- **Supports Provisions of IEEE 1394-1995 (1394) Standard for High-Performance Serial Bus†**
- **Fully Interoperable With FireWire™ Implementation of 1394**
- **Provides A Backplane Environment That Supports 50 or 100 Megabits per Second (Mbits/s)**
- **Logic Performs System Initialization and Arbitration Functions**
- **Encode and Decode Functions Included for Data-Strobe Bit-Level Encoding**
- **Incoming Data Resynchronized to Local Clock**
- **Separate Transmitter and Receiver for Greater Flexibility**
- **Data Interface to Link-Layer Controller (Link) Provided Through Two Parallel Signal Lines at 25/50 MHz**
- **100-MHz or 50-MHz Oscillator Provides Transmit, Receive-Data, and Link Clocks at 25/50 MHz**
- **Single 5-V Supply Operation**
- **Packaged in a High-Performance 64-Pin TQFP (PM) Package for 0°C to 70°C Operation and –40°C to 85°C Operation**
- **Packaged in a 68-Pin CFP (HV) Package for –55°C to 125°C Operation**

description

The TSB14C01A provides the transceiver functions needed to implement a single port node in a backplane-based 1394 network. The TSB14C01A provides two terminals for transmitting, two terminals for receiving, and a single terminal to externally control the drivers for data and strobe. The TSB14C01A is not designed to drive the backplane directly, this function must be provided externally. The TSB14C01A is designed to interface with a link-layer controller (link), such as the TSB12C01A.

The TSB14C01A requires an external 98.304-MHz or 49.152-MHz reference oscillator input for S100/50 operation. The reference signal is internally divided to provide the 49.152-MHz ±100-ppm system clock signals used to control transmission of the outbound encoded strobe and data information. The 49.152-MHz clock signal is supplied to the associated link for synchronization of the two chips. When this device is in the S100 mode of operation, OSC_SEL is asserted high. When the TSB14C01A is in the S50 mode of operation, the clock rate supplied to the link is 24.576 MHz.

Data bits to be transmitted are received from the link on two parallel paths and are latched internally in the TSB14C01A in synchronization with the 49.152-MHz system clock. These bits are combined serially, encoded, and then transmitted at 98.304-Mbits/s (in S100 mode) as the outbound data-strobe information stream. During transmission, the encoded data information is transmitted on TDATA, and the encoded strobe information is transmitted on TSTRB.

During packet reception the encoded information is received on RDATA and strobe information on RSTRB. The received data-strobe information is decoded to recover the receive clock signal and the serial data bits. The serial data bits are split into two parallel streams, resynchronized to the local system clock, and sent to the associated link.

The TSB14C01A is a 5-V device and provides CMOS-level outputs.

NOTE

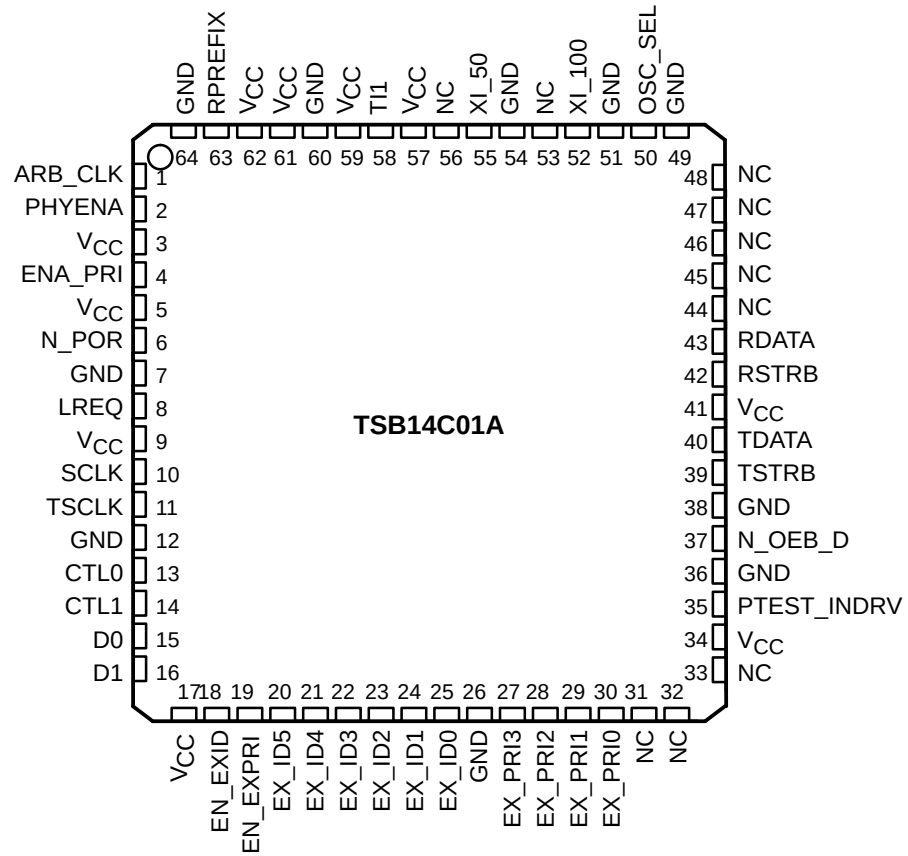
Designing with this device may require extensive support. Before incorporating this device into a design, customers should contact TI or an Authorized TI Distributor.

† This serial bus implements technology covered by one or more patents of Apple Computer, Incorporated and SGS Thomson, Limited. FireWire is a trademark of Apple Computer, Incorporated.

AVAILABLE OPTIONS

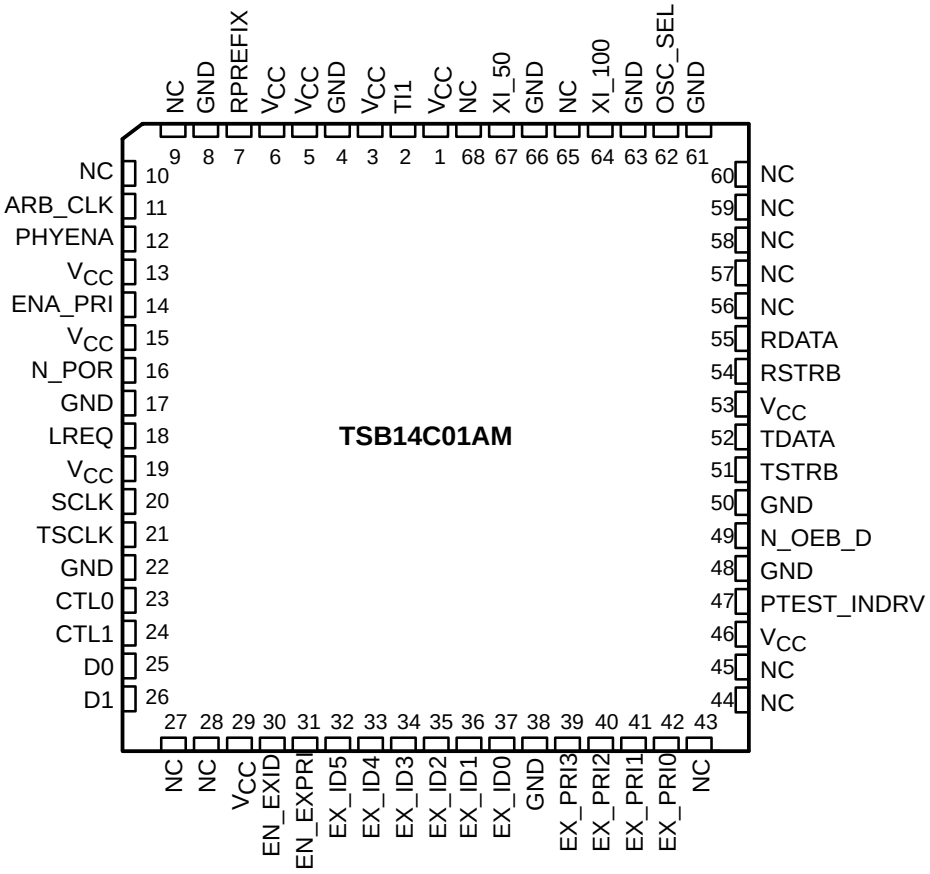
T_A	PACKAGES	
	CERAMIC FLAT PACK (HV)	THIN QUAD FLAT PACK (PM)
0°C to 70°C	—	TSB14C01APM
–40°C to 85°C	—	TSB14C01AIPM
–55°C to 125°C	TSB14C01AMHV	—

PM PACKAGE
(TOP VIEW)



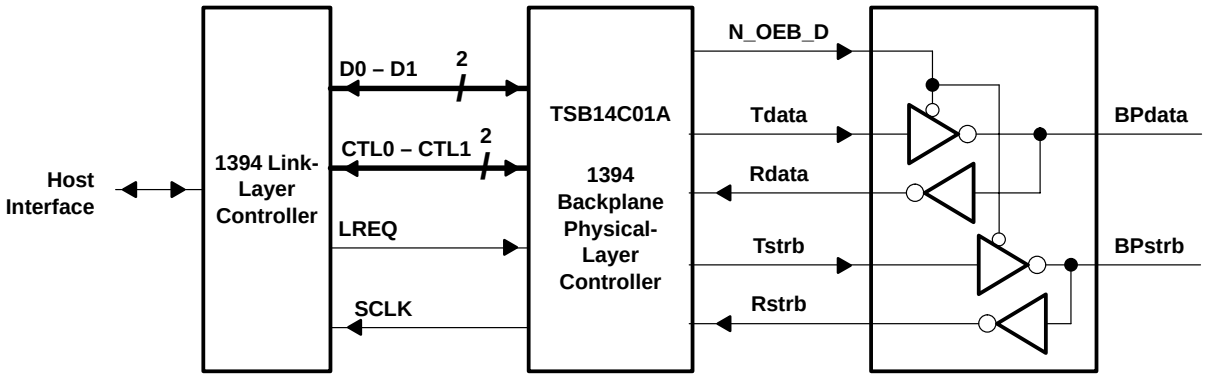
NC – No connection

HV PACKAGE
(TOP VIEW)



