Interfacing the TSB12LV41 1394 Link Layer Controller to the TMS320AV7100 DSP Embedded ARM Processor

APPLICATION BRIEF: SLLA015

Randy Lawson
1394 Applications Group

Mixed Signal and Logic Products
BuS Solutions Group
15 December 1997

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

GPLynx, PCILynx, Lynxsoft, Mpeg2Lynx, 1394 TImes are trademarks of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

CONTACT INFORMATION

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	dsph@ti.com
US TI 1394 Applications	1394@ti.com

Contents

Figures

Tables

Interfacing the TSB12LV41 1394 Link Layer Controller to the TMS320AV7100 DSP Embedded ARM Processor

Abstract

This document describes the physical interconnections and timing requirements of the interface between the TMS320AV7100 and the TSB12LV41 1394 Link Layer Controller. The TMS320AV7100 has an embedded ARM7TDMI processor core. The TSB12LV41 is TI's 1394 Link Layer Controller, which supports the proposed IEC61883 specification's definition for transmitting and receiving compressed MPEG2 video/audio on the 1394 High Speed Serial Bus. Both DSS and DVB formatted MPEG2 Transport Packets are supported.

Conventions

Active low signals are indicated by adding a "Z" to the end of the signal name. For example, the active low signal named CS is noted in this document as CSZ.

Product Support

Related Documentation

The following list specifies product names, part numbers, and literature for the TI documents.

- *TSB12LV41 (MPEG2Lynx), IEEE 1394-1995 Link-Layer Controller for MPEG2 Transport, Product Preview Data Sheet*, Literature number SLLS276A
- *Endianness and the TSB12LV41 (MPEG2Lynx) Microprocessor Interface*, Literature Number SLLA021

The IEEE 1394-1995 standard is available for purchase at http://standards.ieee.org/catalog/index.html

All other future TI 1394 application notes can be found at the TI 1394 external web page.

World Wide Web

Texas Instruments World Wide Web site at www.ti.com contains the most up to date product information, revisions, and additions. Users registering with TI&ME can build custom information pages and receive new product updates automatically via email.

The URL specifically for the TI 1394 external web site is http://www.ti.com/sc/1394. On this page, one can subscribe to 1394Times, which periodically updates subscribers on events, articles, products, and other news regarding 1394 developments.

Email

For technical issues or clarification on products, please send a detailed email to 1394@ti.com.

Functional Description

This document will focus on the TSB12LV41 Link Layer Controller's Microprocessor/Microcontroller Interface Block support for the TMS320AV7100 DSP with the integral ARM7TDMI core.

To be able to detect which kind of microprocessor/microcontroller (MP/MC) is connected to the TSB12LV41, two MP/MC select lines, MCSEL1 and MCSEL0, need to be driven to certain logic levels at all times. The values shown in the table below for MP/MC Type show the correct combined logic level for the TMS320AV7100 microprocessor.

Once the type of MP/MC has been determined, all the I/O control pins related to MP/MC Interface map their functions to be compatible with the TMS320AV7100 microprocessor.

Table 1. Extension Bus Pin/Function Matrix for TSB12LV41/TMS320AV7100 Interface

TSB12LV41 Signal Name	**TMS320AV7100 MCSEL1 = 0 MCSEL0 = 0**
ADR[0:8] (Mcaddr)	EXTADDR[8:0]
DATA[0:15] (Mcdatain, Mcdataout)	EXTDATA[15:0]
CS/CSZ (MccsZ)	CSXZ
MCCTL0 (McwrZ)	EXTR/WZ
MCCTL1 (McrdZ)	Unused, Tied High
RDY (McRdy)	EXTWAITZ
BCLK	CLK40

