	-1.5			mA
VZAP Note3	Electrostatic discharge		500			V

Note1: Direction of current may be out or in (source or sink).
For CLK,ARBIN0-ARBIN5,and PAR0-par3, IIL=400uA (with pullup)

Note2: Measured at TTL levels of 0.4V and 2.4V
EINT,DIREC,and DBERR are double buffered and have IOL=6.0mA and IOH=6.0mA

Note3: Minimum electrostatic discharge voltage on any pin per MIL-STD-883, Method 3015.

CAPACITANCE (BAC: TA = 0°C to + 70°C, VCC = 5V ± 5%)

SYMBOL	PARAMETER	TYP.	UNIT
COUT	OUTPUT CAPACITANCE	7	pF
CIN	INPUT CAPACITANCE EXEPT CLK	5	pF
CIN	CLK INPUT CAPACITANCE	10	pF

As a guideline, maximum capacitance of twice typical may be assumed.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

AC CHARACTERISTICS

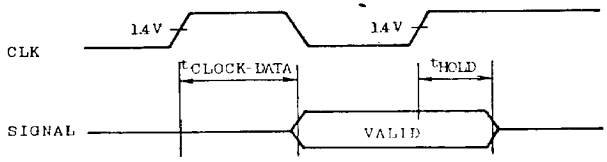


Figure 10 Driver Timing Parameters

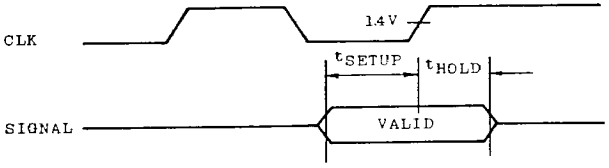


Figure 11 Receiver Timing Parameter

The following AC specifications for the BAC component are derated for commercial operating ranges of temperature and voltage.

A minimum on-board clock skew (for CLK at the chip inputs with respect to BCLK* on the iPSB bus) of -0.5ns and a maximum of 3.5ns is assumed. This includes the change in thresholds from the bus logic levels to the conventional TTL midpoint level of 1.4V for the clock signal.

The on-board loading of the BCLK* signal should be restricted to the test specifications of the AS804A-type devices (as shown in Figure 4). All timings are with respect to CLK (not BCLK) in nanoseconds and measured to 1.4V thresholds. For the S38-type devices (shown in Figure 4), a minimum delay of 2.5 ns and a maximum of 12.5 ns is assumed to 1.4V thresholds.

All output timings are specified at 50pF loadings with a few exceptions as noted below. Output loading should be restricted so as to guarantee that the skew time between TTL logic levels is 2ns or better for all BAC component outputs.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

(V_{CC} = 5V ± 5%, T_A = 0°C to 70°C)

NAME	t_{SU} MIN	t_{CD} MIN	t_{CD} MAX	t_{HOLD} MIN	UNIT	TEST CONDITIONS
			(Rise/			
BUS INTERFACE			Fall)			
ARBIN0*-ARBIN5*	22			0	ns	
Note 1					ns	
ARBOUT4		5.0	36		ns	
ARBOUT0		5.0	60		ns	
IBREQ*	22			0	ns	
OBREQ* Note 2		4.0	20		ns	
LACHN Note 3	28.5			2.5	ns	
TIMOUT Note 3	28.5			2.5	ns	
IBUSERR*	22			0	ns	
RESET Note 3	28.5			2.5	ns	
RSTNC*	22			0	ns	
OBUSERR Note 2		4.5	23.5		ns	
SC0*-SC9* Note 2	24	4.5	29	2.5	ns	
SCOEL-SCOEH Note 2		5.0	25/19.5		ns	
					ns	
AGENT INTERFACE					ns	
					ns	
GRANTA/B		5.0	33		ns	
OTHERREADY		5.0	33		ns	
LASTOUT		5.0	33		ns	
DBERR*		5.0	30		ns	
DIREC Note 2		6.0	27/28		ns	
REQUESTA/B	30			2	ns	
PRIORITY	30			2	ns	
READY A/B	40			2	ns	
WIDTH0*-WIDTH1*	30			2	ns	
SPACE0*-SPACE1*	30			2	ns	
WRITE*	30			2	ns	
BROADCAST	30			2	ns	
LASTINA*/B*	30			2	ns	
LOCK*	30			2	ns	
SELECTA*/B*	30			2	ns	
AGERR0-AGERR2	30			2	ns	
ADD0*-ADD1*	24			3.5	ns	
PAR0*-PAR3*	7			9	ns	