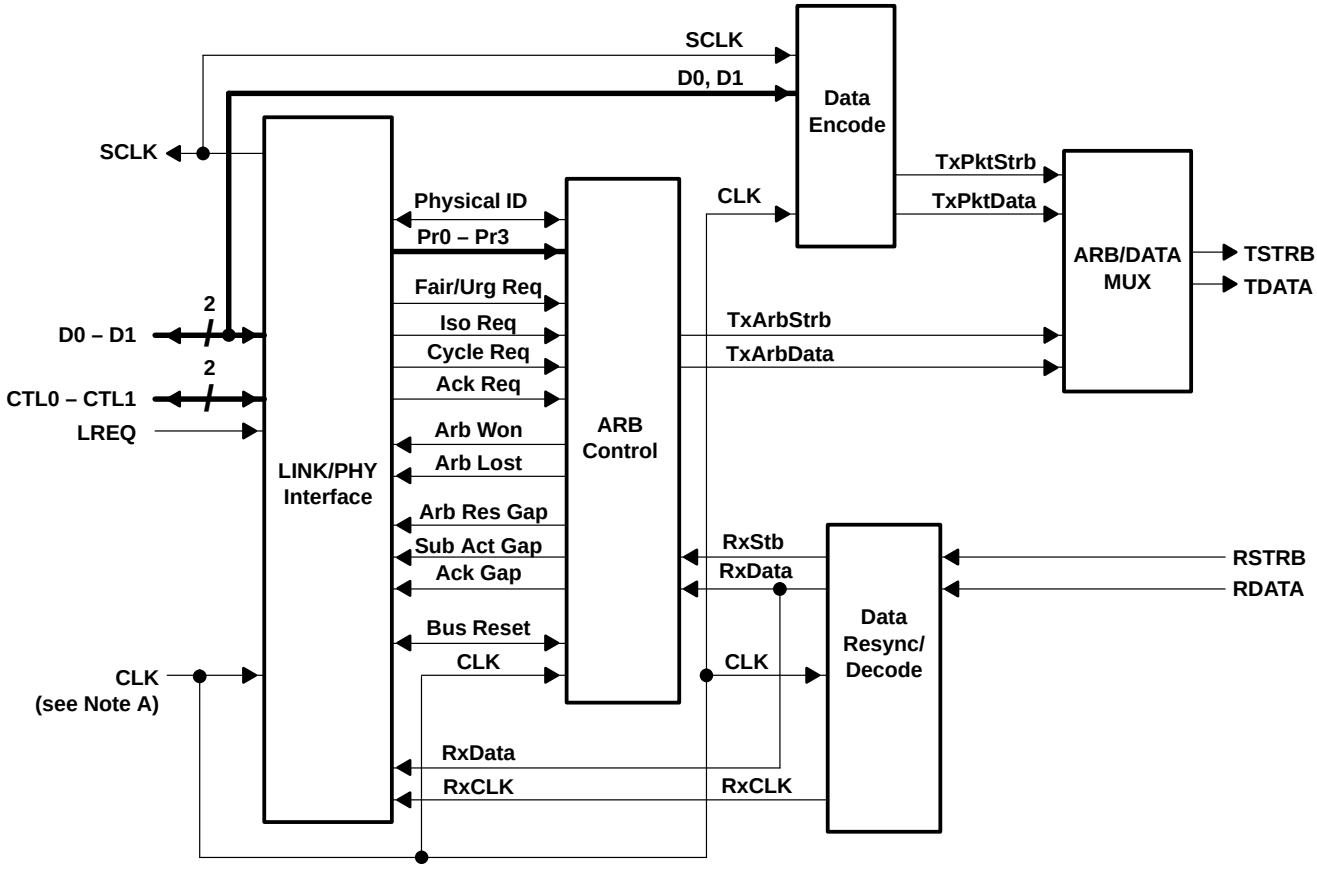
NC – No connection

system block diagram



NOTE A: The backplane transceiver is customer supplied and is different for each type of backplane.

functional block diagram



NOTE A: CLK is either terminal XI_50 or XI_100 depending on the mode selection.

Terminal Functions

TERMINAL NAME	TERMINAL PM NO.	TERMINAL HV NO.	TYPE	I/O	DESCRIPTION
ARB_CLK	1	11	TTL	O	Arbitration clock. ARB_CLK is the clock used for arbitration. ARB_CLK is for test and debug. It can be put into a high-impedance state by PTEST_INDRV. This terminal is not used in normal operation and is always at 49.152 MHz.
CTL0, CTL1	13, 14	23, 24	TTL	I/O	Control I/O. These are bidirectional signals that communicate between the TSB14C01A and the link that controls passage of information between the two devices.
D0, D1	15, 16	25, 26	TTL	I/O	Data I/O. These are bidirectional information signals that communicate between the TSB14C01A and the link layer.
ENA_PRI	4	14	TTL	I	Enable priority. ENA_PRI is tied low to enable the 7-bit bus request. See Table 1 for more information.
EN_EXID	18	30	TTL	I	Enable external ID. When EN_EXID is asserted high, the ID for this node is set externally by EX_ID. When this terminal is tied/driven low, the source of the ID comes from the internal ID register.
EN_EXPRI	19	31	TTL	I	Enable external priority. When EN_EXPRI is asserted high (external priority enabled) the priority level for this node is set externally (see Table 1). This terminal should be tied low when not used.
EX_ID5 – EX_ID0	20,21,22, 23,24,25	32,33,34, 35,36,37	TTL	I	External ID. The ID for this node is determined by the value on the EX_ID terminals. Bit 0 is the MSB.
EX_PRI3 – EX_PRI0	27,28, 29,30	39,40, 41,42	TTL	I	External priority. The priority for this node is determined by the values on the EX_PRI terminals. See Table 1 for more information.
GND	7,12,26, 36,38,49, 51,54,60, 64	4,8,17, 22,38,48, 50,61,63, 66	Supply	—	Circuit ground
LREQ	8	18	TTL	I	Link request input. LREQ is an input from the link used by the link to signal the TSB14C01A of a request to perform some service.
V_{CC}	3,5,9,17, 34,41,57, 59,61,62	1,3,5,6, 13,15,19, 29,46,53	Supply	—	Circuit power
NC	31,32,33, 44,45,46, 47,48,53, 56	9,10,27, 28,43–45, 56–60, 65,68	—	—	Not connected. These terminals must be left floating.
N_OEB_D	37	49	TTL	O	External driver enable. N_OEB_D is a negative active signal that enables the external driver for TDATA and TSTRB.
N_POR	6	16	TTL	I	Logic reset input . Forcing N_POR low causes a reset condition and resets the internal logic to the reset start state.
OSC_SEL	50	62	VCC / GND	I	Select clock frequency. OSC_SEL should be pulled up to V_{CC} when the operating frequency is 50 MHz. When the operating frequency is 100 MHz then it should be pulled to ground. It should not be left floating.
PHYENA	2	12	TTL	O	Phy enable. When the phy is driving it is low, PHYENA is the control to the CTL0, CTL1, D0, and D1 drivers. PHYENA is for test and debug. It can be put into a high-impedance state by PTEST_INDRV. This terminal is not used in normal operation.
PTEST_INDRV	35	47	TTL	I	Test output enable. PTEST_INDRV enables/disables the drivers to the test terminals ARB_CLK, PHYENA, and RPREFIX. During normal operation, PTEST_INDRV should be tied to V_{CC} to disable the drivers.
RDATA	43	55	TTL	I	Receive data. Incoming data is received at the data rate.

Terminal Functions (continued)

TERMINAL NAME	TERMINAL PM NO.	TERMINAL HV NO.	TYPE	I/O	DESCRIPTION
RPREFIX	63	7	TTL	O	Receiver prefix. When asserted high (enabled), RPREFIX alerts the receiver of an incoming packet. RPREFIX is for test and debug and is not used in normal operation.
RSTRB	42	54	TTL	I	Receive strobe. RSTRB decodes the received data.
SCLK	10	20	TTL	O	System clock output. A 49.152-MHz or 24.576-MHz clock signal synchronized with the data transfers, and provided to the link.
TDATA	40	52	CMOS	O	Transmit data. Data to be transmitted is serialized on TDATA.
TI1	58	2	TTL	I	Test input 1. TI1 is used for test purposes only and should be tied to ground for normal operation.
TSCLK	11	21	TTL	O	System clock output. A 49.152-MHz or 24.576-MHz clock signal that is 180 degrees out of phase with SCLK; it is available if needed.
TSTRB	39	51	CMOS	O	Transmit strobe. TSTRB encodes transmit data.
XI_50	55	67	CMOS	I	External oscillator input. An external 49.152-MHz oscillator can drive the TSB14C01A.
XI_100	52	64	CMOS	I	External oscillator input. An external 98.304-MHz oscillator can drive the TSB14C01A.

Table 1. External Priority Coding

EXTERNAL PRIORITY TERMINALS			DESCRIPTION
ENA_PRI	EN_EXPRI	EX_PRI0 – EX_PRI3	
X	H	L L L L L L L H . . . H H H H	The priority for this node is determined by the values on EX_PRI0 – EX_PRI3. The state of ENA_PRI can be either tied to V_{CC} or GND.
L	L	X	This sets a 7-bit bus request and is the cable environment format. Priority must be set by executing a write request (see Table 10).
H	L	X	This sets an 11-bit bus request and is the backplane environment format. Priority is dynamically set as part of the bus request (see Table 8)

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage range, V_{CC} . . . . . –0.5 V to 6.0 V
Input voltage range, V_I . . . . . –0.5 V to V_{CC} + 0.5 V
Output voltage range at any output, V_O . . . . . –0.5 V to V_{CC} + 0.5 V
Continuous total power dissipation . . . . . See Dissipation Rating Table
Operating free air temperature, T_A: TSB14C01A . . . . . 0°C to 70°C
TSB14C01AI . . . . . –40°C to 85°C
TSB14C01AM . . . . . –55°C to 125°C
Storage temperature range, T_{stg} . . . . . –65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds . . . . . 300°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

DISSIPATION RATING TABLE

PACKAGE	$T_A \le 25°C$ POWER RATING	DERATING FACTOR‡ ABOVE $T_A = 25°C$	$T_A = 70°C$ POWER RATING	$T_A = 85°C$ POWER RATING	$T_A = 125°C$ POWER RATING
HV	1689 mW	13.5 mW/°C	1081 mW	879 mW	337 mW
PM	1602 mW	12.8 mW/°C	1025 mW	833 mW	—

‡ This is the inverse of the traditional junction-to-ambient thermal resistance ($R_{\theta JA}$) and uses a board-mounted $R_{\theta JA}$ of 78°C/W for the PM package and 74°C/W for the HV package.

recommended operating conditions

		MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}		4.5	5	5.25	V
High-level input voltage, V_{IH}	CMOS inputs	0.7 V_{CC}			V
	TTL input	2		V_{CC}	V
Low-level input voltage, V_{IL}	CMOS inputs	0		0.2 V_{CC}	V
	TTL input	0		0.8	V
Input voltage, V_I	CMOS/TTL	0		V_{CC}	V
High-level output current, I_{OH}	CMOS Drivers			12	mA
	TTL Drivers			8	
Low-level output current, I_{OL}	CMOS Drivers			24	mA
	TTL Drivers			8	
Virtual Junction Temperature, T_J §	TSB14C01A			115	°C
	TSB14C01AI			125	
	TSB14C01AM			150	
Operating Free-Air Temperature, T_A	TSB14C01A	0		70	°C
	TSB14C01AI	–40		85	
	TSB14C01AM	–55		125	