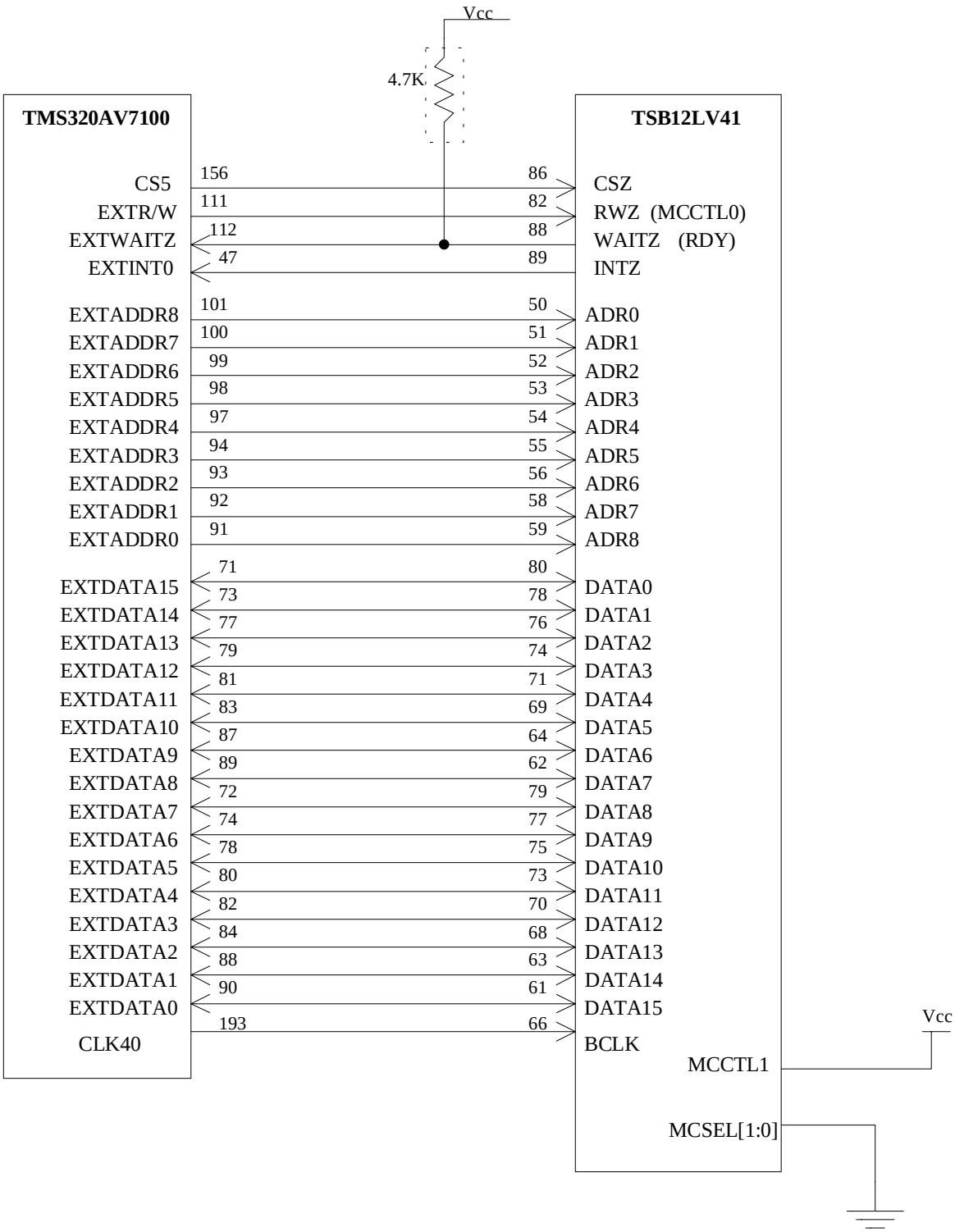
The TSB12LV41's Micro Interface Block is synchronized to the BClk. BClk input is from the TMS320AV7100's extension bus external clock input (CLK40, 40.5MHz, 24.5ns).

Definitions

- Quadlet = 4 bytes, 32 bits
- Doublet = 1/2 quadlet, 16 bits
- MSB = Most significant bit
- LSB = Least significant bit

If not mentioned separately, all the bus signal labeling in the TSB12LV41's Micro Interface Block is denoted as bit0 is MSB and bit15 is LSB.

Figure 1. Connection Diagram of TMS320AV7100 Extension Bus to the TSB12LV41



Note: Pull Up resistor is required on the WAITZ signal if “RdyPushPull” bit in the IOCR register is set to 0.

Table 2. TSB12LV41 IOCR Register (Address 1ECh)

0	1	2	3	4	5	6	7	Bits 8 - 31
MCMP8	BeCtl	IntPol	RdyPushPull	IntPushPull	BlindAccess	DataInvarnt	RdyPol	Reserved

Table 3. TSB12LV41 IOCR Bit/Function Correlation Table and Power-up Default Setting

Bit	Bit Name		Bit Value Setting Meaning		Power Up Default Setting
Number	Symbol	Description	Value = 1	Value = 0	TMS320AV7100
0	**MCMP8**	Micro bus is 8/16 bit access	Byte Access	Word Access	Word Access
1	**BeCtl**	Big Endian Control	Big Endian	Little Endian	Big Endian
2	**IntPol**	Interrupt Polarity Control	High True	Low True	Low True
3	**RdyPushPull**	“RDY” output signal control	Active Push/Pull	Three-State	Three-State
4	**IntPushPull**	“INT” output signal control	Active Push/Pull	Three-State	Active Push/Pull
5	**BlindAccess**	Blind Access Enable/Disable	Enable Blind Access Mode	Disable Blind Access Mode	Enable Blind Access Mode
6	**DataInvarnt***	Data Invariant Endianness Cntl	Data Invariant	Address Invariant	Data Invariant
7	**RdyPol**	Rdy output Polarity Control	High True	Low True	Low True
8 - 31	**Reserved**				

Note: When the BeCtl bit is set to “1” (Big Endian), the DataInvarnt bit setting has no effect.

It is strongly recommended that users program the IOCR setting during the first access after the power reset. This will ensure that the TSB12LV41’s microprocessor interface is working correctly with the TMS320AV7100. (To ensure the data path in the microprocessor interface is working correctly, the IOCR bit settings from the write always updates after the current transaction’s chip select deasserts.) To be able to get the new IOCR setting updated, it is very important to allow a 4 clock (for 40.5 MHz TMS320AV7100 clock) idle time after the last write to fill up the whole quadlet of IOCR. This rule applies to all the IOCR writes, regardless of whether it is the first power up write or a later setting change write.

Since the microprocessor data path from the Host Interface to the TSB12LV41 is 16 bits wide, but the internal registers are 32 bits wide, the TSB12LV41 has to perform write byte stacking and read byte unstacking. It also performs byte swapping for Little Endian processors if the BeCtl bit in the IOCR register is set to 0. Users have to be very careful when they set up the BeCtl and DataInvarnt bits. If the BeCtl is set to an endianness that is opposite the endianness of the actual microprocessor used, the data result from a read/write access is unpredictable.