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Note 1: Maximum delay from ARBIN*(N) to ARBOUT(N-1) = 22ns
Minimum setup of ARBIN*(0) at end of arbitration phase = 30ns

Note 2: Load Capacitances:
For the following outputs of the BAC, the A.C. timings have been specified at reduced load capacitances:
15pF: ARBOUT0-ARBOUT4, SCOEL-SCOEH, OBUSERR, OBREQ*, SC8*-SC9*
30pF: SC0*-SC7*
Derating of all outputs increased load capacitance is as follows (Max 100pF):

	tpHL	tpLH
Single buffered outputs	0.15ns/pF	0.07ns/pF
Double buffered outputs	0.08ns/pF	0.034ns/pF
Triple buffered outputs	0.07ns/pF	0.016ns/pF

Derating figures are normalized over commercial operating ranges of temperature and voltage and subject to a maximum of 100 pF loadings.

Note 3: The following inputs to the BAC component are received from the iPSB bus through external inverters on the same chip so that the relative skew between them does not exceed 1.5ns: CLK, RESET. TIMOUT and LACHN.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

CONSIDERATIONS FOR INTERFACING WITH THE BAC COMPONENT

1. The following agent side inputs need to be synchronized external to the BAC component with the CLK input:

   REQUESTA/B, SELECTA*/B*, READYA/B

   Other agent inputs are qualified by these inputs and, therefore, may be activated in advance of these synchronized inputs.

2. The following inputs are shared between the primary and secondary agent interfaces:

   PRIORITY, AGERR0-AGERR2

   If these inputs are used by both interfaces in any application, external multiplexing is required. An OR gate is adequate to achieve the multiplexing of the AGERR0-AGERR2 lines if both agent interfaces ensure that these signals are kept inactive.

3. On becoming the bus owner, a primary or secondary agent has 3 cycles to assert SC0* on the iPSB bus and thus begin a transfer cycle. This translates to assserting the READYA/B input to the BAC component either in the same cycle or in the cycle after receiving the GRANTA/B as illustrated in Figures 6A and 6B respectively. Delaying the READYA/B beyond this point may result in loss of bus ownership. However, the BAC component does not disassert the GRANTA/B and continues to drive the lower SC* lines. Consequently, bus conflicts may result on the iPSB bus on the ADxx* and SCx* lines. When designing a single board computer, care should be taken to guarantee a READYA/B in response to a GRANTA/B within the permitted 2 cycles.

4. It is recommended that all message accesses be restricted to the secondary port to ensure consistency with the EINT logic.

5. The BAC component does not guarantee correct operation if a Requesting agent selects itself over the iPSB bus. Every agent that supports a replier needs to disable its decode/select logic during its own request phase. The GRANTA/B output may be used for this purpose.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

OUTLINE DRAWING

The Bus Arbiter/Controller is packaged in an 84-pin, PGA. Figure 12 illustrates the package and Figure 1 shows the pinout.

Unit in mm

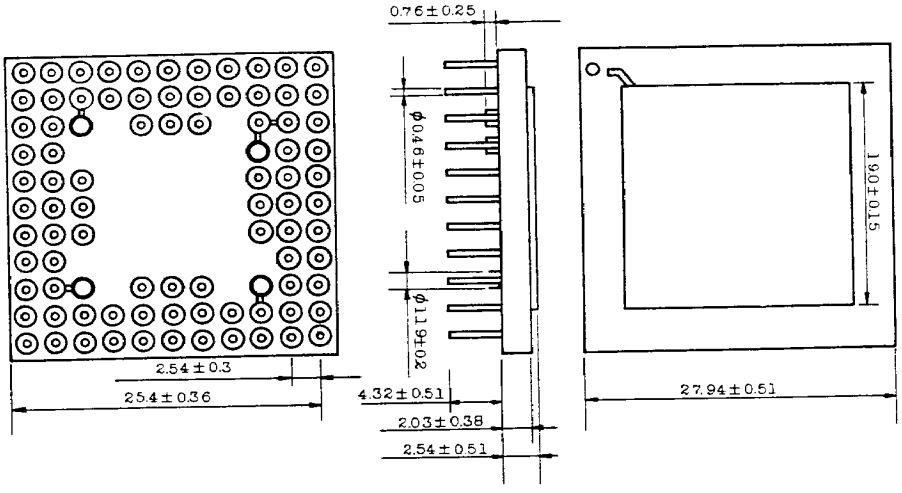


Figure 12 BAC Pin Grid Array Package Dimensions

Powered by ICminer.com Electronic-Library Service CopyRight 2003