§ Actual junction temperature is a function of ambient temperature, package selection, power dissipation, and air flow. Customer is responsible for maintaining the junction temperature within the recommended operating conditions. Operating device at junction temperatures higher than what is recommended will cause device to operate outside the characterization models established during device simulation and may affect the reliability performance.

electrical characteristics over recommended ranges of operating conditions (unless otherwise noted)

device

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	I_{OH} = max, V_{CC} = min	V_{CC}–0.8			V
V_{OL}	Low-level output voltage	I_{OL} = min, V_{CC} = max			0.5	V
	Input current	$V_I = V_{CC}$ or 0			±1	μA
I_{OZ}	Off-state output current	$V_I = V_{CC}$ or 0			±10	μA

thermal characteristics†

PARAMETER			TEST CONDITION	MIN	TYP	MAX	UNIT
$R_{\theta JA}$	Junction-to-free-air thermal resistance	TSB14C01APM, TSB14C01AIPM	Board mounted, No air flow		78		°C/W
		TSB14C01AMHV			74		°C/W
$R_{\theta JC}$	Junction-to-case thermal resistance	TSB14C01APM, TSB14C01AIPM			16		°C/W
		TSB14C01AMHV			3		°C/W

† Thermal characteristics vary depending on die, leadframe, pad size, and mold compound. These values represent typical die and pad sizes for the respective package. The $R_{\theta JA}$ value decreases as the die pad size increases. Thermal values represent PWB bands with minimal amount of metal.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (See Note 1)

PARAMETER		MEASURED	TEST CONDITION	MIN	TYP	MAX	UNIT
t_{su}	D, CTL, LREQ low or high before SCLK high	50% to 50%	See Figure 1	7			ns
t_h	D, CTL, LREQ low or high after SCLK high	50% to 50%	See Figure 1	1			ns
t_d	Delay time, SCLK high to D, CTL high or low	50% to 50%	See Figure 2			10	ns

NOTE 1: These parameters are ensured by design and are not production tested.

PARAMETER MEASUREMENT INFORMATION

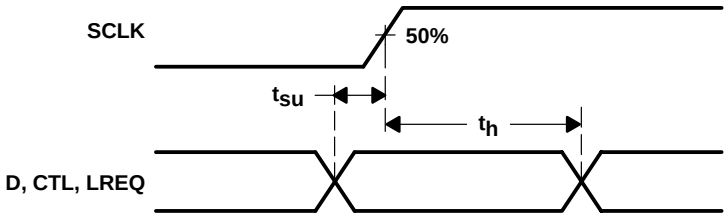


Figure 1. D, CTL, LREQ Input Setup and Hold Timing Waveforms

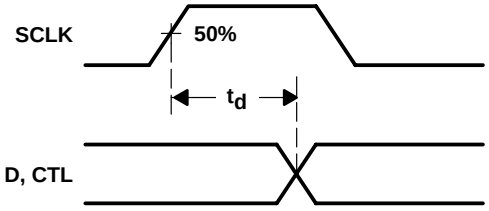


Figure 2. D, CTL Output Delay Relative to SCLK Timing Waveforms

APPLICATION INFORMATION

internal register configuration

The accessible internal registers of this device are listed in Table 2. Bit field descriptions for the registers are given in Table 3.

Table 2. Format for Registers

ADDRESS	0	1	2	3	4	5	6	7
0000	Physical ID[0]	Physical ID[1]	Physical ID[2]	Physical ID[3]	Physical ID[4]	Physical ID[5]	Reserved	Reserved
0001	INHB	IBR	RESERVED					
0011	RESERVED							
0100	Priority level[0]	Priority level[1]	Priority level[2]	Priority level[3]	RESERVED			

Table 3. Register Bit Field Key

FIELD	SIZE (Bits)	TYPE	DESCRIPTION
Physical ID	6	Read/Write	Physical identification. Physical ID is the address of the local node and is set to zero on power up.
INHB	1	Read/Write	Inhibit Drivers. INHB is used to turn off the drivers to TDATA and TSTRB.
IBR	1	Read/Write	Initiate Bus Reset. IBR is turned on by the link and turned off by the phy when reset is complete.
Priority	4	Read/Write	Priority setting. The four bits contain the priority of the local node. A higher value in this field indicates a higher priority.

transceiver selection

The system designer must select transceivers appropriate to the system requirements to be used with the TSB14C01A and the link layer selected. The following lists requirements for the transceivers needed.

- The transceivers used must be appropriate to the backplane technology used.

  The various backplane technologies require different electrical characteristics in their backplanes. For example BTL uses an operating voltage on the backplane of 2.1 V and a characteristic impedance of 33 Ω while GTL uses an operating voltage of 1.2 V and a characteristic impedance of 50 Ω (see *GTL/BTL a Low Swing Solution for High-Speed Digital Logic*, TI literature number SCEA003). When a backplane is designed to use BTL technology, then it would be appropriate to also use that technology for the two lines dedicated to the 1394 serial bus. The drivers selected also must be able to supply the current required for the expected backplane loading. For example, BTL operates correctly for a FutureBus configuration backplane at 50 Mbits/s or for a limited number of nodes in a custom configuration at 100 Mbits/s. See the *GTL/BTL a Low Swing Solution for High-Speed Digital Logic*, TI literature number SCEA003, *Understanding Advanced Bus-Interface Products*, TI literature number SCAA029, or the documentation for the transceiver being considered.

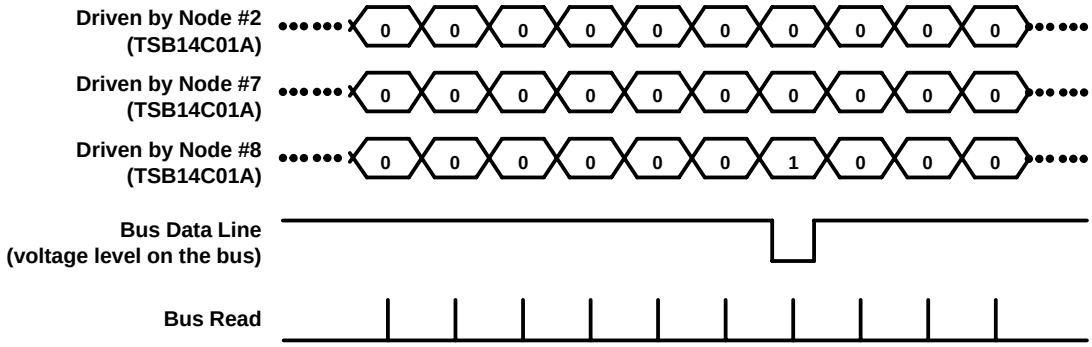
- The transceivers used must assert logic states on the backplane in an appropriate manner for the 1394 backplane arbitration.

  Arbitration under 1394 backplane rules requires the drivers to assert the bus to indicate a logical 1 state, that is a logic 1 being driven by the TSB14C01A. Conversely, the drivers should release the bus to indicate a logic 0 state, a logic 0 being driven by the TSB14C01A. In other words, all drivers must operate in a wired-OR mode during arbitration.

- The transceivers used must be able to monitor the bus and drive the bus at the same time.

During arbitration, each node that is arbitrating for the bus drives its priority code then its node number out onto the bus. During each bit period, each node reads back what has been placed on the bus. If it reads back the same data it was sending, the arbitrating node stays in contention for winning the bus. If it reads something different than what it was driving, the arbitrating node loses the bus and drops out of contention. This is the reason for requiring wired-OR operation during arbitration. As long as each node is still sending 0s onto the bus during arbitration, all nodes are still contending to win the bus. The node with the highest priority (or if all priorities were zero then the highest node number) is the first to drive a 1 onto the bus during arbitration. The node that sends the first 1 (asserting the bus) and reads it back wins the bus. All other nodes read back a 1, which does not match the 0 (releasing the bus) they are sending, and drop out of contention. This arbitration process requires the transceiver selected to be able to read from the bus at the same time it is driving the bus.

For example, if three nodes, each with priority 0 and a node identifiers of 8, 7, and 2, were to arbitrate for the bus, the following would occur (see Figure 3):



NOTE A: This bus is reverse logic, a 1 being driven by the TSB14C01A is asserted by driving a 0 onto the bus by the transceiver.

Figure 3. Three Nodes Arbitrating for the Bus

Since the highest node number is 8 (1000b), node 8 outputs the first 1 (assert the bus) and wins the arbitration. The other nodes drop out and do not try to drive their node number onto the bus.

- The transceivers used must be appropriate for the transfer speed required.

  The 1394 bus has two data lines that use data-strobe encoding on the bus. This requires that the transceivers be able to operate at a maximum frequency of one half of the maximum data transfer rate. When operating at 50 Mbits/s, the maximum frequency the drivers are required to operate at is 25 MHz. When operating at 100 Mbits/s, the maximum frequency the drivers are required to operate at is 50 MHz.

PRINCIPLES OF OPERATION

The TSB14C01A (phy) is designed to operate with a link such as the Texas Instruments TSB12C01A. These devices use an interface such as described in Annex J of the IEEE 1394-1995 standard. Details of how the TSB12C01A device operates are given in the TSB12C01A data manual (literature number SLLS219). For information on the operation of 1394, see the IEEE 1394-1995 standard. For more specific information on the backplane phy, see the following sections of IEEE 1394-1995 standard: Chapter 5, *Backplane physical layer specification*; Annex D, *Backplane physical layer timing*; Annex F, *Backplane physical implementation example*; Annex G, *Backplane isochronous resource manager selection*. The following paragraphs describe the operation of the phy-link interface.

The TSB14C01A supports 100 Mbit/s data transfers and has two bidirectional data lines (D0 and D1) crossing the interface. In addition there are two bidirectional control lines (CTL0 and CTL1), the SCLK line from the phy to the link, and the link request line (LREQ) from the link to the phy. The TSB14C01A phy has control of all the bidirectional terminals. The link is allowed to drive these terminals only after it has been given permission by the phy. The dedicated LREQ request terminal is used by the link for any activity that it can initiate.

There are four basic operations that can occur in the interface: request, status, transmit, and receive. All operations but request are initiated by the phy. The link uses the request operation to read or write an internal phy register or to ask the phy to initiate a transmit action. The phy initiates a receive action whenever a packet is received from the serial bus.

When the phy has control of the bus, the CTL0 and CTL1 lines are encoded as shown in Table 4.

Table 4. CTL0 and CTL1 Control Lines When the Phy Has Control

CTL0	CTL1	Name	Description of Activity
0	0	Idle	No activity is occurring (this is the default mode).
0	1	Status	Status information is being sent from the phy to the link.
1	0	Receive	An incoming packet is being sent from the phy to the link.
1	1	Transmit	The link has been given control of the bus to send an outgoing packet.