For the TMS320AV7100, both Byte Access and Word Access are supported. Little Endian will be supported, but users have to take the risk of wrong data byte swapping, because the TMS320AV7100 embedded processor is Big Endian.

General Read/Write Functional Description

The microprocessor can access the TSB12LV41 in either handshake mode or non-handshake (blind-access) mode. The handshake signal between TSB12LV41 and the microprocessor is called “WAITZ” (which is the RDY pin on the TSB12LV41). When this is low the TMS320AV7100 interprets this as a wait request from the 12LV41. A read or write request is complete after the TMS320AV7100 sees the WAITZ go back high. The read/write transaction cycle from the extension bus side is controlled by the chip select. In handshake mode, the microprocessor normally drives the target address on the address bus, asserts the chip select and write/read control line for the type of transaction and provides/waits for data on the data bus. Then the microprocessor counts on the WAITZ signal coming back from TSB12LV41’s microprocessor interface to terminate the cycle. The non-handshake mode is also called the Blind Access Mode.

In Blind Access Mode, the microprocessor doesn’t wait for the WAITZ signal to terminate the cycle. Instead, the microprocessor always terminates the current transaction in a fixed number of cycles. It then polls the Blind Access Status Register address 1F0h to learn if the current transaction is finished or not. Regardless of whether the interface is handshake or non-handshake, there are some common read/write rules that have to be followed in order to carry out the correct transaction.

General Write Instructions

For the write case, the TSB12LV41's Micro Interface will only initiate the write data transfer when the entire quadlet is filled up. This means that the first three-byte write (in byte mode), or first word write (in word mode), only loads part of the quadlet into the temporary byte stacking storage in the Micro Interface. Once the TSB12LV41 has received the complete quadlet, it then generates a cycle start to the internal Host Interface and also distributes the quadlet data to the internal link logic based on the given quadlet address. Meanwhile, it generates a cycle acknowledge (WAITZ) to the Micro Interface, even if a response doesn't come back from the internal Host Interface. This would allow the Micro Interface to terminate the current transaction and accept a new transaction, which can be a new read process or some write process's pre-loading of the first three bytes or first word. If the microprocessor is very fast and the link core is slow to send back the response signal, the Micro Interface could issue a new transaction to the internal Host Interface. At that time, if the Host Interface is still busy on the first transaction, it will latch the new transaction and put the Host Interface's state machine into a "WAIT" state without generating another acknowledge back to the Micro Interface. This would block any possible third transaction. When the response comes back, the Host Interface starts the new or latched transaction, if any.

NOTE:

The TSB12LV41 will not hold the Microprocessor write transaction cycle while the Host Interface is still working on delivering the quadlet data to the link side. This could provide some performance gain.

General Read Instructions

For the read case, the TSB12LV41 Micro Interface initiates the read process to the internal link logic after it senses a read request to a new quadlet address generated by the microprocessor, regardless of which byte position the read address falls into inside the quadlet. The Micro Interface generates a cycle start to the internal Host Interface and passes the read quadlet address to the internal Link Logic. Then the Host Interface starts generating a read request to the link logic based on the given read quadlet address, and waits for a response being sent back from the link side. Once the response comes back, a cycle acknowledge is generated to the Micro Interface to indicate the current read process is finished and the whole quadlet data is valid to be read out from the Micro Interface's data bus. If handshake mode is selected for the Micro Interface via the IOCR register, the TSB12LV41 will hold the Microprocessor read transaction cycle until the acknowledge comes back from the internal Host Interface. Then the TSB12LV41 releases the WAITZ line to indicate to the microprocessor that the current transaction can be terminated.

The microprocessor Interface's temporary byte stacking storage holds the whole quadlet provided by the internal link logic. The follow up reads to different byte/word positions within the same quadlet boundary can obtain the data from the stacking storage rather than access the link logic every time. Therefore the first read to the new quadlet address always takes a little more time than all the other follow up reads. The read access latency time to the internal link logic is about 17 clock cycles (for a 40.5 MHz TMS320AV7100 clock). If you try to read the same byte/word position inside the same quadlet boundary twice, the Micro Interface would think the second read is trying to get the new data. Therefore it will initiate a new read to the same quadlet again to obtain the most recent data.

Summary

Summarizing the above information, the first read to a new quadlet address is the one that will handshake with the internal link core to obtain the quadlet data. The follow up reads within the quadlet boundary only have to read data from Micro Interface's byte stacking storage. The last write to fill up the whole write quadlet is the one that will handshake with the internal link core to transfer the whole quadlet. The first three byte writes or first doublet write only load(s) data into Micro Interface's byte stacking storage.

TMS320AV7100 Microprocessor Interface Timing Requirements

The TSB12LV41 supports any byte/doublet order writes, as long as they are within the same quadlet boundary. If a user tries to do either one of the following operations before the complete quadlet is filled, the current write will be ignored and an INVWROP (bit 11 in the extend Interrupt Register, address 018h) Interrupt will be issued:

- Microprocessor attempts to write to a new address before filling up the presently addressed quadlet
- Microprocessor attempts to read from a new address before filling up the presently addressed quadlet
- Microprocessor attempts to write to the same doublet or byte back-to-back

Figure 2 and Table 4 define the timing requirements for the TMS320AV7100 mode. These requirements must be met for both handshake and blind access modes, and for both read and write operations.