When the link has control of the bus (only with phy permission) the CTL0 and CTL1 lines are encoded as shown in Table 5.

Table 5. CTL0 and CTL1 Control Lines When the Link Has Control

CTL0	CTL1	Name	Description of Activity
0	0	Idle	The link releases the bus (the transmission has been completed).
0	1	Hold	The link is holding the bus while data is being prepared for transmission or sending another packet without arbitrating.
1	0	Transmit	An outgoing packet is being sent from the link to the phy.
1	1	Reserved	None

When the link needs to request the bus or access a register that is located in the TSB14C01A phy, a serial stream of information is sent across the LREQ line. The length of the stream varies depending on whether the transfer is a bus request, a read command, or a write command (see Table 6). Regardless of the type of transfer, a start bit of 1 is required at the beginning of the stream, and a stop bit of 0 is required at the end of the stream. Bit 0 is the MSB, and is transmitted first. The LREQ line is required to idle low (logic level 0).

PRINCIPLES OF OPERATION

Table 6. Request Bit Length

Request Type	Number of Bits
Bus request (cable)	7
Bus request (backplane)	11
Read register request	9
Write register request	17

For a bus request in the cable environment, the length of the LREQ data stream is 7 bits as shown in Table 7.

Table 7. Bus Request For Cable Environment

Bit(s)	Name	Description
0	Start bit	Indicates the beginning of the transfer (always 1).
1-3	Request type	Indicates the type of bus request (see Table 11 for the encoding of this field).
4-5	Request speed	Indicates the speed at which the phy sends the packet for this request. This field has the same encoding as the speed code from the first symbol of the receive packet. See Table 8 for the encoding of this field. This field can be expanded to support data rates higher than 400 Mbit/s in the future.
6	Stop bit	Indicates the end of the transfer (always 0).

When LREQ transfer is a bus request in the backplane environment, it is 11 bits long and has the format shown in Table 8.

Table 8. Bus Request For Backplane Environment

Bit(s)	Name	Description
0	Start bit	Indicates the beginning of the transfer (always 1).
1-3	Request type	Indicates the type of bus request (see Table 11 for the encoding of this field).
4-5	Request speed	Ignored (set to 00) for the backplane environment
6-9	Request priority	Indicates priority of urgent requests. It is only used with a FairReq request type. All zeros indicates fair request. All ones is reserved (this priority is implied by a PriReq). Other values are used to indicate the priority of an urgent request.
10	Stop bit	Indicates the end of the transfer (always 0).

For a Read Register Request the length of the LREQ data stream is 9 bits as shown in Table 9 (also see Table 3 for the bit definitions).

Table 9. Read Request Format

Bit(s)	Name	Description
0	Start bit	Indicates the beginning of the transfer (always 1).
1-3	Request type	Always a 100 indicating that this is a read register request.
4-7	Address	Is the address of the phy register to be read.
8	Stop bit	Indicates the end of the transfer (always 0).

PRINCIPLES OF OPERATION

For a Write Register Request the length of the LREQ data stream is 17 bits as shown in Table 10 (see Table 2 for the bit field format).

Table 10. Write Request Format

Bit(s)	Name	Description
0	Start bit	Indicates the beginning of the transfer (always 1).
1-3	Request type	Always a 101 indicating that this is a write register request.
4-7	Address	The address of the phy register to be written to.
8-15	Data	The data that is to be written to the specified register address.
16	Stop bit	Indicates the end of the transfer (always 0).

The 3-bit request field is defined in Table 11.

Table 11. Request Field

LREQ1	LREQ2	LREQ3	Name	Description
0	0	0	ImmReq	Immediate request. When an idle is detected, the phy takes control of the bus immediately (no arbitration)
0	0	1	IsoReq	Isochronous request. When an acknowledge gap is detected, the phy arbitrates for the bus.
0	1	0	PriReq	Priority request. The phy arbitrates after a subaction gap and ignore fair protocol. PriReq is used for cycle master request.
0	1	1	Fair/Urgent Req	Fair or urgent request. The phy arbitrates after a subaction gap using fair protocol. Fair/Urgent Req is used for fair transfers with the request priority field differentiating fair and urgent transfers for the backplane environment.
1	0	0	RdReg	Read register. Returns the specified register contents through a status transfer
1	0	1	WrReg	Write register. Writes to the specified register
1 1	1 1	0 1	Reserved	Reserved

LREQ Timing (each cell represents one clock sample time) is shown in Figure 4.

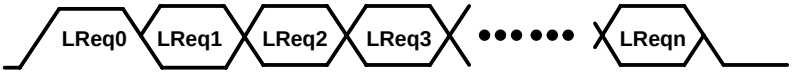


Figure 4. LREQ Timing

PRINCIPLES OF OPERATION

bus request

For fair or priority access, the link requests control of the bus at least one clock after the phy-link interface becomes idle. When the link senses that the CTL terminals are in a receive state (CTL 0 and CTL1 = 10), then it knows that its request has been lost. This is true anytime during or after the link sends the bus request transfer. Additionally, the phy ignores any fair or priority requests if it asserts the receive state while the link is requesting the bus. The link then reissues the request one clock after the next interface idle.

The cycle master node uses a priority request (PriReq) to send the cycle start message. To request the bus to send isochronous data, the link can issue the request at any time after receiving the cycle start. The phy clears an isochronous request only when the bus has been won.

To send an acknowledge, the link must issue an ImmReq request during the reception of the packet addressed to it. This is required because the delay from end-of-packet to acknowledge request adds directly to the minimum delay every phy must wait after every packet to allow an acknowledge to occur. After the packet ends, the phy immediately takes control of the bus and grants the bus to the link. If the header cyclic redundancy check (CRC) of the packet turns out to be bad, the link releases the bus immediately. The link cannot use this grant to send another type of packet. To ensure this, the link must wait 160 ns after the end of the received packet to allow the phy to grant it the bus for the acknowledgement, then releases the bus and proceeds with another request.

Although highly improbable, it is conceivable that two separate nodes can believe that an incoming packet is intended for them. The nodes then issue an ImmReq request before checking the CRC of the packet. Since both phys seize control of the bus at the same time, a temporary, localized collision of the bus occurs somewhere between the competing nodes. This collision would be interpreted by the other nodes on the network as being a ZZ line state, not a bus reset. As soon as the two nodes check the CRC, the mistaken node drops its request and the false line state is removed. The only side effect would be the loss of the intended acknowledgment packet (this is handled by the higher-layer protocol).

Once the link issues an immediate, isochronous, fair, or priority request for access to the bus, it cannot issue another request until the phy indicates a lost (incoming packet) or won (transmit) signal. The phy ignores new requests while a previous request is pending.

read/write requests

For write requests, the phy takes the value in the data field (see Table 2) of the transfer and loads it into the addressed register as soon as the transfer is complete. For read requests (see Table 2), the phy returns the contents of the addressed register at the next opportunity through a status transfer. The link is allowed to perform a read or write operation at any time. When the status transfer is interrupted by an incoming packet, the phy continues to attempt the transfer of the requested register until it is successful.

PRINCIPLES OF OPERATION

status

When the phy has status information to transfer to the link, it initiates a status transfer. The phy waits until the interface is idle to perform the transfer. The phy initiates the transfer by asserting status (01) on the CTL terminals, along with the first two bits of status information on D0 and D1. The phy maintains CTL == status for the duration of the status transfer. The phy can temporarily halt a status transfer by asserting something other than status on the CTL terminals. This is done in the event that a packet arrives before the status transfer completes. There must be at least one idle cycle between consecutive status transfers.

The phy normally sends only the first 4 bits of status to the link. These bits are status flags that are needed by link state machines. The phy sends an entire status packet to the link after a request transfer that contains a read request.

The definition of the bits in the status transfer are shown in Table 12 (also see Table 2 and Table 9). The 16-bit status stream is defined in Table 12.

Table 12. Status Bit Description

Bit(s)	Name	Description
0	Arbitration Reset gap	Indicates that the phy has detected that the bus has been idle for an arbitration reset gap time (this time is defined in the 1394 standard). This bit is used by the link in its busy/retry state machine.
1	Subaction gap	Indicates that the phy has detected that the bus has been idle for a subaction gap time (this time is defined in the 1394 standard). This bit is used by the link to detect the completion of an isochronous cycle.
2	Bus reset	Indicates that the phy has entered the bus reset state.
3	Reserved	Reserved
4-7	Address	Holds the address of the phy register whose contents are transferred to the link.
8-15	Data	Indicates the data that is to be sent to the link.