Critical Timing Requirements

Figure 2. TSB12LV41/TMS320AV7100 Critical Timing Diagram

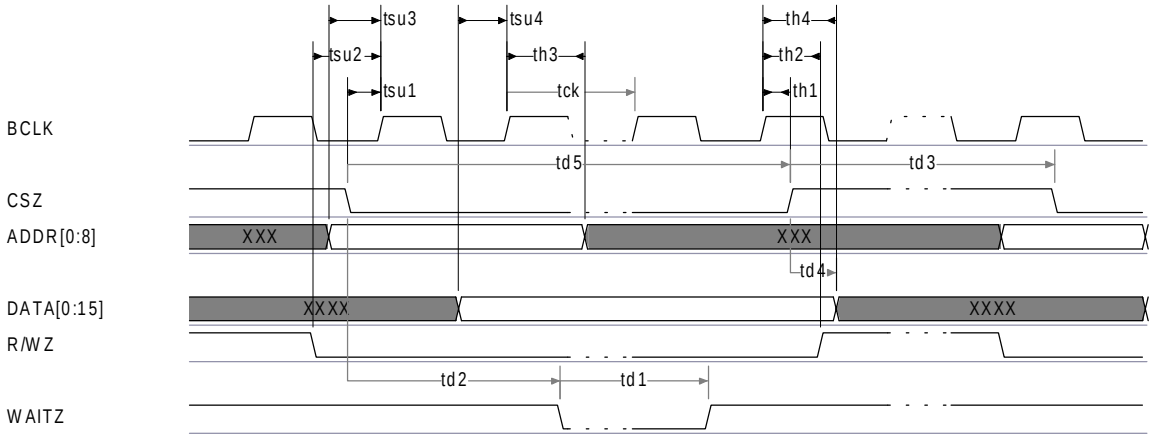


Table 4. Critical Timing Parameters

Parameter	Min (ns)	Max (ns)	Description
tsu1	12.2		CSZ setup time to BCLK rising edge
tsu2	12.1		R/WZ setup time to BCLK rising edge
tsu3	8.9		Address setup time to BCLK rising edge
tsu4	1.2		Data setup time to BCLK rising edge[1]
th1		0.8	CSZ hold time after BCLK rising edge
th2		1.0	R/WZ hold time after BCLK rising edge
th3		0.8	Address hold time after BCLK rising edge
th4		0.9	Data hold time after BCLK rising edge[1]
td1		17 BCLK cycles	WAITZ low[3]
td2		2 BCLK cycles + 8ns	CSZ falling edge to WAITZ
td3	2.5 BCLK cycles		CSZ rising edge to CSZ falling edge
td4		5.0	DATA valid after CSZ rising edge[2]
td5	5 BCLK cycles		CSZ low[4]
tck	24.5		BCLK period

Notes: 1) For Write operations.
2) For Read operations. The TSB12LV41 will hold data on the bus for a certain amount of time after the CSZ is deasserted.
3) WAITZ can go low for a maximum of 17 BCLK cycles on the *last* byte or word Write access to a specific address, OR on the *first* byte or word Read access to a specific address. Previous writes or consecutive reads to the specified address will be much quicker.
4) CSZ minimum low time for read/write cycles without an acknowledge.

Examples of these erroneous operations are given in the section TMS320AV7100 Microprocessor Interface Timing Requirements.

TMS320AV7100 Handshake Mode

TSB12LV41 is designed to meet the read/write access timing defined for the TMS320AV7100 embedded ARM processor. The following table shows the write timing for TMS320AV7100 interface.

Handshake Mode Write Operations

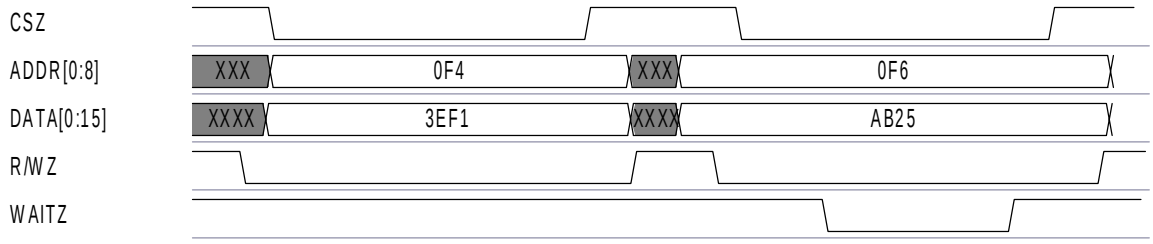
In the example below, the TMS320AV7100 embedded ARM processor wants to write hex value 3EF1AB25h to the TSB12LV41 Register 0F4h (DSS Formatter Control Register).

This operation assumes the IOCR register (1ECh) setting is 4A000000h, which corresponds to the following bit settings:

Table 5. IOCR Register Settings

Bit #	Bit Name	Bit Value	Bit Meaning
0	MCMP8	0	Word Access
1	BeCtl	1	Big Endian
2	IntPol	0	Low True
3	RdyPushPull	0	High Impedance State
4	IntPushPull	1	Active Push/Pull
5	BlindAccess	0	Disable Blind Access Mode
6	DataInvarnt	1	Data Invariant
7	RdyPol	0	Low True
8-31			Reserved

Figure 3. TSB12LV41/TMS320AV7100 Handshake Write Timing Example



Note: The BCLK signal is not shown in the above diagram for purposes of clarity, but is used in this mode.