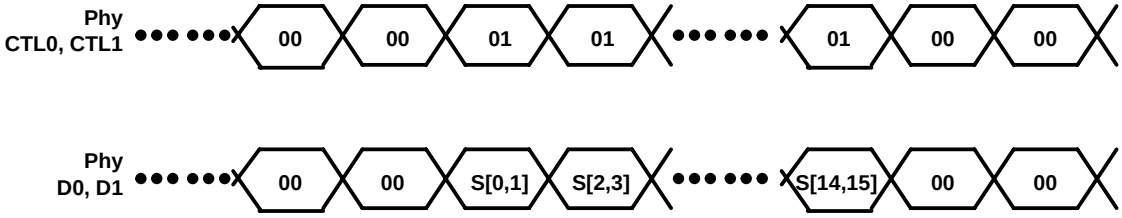


Figure 5. Status Transfer Timing

PRINCIPLES OF OPERATION

transmit

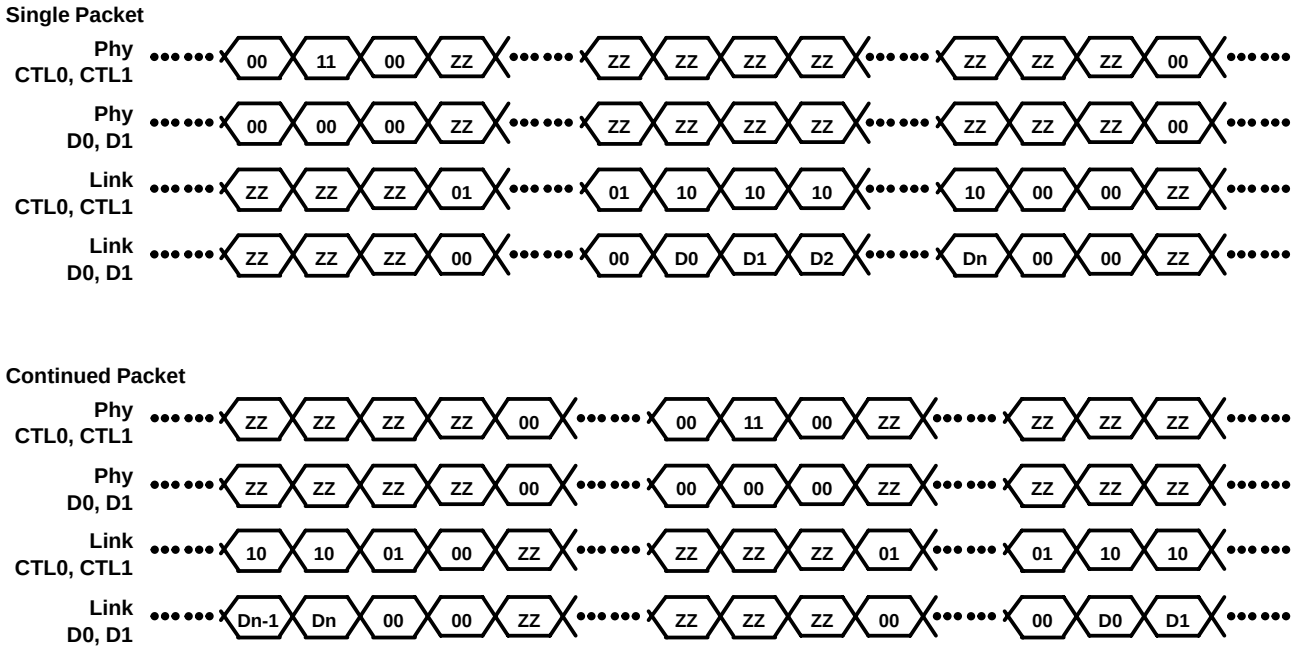
When the link requests access to the serial bus through the LREQ terminal, the phy arbitrates for access to the serial bus. When the phy wins the arbitration, it grants the bus to the link by asserting transmit on the CTL terminals for one SCLK cycle, followed by idle for one cycle. After sampling the transmit state from the phy, the link takes over control of the interface by asserting either hold or transmit on the CTL terminals. The link asserts hold to keep ownership of the bus while preparing data. The phy asserts the data-on state on the serial bus during this time. When it is ready to begin transmitting a packet, the link asserts transmit on the CTL terminals along with the first bits of the packet. After sending the last bits of the packet, the link asserts either idle or hold on the CTL terminals for one cycle, and then idle for one additional cycle, before asserting those terminals to a high-impedance state.

The hold state here indicates to the phy that the link needs to send another packet without releasing the bus. The phy responds to this hold state by waiting the required minimum time and then asserting transmit as before. This function would be used after sending an acknowledgement if the link intends to send a unified response, or to send consecutive isochronous packets during a single cycle. The only requirement when sending multiple packets during a single bus ownership is that all packets must be transmitted at the same speed, since the speed of the packet transmission is set before the first packet.

As noted above, when the link has finished sending the last packet for the current bus ownership, it releases the bus by asserting idle on the CTL terminals for two SCLK cycles. The phy begins asserting idle on the CTL terminals one clock after sampling the second idle from the link. Note that whenever the D and CTL lines change ownership between the phy and the link, there is an extra clock period allowed so that both sides of the interface can operate on registered versions of the interface signals, rather than having to respond to a CTL state on the next cycle.

It is not required that the link enter the hold state before sending the first packet when implementation permits the link to be ready to transmit as soon as bus ownership is granted. The timing for a single packet transmit operation is shown in Figure 6. In the diagram, D0 – Dn are the data symbols of the packet; ZZ represents the high-impedance state.

transmit timing



NOTE A: ZZ = High Impedance State
D0 = > Dn = Packet data

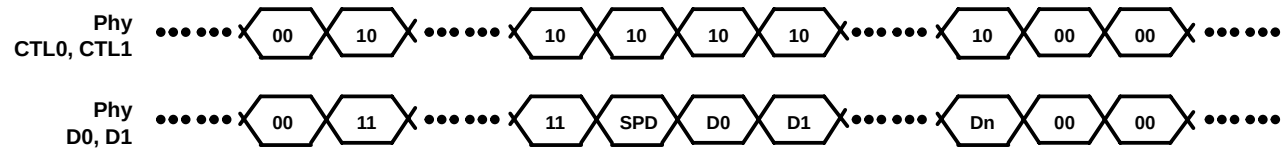
Figure 6. Transmit Timing

receive

When data is received by the phy from the serial bus, it transfers the data to the link for further processing. The phy asserts receive (see Table 4) on the CTL lines and asserts each D terminal high. The phy indicates the start of the packet by placing the speed code on the data bus. The phy then proceeds with the transmittal of the packet to the link on the D lines while still keeping the receive status on the CTL terminals. Once the packet has been completely transferred, the phy asserts idle on the CTL terminals, which completes the receive operation (see Figure 7).

NOTE

The speed code is a phy-link protocol and is not included in the CRC.



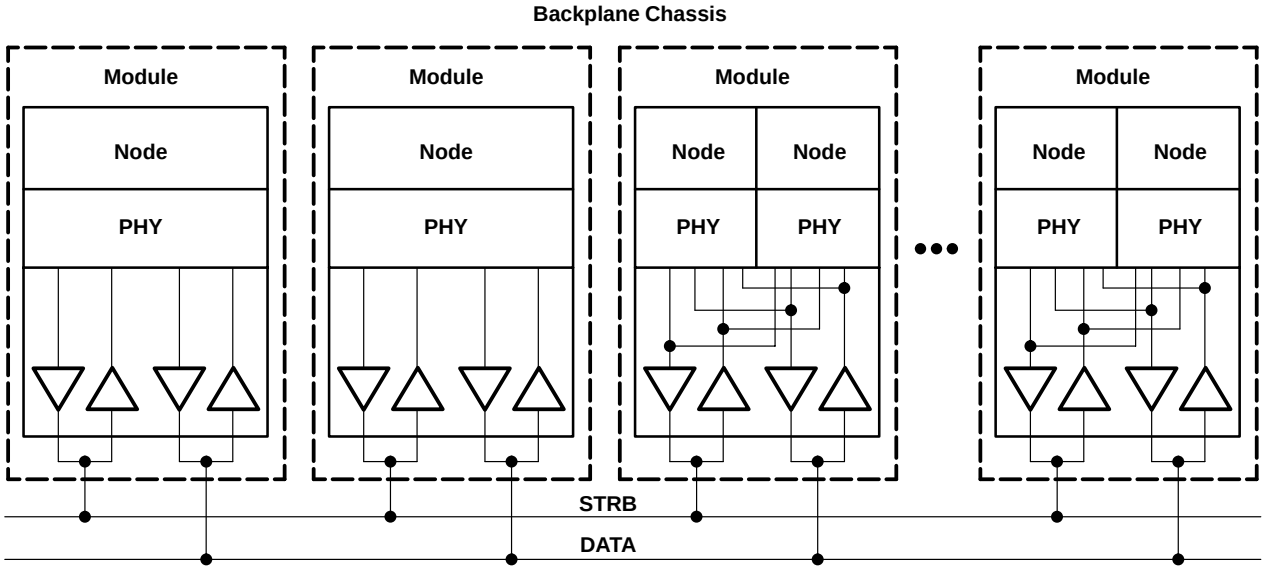
NOTE A: SPD = Speed Code (For the backplane, this speed is fixed at D0, D1 = 00).
D0 to Dn = Packet data

Figure 7. Receive Timing

PRINCIPLES OF OPERATION

backplane phy connection

Within the backplane environment the serial bus is implemented with a pair of signals (STRB and DATA). The topology is a simple pair of bussed signals as shown in Figure 8.



NOTE A: On a given bus, there can be as many as 63 nodes. There is no restriction on the distribution of nodes throughout modules on the bus. When more than one node occupies a module, they must share the same transceivers.

Figure 8. Backplane Topology

The backplane environment can be implemented with a number of different interface technologies. These include, but are not limited to: TTL for industry-standard transistor-transistor logic, BTL for backplane transceiver logic as defined by IEEE Std 1194.1, and ECL for emitter-coupled logic.

In addition to the requirements specified by the application environment, the physical media of the serial bus should meet the requirements defined for media attachment, media signal interface, and media signal timing. Timing requirements must be met over the ranges specified in the application environment. These include temperature ranges, voltage ranges, and manufacturing tolerances.

PRINCIPLES OF OPERATION

definition of logic states

Drivers assert the bus to indicate a 1 logic state, or release the bus to indicate a 0 logic state. To assert the bus, a driver sinks current. To release the bus, drivers are asserted to a high-impedance state or turned off, allowing the bus signal to be pulled to the termination voltage of the bus.

NOTE

This typically results in a logical inversion of signals on TTL and BTL buses. Signals on ECL buses typically are not inverted.

All drivers operate in a wired-ORed or open-collector mode during arbitration. Drivers can operate in a totem pole mode during data packet and acknowledge transfers. In this mode, a driver can drive the bus into its released state in order to decrease the rise time of the bus signal (referred to as a rescinding release with TTL technology).

bit rates

Data transmission and reception occurs at 49.152 Mbit/s or 98.304 Mbit/s (±100 ppm). Regardless of the interface technology, arbitration occurs at an arbitration clock rate of 49.152 MHz.

backplane transmit data timing

Edge separation is the minimum required time between any two consecutive transitions of the backplane bus signals, as they appear from the output of the transmitters, whether they be transitions on the same signal or transitions on the two separate signals. A minimum edge separation is required to ensure proper operation of the data strobe bit-level encoding mechanism. TDATA and TSTRB have the relationship shown in Figure 9 and Table 13.

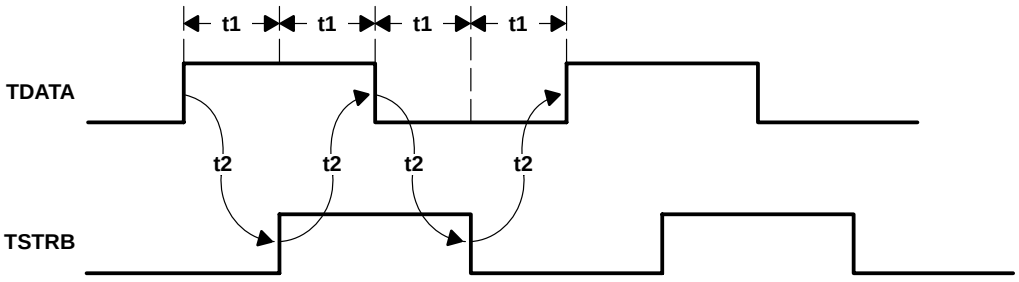


Figure 9. Minimum Edge Separation

Table 13. TSB14C01A to Backplane Transceiver Timing

PARAMETER		98.304 MHz	49.152 MHz
t1	Bit cell period for data	9.44 ns minimum	19.44 ns minimum
t2	Transmit (Tx) edge separation	8.65 ns minimum	18.65 ns minimum

PRINCIPLES OF OPERATION

backplane receive data timing

The receiver typically uses the transitions on the incoming bus signals RDATA and RSTRB to derive a clock at the code bit frequency to extract the NRZ signal on RDATA. This clock can be derived by performing an exclusive-OR (XOR) of RDATA and RSTRB.

The bus signals, as they appear from the backplane transceiver media and into the receivers, should be within the constraints outlined by Figure 10.

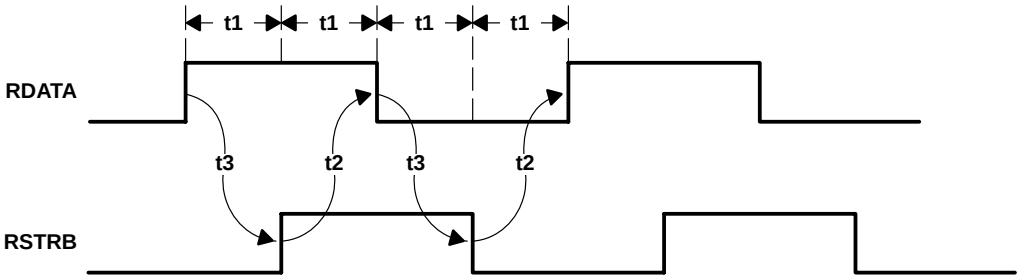


Figure 10. Backplane Receive Data Timing

Table 14. TSB14C01A to Backplane Transceiver Timing

PARAMETER		98.304 Mhz	49.152 MHz
t1	Bit cell period	10.1715 ns nominal	20.34 ns nominal
t2†	Receive (Rx) edge separation	3.4 ns minimum	3.4 ns minimum
t3	Receive (Rx) edge separation	16.3 ns maximum	36.6 ns maximum

† This parameter is based on a maximum total transmit skew of 2 ns and a maximum backplane skew of 0.5 ns. This assumes total receive skew is less than receive edge separation (i.e., some skew margin exists).

backplane timing definitions

- Logic Skew — The skew between data and strobe within the physical layer itself due to internal skews between data and strobe logic.
- Spatial Skew — The skew between data and strobe due to differences in propagation delays along the transmission line from arbiter to transceiver. This board delay is of concern when the physical layer and transceivers are packaged separately.
- Package Skew — The propagation delay difference through the transceiver between the data and strobe channels.
- Backplane Skew — The skew along the backplane itself due to impedance and/or mismatching of lengths in the data and strobe lines.
- Receive Setup/Hold — The setup and hold time needed to latch in the incoming data within the phy arbiter, based off of the recovered clock from Data-Rx and Strobe-Rx.
- Total Transmit Skew — The total skew between data and strobe in transmitting data from the phy out to the bus. This is given by the following equation.

Total Transmit Skew = Transmit Package Skew + Spatial Skew + Logic Skew

- Total Receive Skew — The total skew between data and strobe in receiving data from the bus into the phy. This is given by the following equation.

  Total Receive Skew = Receive Package Skew + Spatial Skew + Receive Setup + Receive Hold

- Skew Margin — The bit cell period minus all skews. This is given by the following equation.

  Skew Margin = Bit Cell Period − Total Transmit Skew − Backplane Skew − Total Receive Skew

- Transmit Edge Separation — The minimum time required between any two consecutive transitions of the bus signals to ensure proper operation of data-strobe bit level encoding. Transmit edge separation is measured from the midpoint of the signal transition to the midpoint of the next signal transition out on the bus. Minimum transmit edge separation is the minimum bit cell period less the maximum total transmit skew.
- Receive Edge Separation — The minimum time required between any two consecutive transitions of the bus signals to ensure proper operation of data-strobe bit level decoding. Receive edge separation is measured from the midpoint of the signal transition to the midpoint of the next signal transition out on the bus. This is the minimum bit cell period reduced by the amount of maximum total transmit skew and maximum backplane skew.

gap timing

A gap is a period of time during which the bus is idle (Data_Rx and Strb_Rx are unasserted). There are three types of gaps:

- Acknowledge Gap — Appears between the end of a packet and an acknowledge, as well as between isochronous transfers. A node should detect the occurrence of an acknowledge gap after the bus has been in an unasserted state for 4 arbitration clock times (approximately 81.38 ns), but should not assert the bus until a total of 8 arbitration clock times (approximately 182.76 ns) have occurred. This requirement ensures that a node is given adequate time to detect the acknowledge gap before the bus is asserted by another node upon detecting an acknowledge gap. This includes the minimum time required to detect a BUS_IDLE (4 arbitration clock times), as well as the maximum delay between the arbitration state machines within any two nodes on the bus (another 4 arbitration clock times).
- Subaction Gap — Appears before asynchronous transfers within a fairness interval. A node should detect the occurrence of a subaction gap after the bus has been in an unasserted state for at least 16 arbitration clock times (approximately 325.52 ns), but should not assert the bus until a total of 20 arbitration clock times (approximately 406.9 ns) have occurred. This requirement ensures that a node is given adequate time to detect the subaction gap before the bus is asserted by another node (upon detecting a subaction gap). The duration of the subaction gap ensures that another node asserting the bus after an acknowledge gap has been detected by this time.
- Arbitration Reset Gap — Appears before asynchronous transfers when the fairness interval starts. A node should detect the occurrence of an arbitration reset gap after the bus has been in an unasserted state for at least 28 arbitration clock times (approximately 569.66 ns), but should not assert the bus until a total of 32 arbitration clock times (approximately 651.04 ns) have occurred. This requirement ensures that a node is given adequate time to detect the arbitration reset gap before the bus is asserted by another node (upon detecting an arbitration reset gap). The duration of the arbitration reset gap ensures that another node asserting the bus after a subaction gap or an acknowledge gap has been detected by this time.

If a node is waiting for the occurrence of a particular gap, and the bus has become idle for the specified time (e.g., 32 arbitration clock times for an arbitration reset gap), the node detects the gap and asserts the bus within the time constraints described under the bus synchronization and propagation delay section of this document. These constraints ensure that an asserted signal propagates through the node decision/transceiver circuitry and onto the bus soon enough to allow arbitration to occur properly.

PRINCIPLES OF OPERATION

arbitration sequence

The arbitration sequence is the process that the TSB14C01A uses to arbitrate the bus between competing modules.

arbitration number

The arbitration sequence uses a unique arbitration number for each module. This 6-bit number is the same as the node physical_ID. When less than 6 bits are provided for the arbitration number, they occupy the MSBs of the arbitration number. The remaining bits are zero-filled. The MSBs are transmitted first.

NOTE

If the serial bus is contained within a host backplane, it is expected that the arbitration number (i.e., physical_ID) is set by the host backplane at power up (e.g., with a built-in slot identifier or configuration mechanism).

It is recommended that this number be software programmable to facilitate testing and to allow for consistent system operation and repeatability.

priority

Within the arbitration sequence, the arbitration number is preceded by four bits that define a priority level. The method by which priority is assigned is to be determined by the system integrator with two exceptions. The lowest priority (all zeros) is reserved for fair arbitration and the highest priority (all ones) is reserved for cycle start requests. This allows 14 priority levels to be used for the urgent arbitration process.

The use of an urgent priority class allows nodes to be granted a larger portion of the bandwidth on the bus. High priority nodes are granted the bus before lower priority nodes during urgent allocation of the bus, allowing such nodes to be granted more bandwidth.

In order to ensure forward progress, the lowest priority level is reserved for fair arbitration. This allows all nodes arbitrating with this priority level to be allowed one fair access to the bus for each fairness interval. For fair arbitration, the value of the arbitration number has a minimal impact on the allocation of the bus. Although nodes with higher arbitration numbers are granted the bus sooner, there is only a small decrease in latency.

The 4-bit priority field is not used in isochronous arbitration. When arbitrating for an isochronous transfer, the priority field is zero-filled.

PRINCIPLES OF OPERATION

format of arbitration sequence

The following format for the arbitration sequence should be used:

PRIORITY	ARBITRATION NUMBER
4 bits	6 bits

arbitration sequence

- Each module on the backplane has a unique 6-bit arbitration number that is equal to the node's physical_ID.
- The arbitration number is preceded by four bits of priority. The MSB of the priority field is transmitted first. The LSB of the priority field is followed by the MSB of the arbitration number.
- Dynamic assignment of priority is accommodated.
- The lowest priority level (all zeroes) is reserved for fair arbitration, and the highest priority level (all ones) is reserved for the identification of the cycle start packet.

arbitration

Unless a node is using immediate arbitration to access the bus (in which case there is no contention for the bus), it is possible that more than one node can attempt to access the bus at a given time. Consequently, it is necessary for a node to arbitrate for the bus in order to gain access to the bus.

NOTE

A node uses immediate arbitration to send an acknowledge. Since there is no contention for the bus in this case, arbitration is not necessary. A node that is transmitting an acknowledge does not arbitrate for the bus, but merely waits for an acknowledge gap to be detected before it begins transmission. If a node is attempting to gain access to the bus without using immediate access, it must first arbitrate for the bus.

Arbitration occurs in response to a phy arbitration request from the link. Nodes begin arbitrating once the bus has become idle for a predetermined amount of time (the appropriate gap indication occurs). Once this happens, nodes begin a bit-by-bit transmission of their arbitration sequence.

A node can obtain access to the bus in a limited number of ways. Since some arbitration classes allow nodes to begin arbitration before others, nodes arbitrating with certain arbitration classes can detect that the bus is busy before they can begin to arbitrate. In this way, certain arbitration classes can be bypassed. For example, fair and urgent nodes do not get a chance to arbitrate when another node is sending an acknowledge or when it is arbitrating for an isochronous transfer.

The backplane environment supports the fair, urgent, cycle_master, isochronous, and immediate arbitration classes.

fairness intervals

The fairness protocol is based on the concept of a fairness interval. A fairness interval consists of one or more periods of bus activity separated by short idle periods called subaction gaps and is followed by a longer idle period known as an arbitration reset gap. At the end of each gap, bus arbitration is used to determine the next bus owner. This concept is shown in Figure 11.