After this operation in the Microprocessor interface the contents of Register 0F4h are as shown below:

Table 6. Register 0F4h Contents

MSByte			LSByte
Bits 0-7	**Bits 8-15**	**Bits 16-23**	**Bits 24-31**
3E	F1	AB	25

For the write case, the TSB12LV41 latches the data at the next clock rising edge after the CSZ goes low, which meets the td3 (5 Ns) write data setup time requirement. Once the TSB12LV41 is ready to terminate the current cycle, it deasserts the WAITZ signal. After certain internal combinational logic delay, the TMS320AV7100 will sample this deassertion by the next rising edge of its CLK40 (same source of BClk). Half cycle later, at the falling edge of CLK40, the TMS320AV7100 deasserts its CSZ.

NOTE:
The TSB12LV41 will not use the rising edge of CSZ to latch the data. This should be OK as long as TMS320AV7100 maintains the write data setup time (td3) at 5.0 Ns.

Handshake Mode Read Operations

In the example below, the TMS320AV7100 embedded ARM processor wants to read TSB12LV41 Register 0F4h (DSS Formatter Control Register). The value stored in this register is 1AB2AB25h.

This operation assumes the IOCR register (1ECh) setting is 4A000000h, which corresponds to the following bit settings:

Table 7. IOCR Register Settings

Bit #	Bit Name	Bit Value	Bit Meaning
0	MCMP8	0	Word Access
1	BeCtl	1	Big Endian
2	IntPol	0	Low True
3	RdyPushPull	0	High Impedance State
4	IntPushPull	1	Active Push/Pull
5	BlindAccess	0	Disable Blind Access Mode
6	DataInvarnt	1	Data Invariant
7	RdyPol	0	Low True
8-31			Reserved

Figure 4. TSB12LV41/TMS320AV7100 Handshake Read Timing Example

Note: The BCLK signal is not shown in the above diagram for purposes of clarity, but is used in this mode.

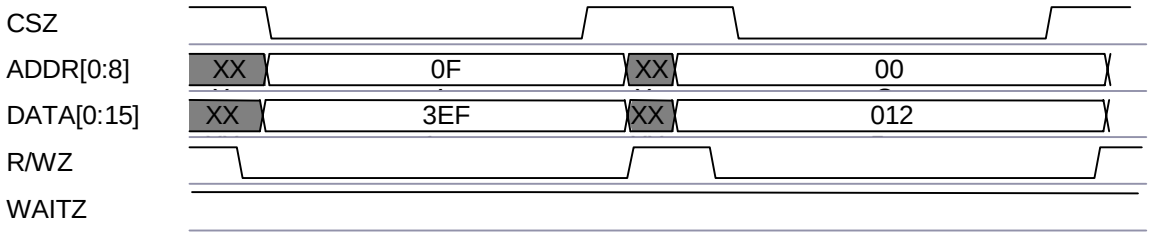
For the read case, once the TSB12LV41 has the read data available, it deasserts WAITZ. The TMS320AV7100's internal combinational logic senses the change. Then the output of this combinational logic is sampled by the next rising edge of its BCLK and CSZ will deassert at the same time. The TSB12LV41 uses this CSZ rising edge asynchronously to turn off the data bus. Therefore, turning off the read cycle after WAITZ takes approximately two BCLK cycles.

Incorrect Write Operation Examples

In Example #1 below, the host processor wants to write to address 0F4h and is using Handshaking and 16 bit (Word) access. The host correctly writes value 3EF1h to the upper two bytes of register 0F4h. Then the host attempts to do a write operation to register 00Ch without finishing the write to address 0F4h. In Example #2 the host writes to the upper two bytes of address 0F4h then attempts to read address 00Ch. In Example #3 the host attempts to write to the same doublet address two times in a row. Each of these examples will result in an INVWROP interrupt and the last write/read attempt will be ignored. The entire quadlet at address 0F4h must be written to before any other write or read operation can occur.

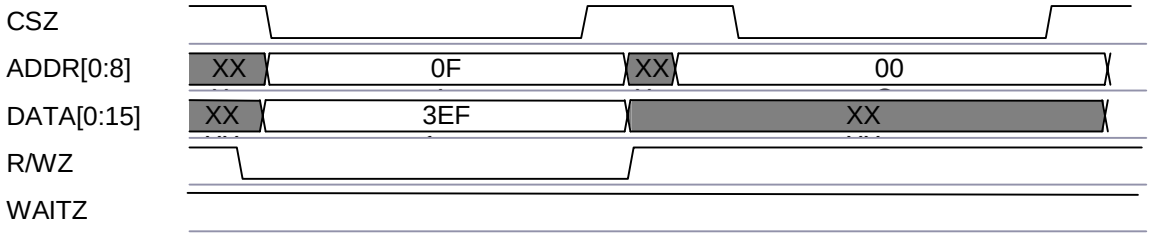
Example #1

Figure 5. INCORRECT WORD ACCESS WRITE – Attempted Write to a New Address Before Finishing Quadlet Write