PRINCIPLES OF OPERATION

fairness intervals (continued)

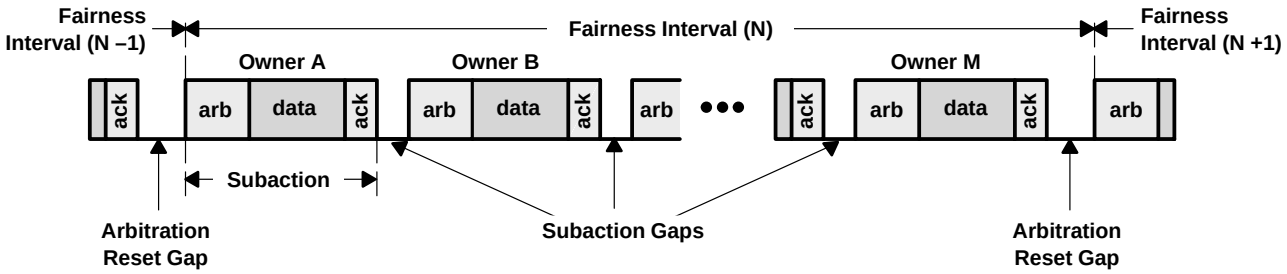
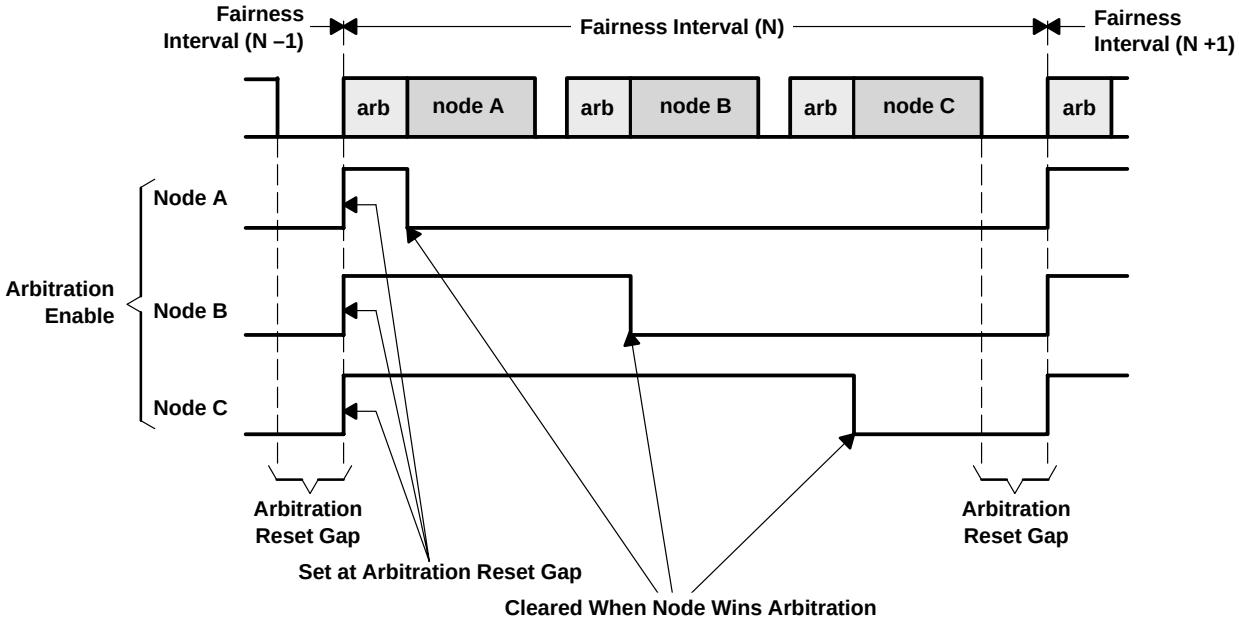


Figure 11. Fairness Interval

The implementation of the fair arbitration protocol is defined in terms of these fairness intervals as discussed in the following paragraphs.

fair arbitration

When using this arbitration class, an active node can send an asynchronous packet only once each fairness interval. Once a subaction gap is detected, a node can begin arbitration when its arbitration_enable signal is set. The arbitration_enable signal is set at the beginning of the fairness interval and is cleared when the node successfully accesses the bus through fair arbitration. This disables further fair arbitration attempts by that node for the remainder of the fairness interval. In the absence of urgent nodes, a fairness interval ends once all of the nodes attempting fair arbitration have successfully accessed the bus. At this time, all of the fair nodes have their arbitration_enable signals reset and cannot arbitrate for the bus. The bus remains idle until an arbitration reset gap occurs. Once this happens, the next fairness interval begins. All of the nodes set their arbitration_enable signal and can begin to arbitrate for the bus. This process is illustrated in Figure 12.



NOTE A: The arbitration number of A > B > C.

Figure 12. Fair Arbitration Timing

Note that a node sending a concatenated subaction does not reset its arb_enable bit.

urgent arbitration

The backplane environment enhances the fair priority algorithm by splitting access opportunities among nodes based on two priority classes: fair and urgent. Nodes using an urgent priority can use up to three-fourths of the access opportunities, with the remaining ones equally shared among nodes using the fair priority. All nodes are required to implement the fair priority class, while the urgent priority class is optional. Packets are labeled as urgent when that priority class was used.

The fair/urgent allocation uses the same fairness interval described in fair arbitration but accompanies the arbitration_enable flag with an urgent_count. The fair/urgent method works as follows:

- When the bus is idle for longer than an arbitration reset gap, a fairness interval begins and all nodes set their arbitration_enable flags, while nodes implementing urgent priority set their urgent_count to three.
- A node that is waiting to send a packet using the fair priority class should begin arbitrating after detecting a subaction gap as long as its arbitration_enable flag is set. When its arbitration_enable flag is cleared, it waits for an arbitration reset gap before it begins arbitrating. When such a node wins an arbitration contest, it sends a packet without the urgent label and its arbitration_enable flag is cleared.
- A node that is waiting to send a packet with urgent priority begins arbitrating after detecting a subaction gap if its urgent_count is nonzero. When its urgent_count is zero, it waits for an arbitration reset gap before it begins arbitrating. Whenever such a node wins an arbitration contest, it sends a packet with the urgent label.
- A node implementing urgent priority sets its urgent_count to three whenever an unlabeled (i.e., fair) packet is transmitted or received. This includes received packets that are addressed to other nodes.
- A node decrements its urgent_count whenever a packet with the urgent label is transmitted or received. This includes received packets that are addressed to other nodes. This ensures that there is at most three urgent packets for every fair packet. This does not ensure that every node using urgent priority obtains the bus three times for each fairness interval. The node arbitrating with the highest priority always obtains the bus before other nodes arbitrating with an urgent, but lower, priority.

In the presence of urgent nodes, a fairness interval ends after the final fair node and up to three remaining urgent nodes have successfully accessed the bus. Since all fair nodes now have their arbitration_enable signals reset and all urgent nodes have their urgent_count decremented to zero, none of the nodes can access the bus. The bus remains idle until an arbitration reset gap has occurred, re-enabling arbitration on all nodes and starting the next fairness interval. This process is illustrated in Figure 13, which illustrates a situation where there are three nodes arbitrating for the bus with physical_IDs such that A has the highest priority, B is in the middle priority, and C has the lowest priority, with nodes A and C using fair priority and B using urgent:

PRINCIPLES OF OPERATION

urgent arbitration (continued)

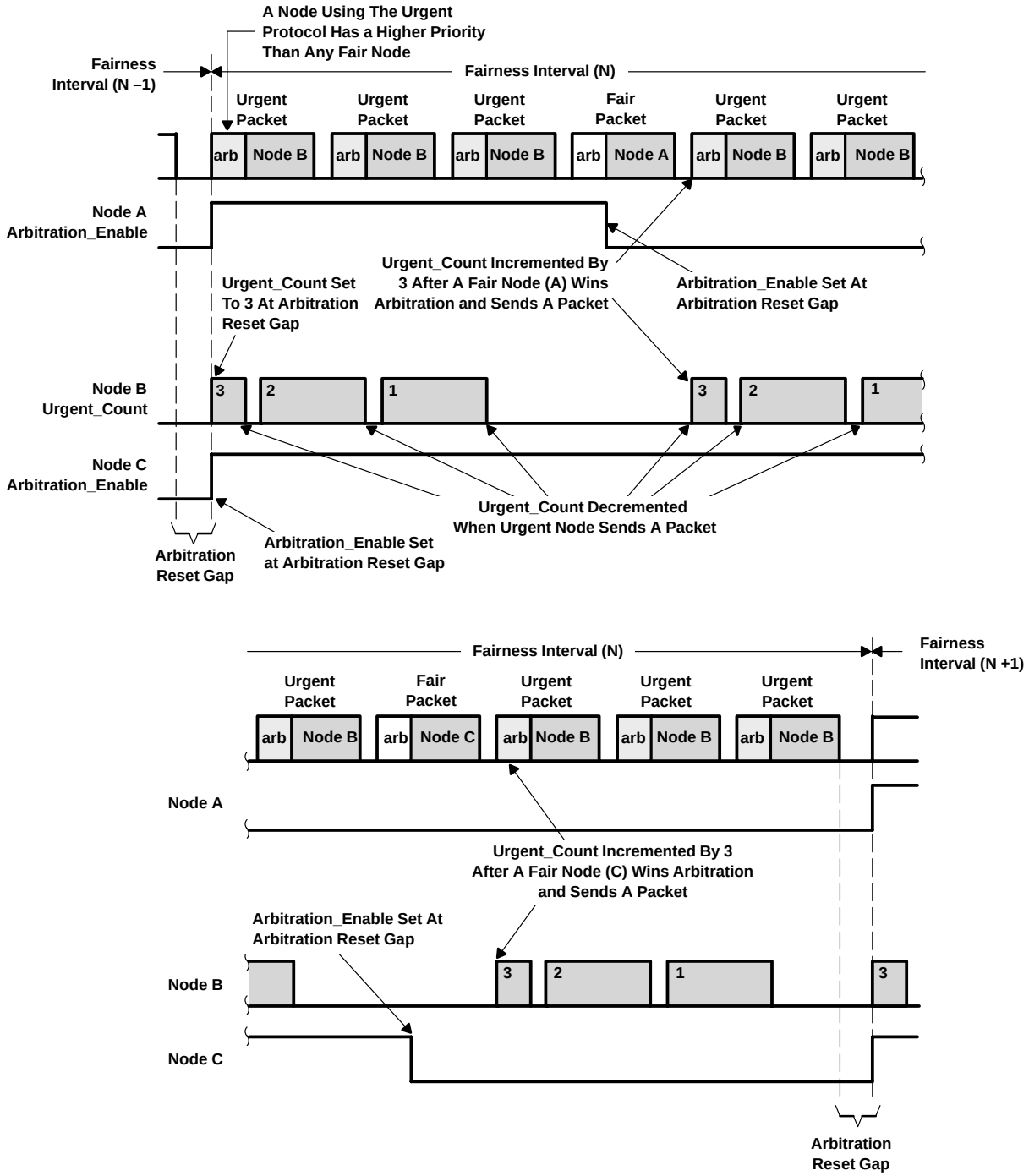


Figure 13. Urgent Arbitration

PRINCIPLES OF OPERATION

urgent arbitration (continued)

In the backplane environment, the natural priority is the concatenation of the 4-bit urgent priority level with the physical_ID. These results are listed in the following list.