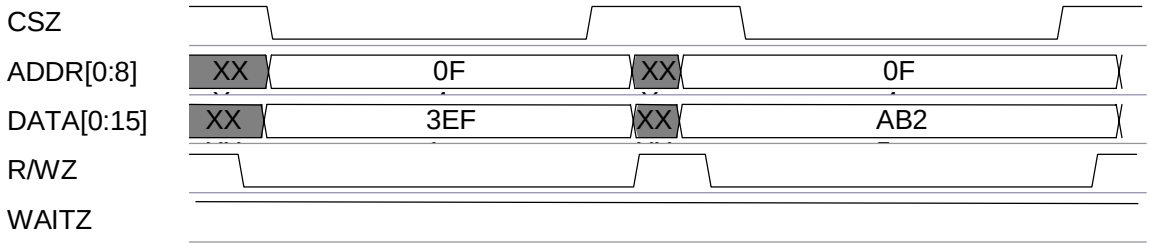
Example #2

Figure 6. INCORRECT WORD ACCESS WRITE – Attempted Read Operation Before Finishing Quadlet Write



Example #3

Figure 7. INCORRECT WORD ACCESS WRITE – Attempted Write to the Same Byte(s)



Important Notes on Handshake Mode

- 20 CLK40 Rule for WAITZ signal

  Since the WAITZ signal has the potential to stall the whole decoding process, the ARM7TDMI processor embedded in the TMS320AV7100 will cap its waiting at 500 ns (about 20 CLK40 cycles). Afterwards, the TMS320AV7100 assumes the devices that generated the WAITZ have failed and will ignore

EXTWAITZ from then on. Only a software or hardware reset can activate the EXTWAITZ signal again. In other words, the maximum allowed EXTWAITZ assertion time is 500ns for each chip select, regardless of whether the current transaction is 8-bit wide or 16-bit wide.

To avoid this timeout on the WAITZ signal, the TSB12LV41 Micro Interface state machine is designed to abort the transaction and deassert the WAITZ signal when the TSB12LV41's internal logic can't finish the internal read or write transaction after 17 BClk cycles.

- WAITZ to be recognized by TMS320AV7100 at the beginning of the transaction.

  According to TMS320AV7100 spec, the EXTWAITZ signal must assert before the number of wait states (can be set up in TMS320AV7100's wait state register) expires. The default wait state number is 3.

  From the TMS320AV7100 data sheet, it can be seen that the delay time from CSZ going low to EXTWAITZ going low is a maximum of 30ns and the default wait state is one. This may cause a problem to the TSB12LV41, because in the worst case, the above delay time is hard to meet. In order to let TSB12LV41 work with TMS320AV7100 seamlessly, it is recommended that after power up, the application software should change the wait state number from one to four in the TMS320AV7100 ARM core. Otherwise, handshake mode may fail to function. Blind Access Mode is unaffected by the wait state setting.

- EXTWAITZ Sharing Issue.

  Because the TMS320AV7100 only has one EXTWAITZ input signal line, the TSB12LV41 may have to share this pin with other devices on the bus. Since EXTWAITZ has been defined as an open-drain type in the TMS320AV7100, users have to set the IOCR RdyPushPull bit correctly to get the correct signal level output from the TSB12LV41. If the TSB12LV41 is the only device connected to the EXTWAITZ on the TMS320AV7100 extension bus, then the user can set the RdyPushPull bit to 1. However, if there are other devices connected to the EXTWAITZ line then the user must set RdyPushPull to 0 to avoid bus contention on this pin. In this case the user must add an external pull-up resistor on the line connecting EXTWAITZ to WAITZ. See Figure 1.

- INTZ (Interrupt) Output Line Sharing Issue.

  The TMS320AV7100 has dedicated INT lines, therefore no sharing of this signal is needed. The TSB12LV41 outputs normal totem-pole signal levels for this pin.

Also, the application can clear the interrupt by writing a "1" to each interrupt bit that is *active* in the Interrupt Registers. This can be done without using the TMS320AV7100's EXTACK signal (interrupt acknowledge).

TMS320AV7100 Blind Access Mode

The Blind Access Mode is designed for fast microprocessors, which do not have the external wait/ready line handshake signal. It can potentially provide faster performance on the microprocessor interface since it eliminates the need for the processor to wait for an acknowledge signal from the link in order to terminate the cycle. The embedded ARM processor in the TMS320AV7100 can use either Handshake or Blind Access Mode.

Two Registers are involved in Blind Access Mode, BASTAT (Blind Access Status Register, at Address 1F0h) and BAHR (Blind Access Holding Register, at Address 1F4h).

If the bit BAcmp is set to 1 in the BASTAT register, this indicates that the current Blind Access read or write operation is finished. For Blind Read, it means the data from the requested address is available in the BAHR register. For a Blind Write, it means the incoming data has been delivered to the requested address.

Once the BAcmp bit has been set, reading to either BASTAT or BAHR will cause the BAcmp bit to be cleared, regardless of whether the current Blind Access is a read or a write.

For those consecutive read/write operations whose access time is longer than the read/write latency (and a new Blind Write/Read is issued without reading the BASTAT to check BAcmp status), then, for Blind write, the last write to fill up the whole quadlet will clear the BAcmp. For a Blind read, the first read transaction will clear the BAcmp.

Blind Access Write Operations

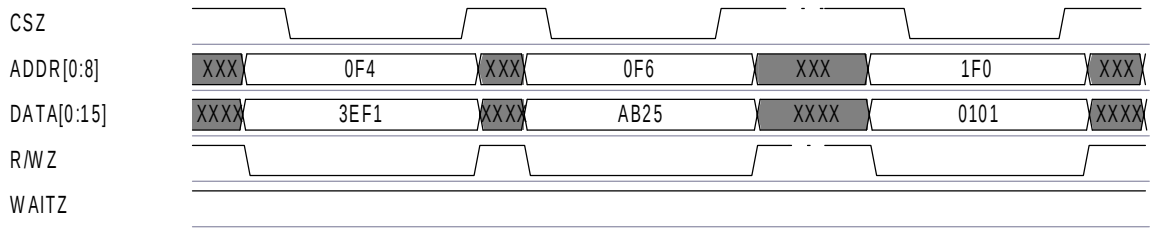
In the example below, the TMS320AV7100 embedded ARM processor wants to write hex value 3EF1AB25h to TSB12LV41 Register 0F4h (DSS Formatter Control Register) and will use Blind Access Mode.

This operation assumes the IOCR register (1ECh) setting is 4E000000h, which corresponds to the bit settings shown in Table 8.

Table 8. IOCR Register Settings

Bit #	Bit Name	Bit Value	Bit Meaning
0	MCMP8	0	Word Access
1	BeCtl	1	Big Endian
2	IntPol	0	Low True
3	RdyPushPull	0	High Impedance State
4	IntPushPull	1	Active Push/Pull
5	BlindAccess	1	Enable Blind Access Mode
6	DataInvarnt	1	Data Invariant
7	RdyPol	0	Low True
8-31			Reserved

Figure 8. TSB12LV41/TMS320AV7100 Blind Access Write Timing Example



Note: The BCLK signal is not shown in the above diagram for purposes of clarity, but is used in this mode.

After this operation the contents of Register 0F4h are as shown below:

Table 9. Register 0F4h Contents

MSByte			LSByte
Bits 0-7	Bits 8-15	Bits 16-23	Bits 24-31
3E	F1	AB	25

The functional differences between this mode and a Write operation using Handshake mode are:

1) There is no handshake signal passed from the TSB12LV41 to the host to signal that the write operation is complete (the WAITZ line stays high).

2) The host must read the Blind Access Status Register (Register 1F0h) in order to detect that the write is complete.

Blind Writes - Notes

- If the time period of consecutive write operations is greater than the write latency time, then the **BAcmp** bit doesn't need

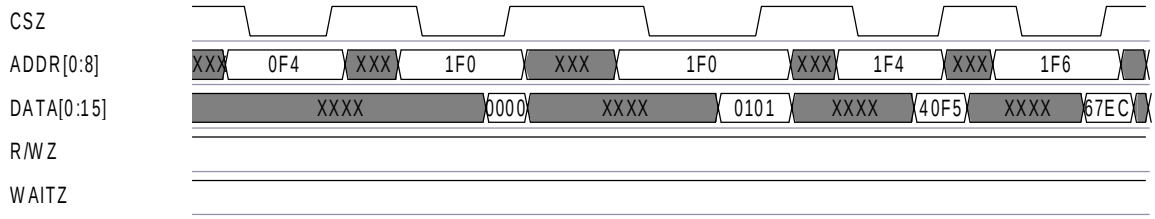
to be checked. The write latency will be a maximum of 19 BCLK cycles (see Figure 2).

- By polling the BAcmp bit in the BASTAT register, the CPU can detect whether the current blind access is finished or not.
- Before the BAcmp bit is set, all reads and writes are prohibited except for reads to the BASTAT and the BAHR registers. Once the whole quadlet has been filled up with the valid write data, a write to the internal logic will be initiated and can not be stopped except by asserting a device reset. If the CPU mistakenly issues a new write or read before BAcmp is set, a write/read error interrupt will be generated and this new write/read will be aborted. Note that for Blind Access write operations, the first 3 bytes (or the first doublet) of the next address to be written to could be loaded without disturbing the write operation currently in progress. This is because the last byte or doublet loaded initiates the actual internal write cycle. This could provide some speed improvement on the interface for consecutive writes.

Blind Access Read Operations

Figure 9. TSB12LV41/TMS320AV7100 Blind Access Read Timing Example

Note: The BCLK signal is not shown in the above diagram for purposes of clarity, but is used in this mode.



The order of events shown in the above timing diagram is as follows:

1) The host reads address 0F4h. Note that no data is returned on this read.

2) The host reads the Blind Access Status Register, address 1F0, to see if the read is complete. The BACMP bit is ‘0’, thus the read cycle is not complete yet.

3) The host reads the Blind Access Status Register again. This time the BACMP bit is ‘1’ indicating that the 0F4h register data contents are available at the Blind Access Holding Register.

4) The host reads the Blind Access Holding Register, address 1F4h, and retrieves Byte 0 (MSByte) and Byte 1.

5) The host reads the Blind Access Holding Register, address 1F6h, and retrieves Byte 2 and Byte 3 (LSByte).

The functional differences between this mode and a Write operation using Handshake mode are:

- There is no handshake signal passed from the TSB12LV41 to the host to signal that the write operation is complete (the WAITZ line stays high).
- The host must read the Blind Access Status Register (Register 1F0h) in order to detect that the read is complete.
- Once the BAcmp bit is detected high in the Blind Access Status Register, the host can retrieve the data by reading the Blind Access Holding Register (register 1F4h).

Blind Reads - Notes

- The first byte or doublet read will result in a dummy data value. However, it does start the actual read cycle.
- The BAcmp bit in BASTAT can be checked to determine if the read cycle is complete.