- A node using the urgent priority always wins an arbitration contest over all nodes using the fair priority.
- The node using the highest priority level wins the arbitration contest.
- When more than one node uses the highest priority level, then the one with the highest physical_ID wins.

arbitration by the cycle_master

This arbitration class is used by the cycle_master when it needs to arbitrate for the transmission of a cycle_start packet. It is similar to the urgent arbitration class, except that the priority field is defined to be all ones. Arbitration begins once a subaction gap is detected, regardless of the state of the arbitration_enable signal or the urgent_count.

isochronous arbitration

This arbitration class is used by nodes arbitrating to send isochronous packets. Arbitration begins once an acknowledge gap is detected, regardless of the state of the arbitration_enable signal or the urgent_count. Because an acknowledge gap is shorter than an arbitration reset gap and a subaction gap, nodes arbitrating with this class win the bus before nodes waiting to send fair, urgent, or cycle_master packets. The priority field is not used in this arbitration class.

immediate arbitration

This arbitration class is used by nodes sending an acknowledge to a received packet. Transmission of the acknowledge (beginning with a DATA_PREFIX) occurs as soon as an acknowledge gap is detected. This arbitration class is referred to as immediate because an arbitration sequence is not transmitted to obtain access to the bus (i.e., the node does not actually arbitrate for the bus).

backplane phy reset

Upon a power_reset event (i.e., power up), registers and control and status registers (CSRs) associated with the operation of the phy are initialized to their default values. State machines associated with phy operations are initialized. The BUS_RESET signal is not transmitted on the bus by the phy.

PRINCIPLES OF OPERATION

bus reset

Once a phy control request of bus reset is communicated from the node controller to the phy, the registers and CSRs associated with the operation of that phy are initialized to their default values. State machines associated with phy operations can be initialized. The phy communicates a BUS_RESET onto the bus. After the BUS_RESET event is detected by the node transmitting it (i.e., approximately 320 arbitration clock times after the signal is initiated), the phy initializes itself.

NOTE

Since a phy transmitting a BUS_RESET must also react accordingly once the BUS_RESET event is detected, its state machines do have the opportunity to advance beyond those of other nodes. This ensures that all nodes are in somewhat similar states after such a bus reset event. Obviously, the logic (e.g., counters) used within the phy to generate the BUS_RESET signal must not be reset once that phy simultaneously detects its own BUS_RESET.

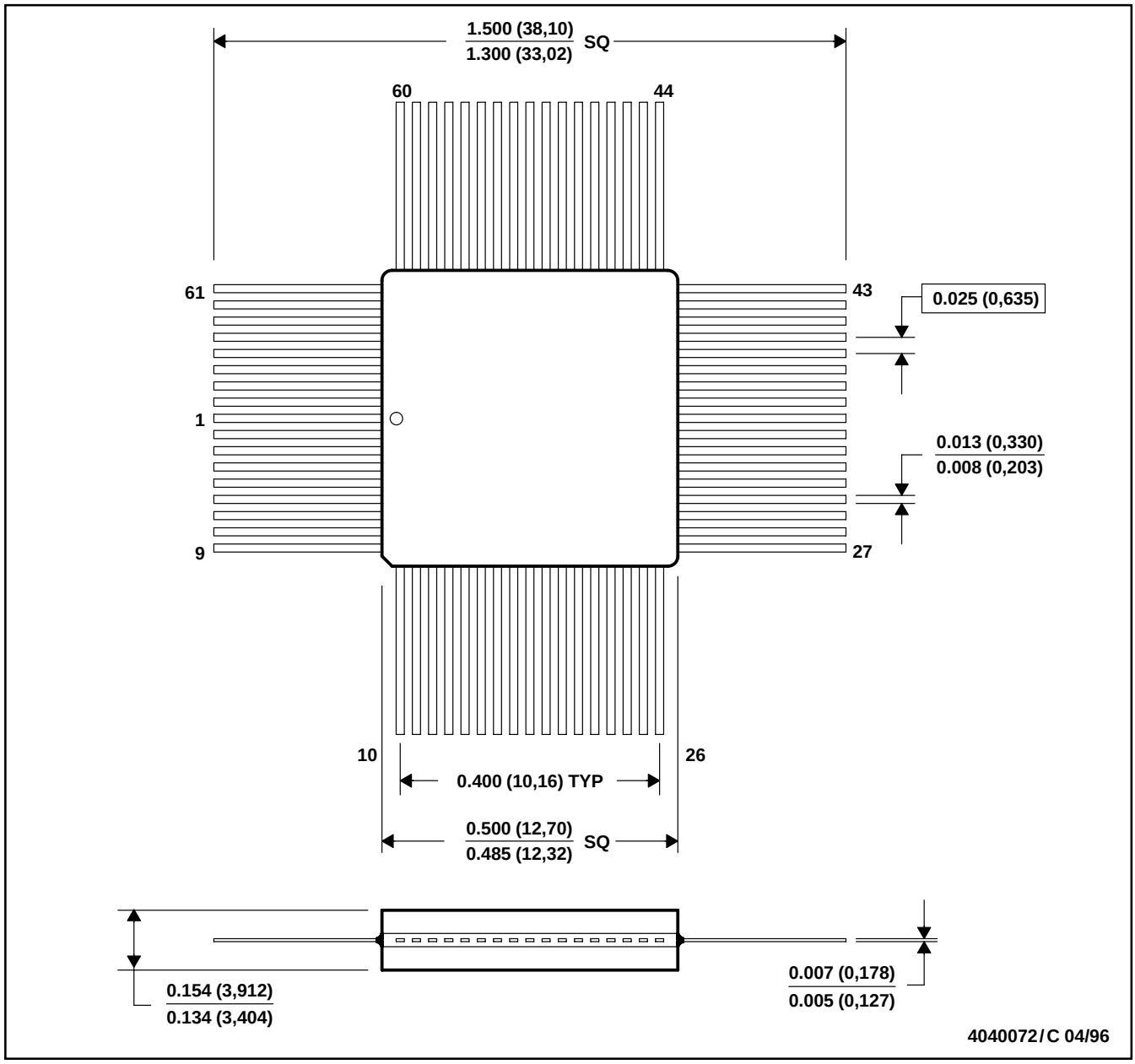
Once a phy detects that a BUS_RESET event has occurred on the bus, it initializes itself. The node controller initiates a bus reset by writing to the phy control register with the IBR bit set to 1. The phy hardware resets the IBR bit to zero on detection of its own bus reset.

live insertion

It is up to the user to design their node or module to safely receive power from the particular backplane they are using during a live insertion or tap-in. But in principal, live insertion is supported and does not cause a bus-reset event to occur. Please see the TI application note, *Live Insertion*, TI literature number SDYA012.

MECHANICAL INFORMATION

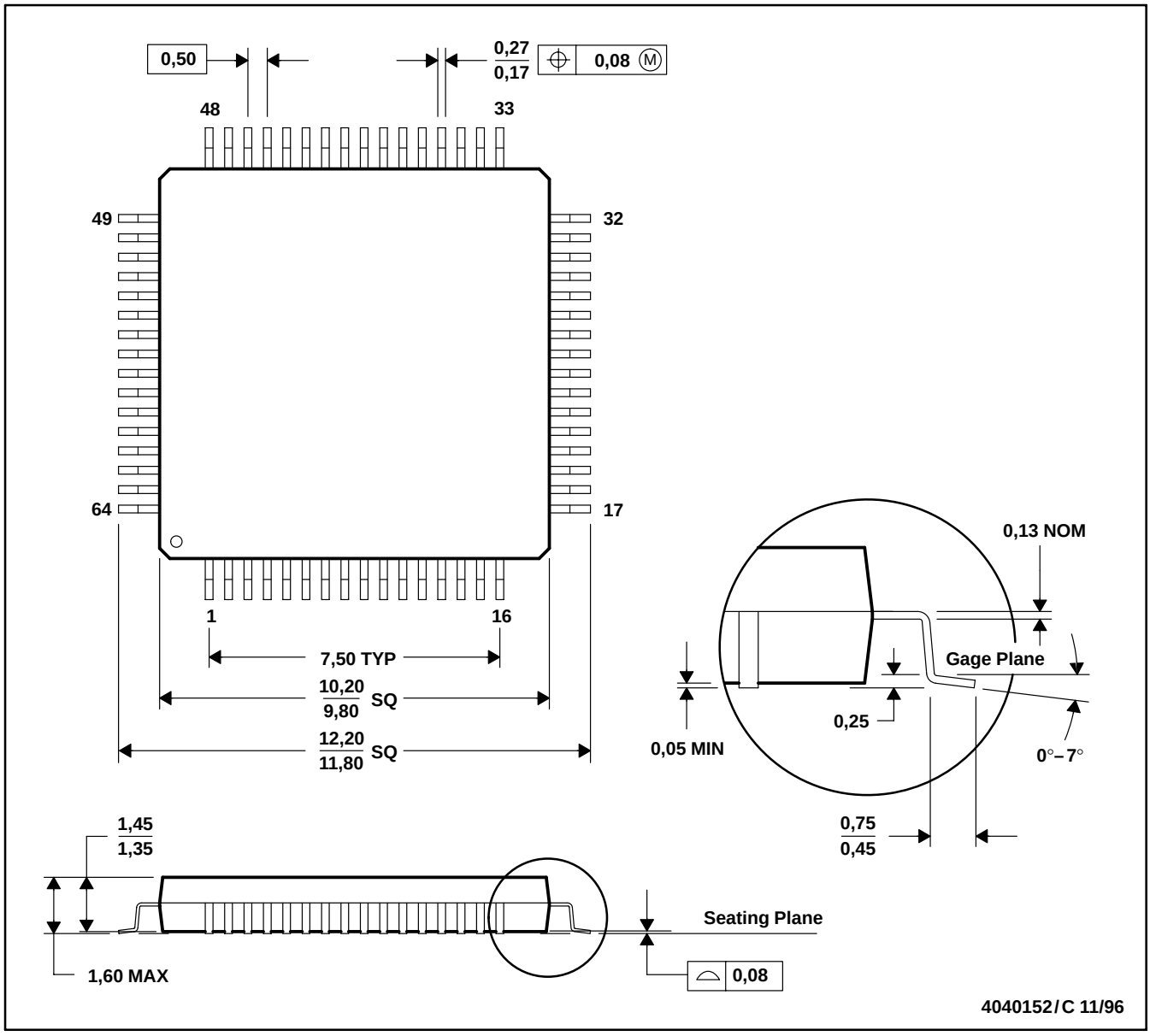
HV (S-GQFP-F68) **CERAMIC QUAD FLATPACK**



NOTES: A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package can be hermetically sealed with a ceramic lid using glass frit.

MECHANICAL INFORMATION

PM (S-PQFP-G64) **PLASTIC QUAD FLATPACK**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-026
D. May also be thermally enhanced plastic with leads connected to the die pads.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third–party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265