- If the time period of consecutive reads is greater than the read latency, then the BAcmp bit does not need to be checked. The BAHR can be directly accessed without reading the BASTAT register.
- Before the BAcmp bit is set, all reads (except reads to BASTAT and BAHR registers) and all writes are prohibited. Once the internal read cycle starts, it can not be stopped except by asserting a device reset. If the CPU mistakenly issues a new write or read before BAcmp is set, a write/read error interrupt (INVWROP bit at address 18h) will be generated and this new write/read will be aborted.
- A follow-up read to BAHR can obtain the requested data after the BAcmp is set.

Important Notes on Blind Access Mode

Blind Read and Blind Write accesses can not be nested inside each other.

Only for TMS320AV7100 in 16-bit Access Mode Read:

- When the first follow-up read to BAHR is finished (which clears the BAcmp bit), a new read to a new address is allowed even before the CPU reads out the next word (doublet) from the BAHR register. The CPU can take advantage of the time slot between the end of the new Blind Read cycle and the moment that the next BAcmp bit is set to grab the second doublet resulting from the previous Blind Read process out from the BAHR.

Only for TMS320AV7100 in 8-bit Access Mode

- In 8-bit access mode, it doesn't matter whether the previous blind access is a read or a write, a new blind write is allowed even if the BAcmp for the previous blind access does not come back. This is due to the fact that the internal full quadlet data transfer will not be generated until the last byte of the quadlet from the new write cycle comes in.
- There are four registers located inside the TSB12LV41's Micro Interface domain: IOCR (I/O Control Register), BASTAT (Blind Access Status Register), BAHR (Blind Access Holding Register), and SRES (Software Reset Register). Accessing these registers in the Blind Access Mode doesn't need to go across the internal synchronization boundary, therefore, the access is immediate and no status check to BASTAT is necessary. (Note that BAcmp bit will not be set.)

Appendix A. TSB12LV41 Endianness

The TSB12LV41 uses the same endianness as the P1394 Link Core, which is Big Endian. Big Endian means the most significant byte is byte 0 at the left-hand side and the least significant byte is byte 3 at right hand side. Please refer to the Figure 1 and Figure 2 below for an illustration of both endianness types.

Figure 10. Big Endian Illustration chart (with Bit 0 as MSB and Bit 31 as LSB)

0			31
Byte#0 (MSByte)	Byte#1	Byte#2	Byte#3 (LSByte)

Figure 11. Little Endian Illustration chart (with Bit 31 as MSB and Bit 0 as LSB)

31			0
Byte#3 (MSByte)	Byte#1	Byte#2	Byte#0 (LSByte)

For the host processor to work correctly with the TSB12LV41, users have to connect their microprocessor's LSB to the TSB12LV41's LSB, and their microprocessor's MSB to the TSB12LV41's MSB, regardless of the bit number labeling and which type of endianness their microprocessor uses. The correct byte swapping can be done by setting TSB12LV41's BeCtl bit (Big Endian Control Bit) in the IOCR Register (register1ECh).

For the read case in byte mode, the upper byte (bit 0-7) of the TSB12LV41's 16-bit data bus contains the byte data, the lower byte (bit 8-15) will be driven to all zeros. In the write case in byte mode, users should put the byte write data in the upper byte (bit 0-7). The data in lower byte has no effect on the data to be written into the TSB12LV41.

NOTES:

1) When the BeCtl bit is set to "1" (Big Endian), the DataInvarnt bit setting has no effect.
2) When the BeCtl bit is set to the opposite value to that of the actual endianness of the microprocessor the TSB12LV41 is connected to, the data result passed through the two different endianness domains is unpredictable.

The TSB12LV41 Microprocessor address bus has the MSB at pin 50 (ADR0) and the LSB at pin 59 (ADR8). The Microprocessor data bus has the MSB at pin 80 (DATA0) and the LSB at pin 61 (DATA15).

TSB12LV41 registers are 32 bits wide (1 quadlet). When using 16 bit data bus with Word access mode (word access set by bit 0 at address 1ec) the least significant bit of the address is ignored. The next consecutive bit determines which doublet, upper or lower, the 16 bit data gets written to or read from.

Big Endian Write Example

A Big Endian processor wants to write the value 3F46CDE2h to address 0F4h. 1st 16bit word written to address 0F4h

Figure 12. Second 16-bit Word Written To Address 0F4h (Address LSB is a don't care)

0 (MSBit)	(LSBit) 15
Byte #0 (MSByte)	Byte #1 (LSByte)
3F	46

Figure 13. Second 16-bit Word Written To Address 0F6h (Address LSB is a don't care)

0 (MSBit)	(LSBit) 15
Byte #2 (MSByte)	Byte #3 (LSByte)
CD	E2

The data in register 0F4h would be:

Figure 14. Data in register 0F4h

0			31
Byte#0 (MSByte)	Byte#1	Byte#2	Byte#3 (LSByte)
3F	46	CD	E2

Big Endian Read Example

Figure 15. Data in Register 0F4h

0			31
Address = 0F4	Address = 0F5	Address = 0F6	Address = 0F7
Byte#0 (MSByte)	Byte#1	Byte#2	Byte#3 (LSByte)
56	CF	3E	7A

The read data would look like the following:

Figure 16. First Read From Address 0F4 (Address LSB is a don't care)

0 (MSBit)	(LSBit) 15
Byte #0 (MSByte)	Byte #1 (LSByte)
56	CF

Figure 17. Second Read From Address 0F6h (Address LSB is a don't care)

0 (MSBit)	(LSBit) 15
Byte #2 (MSByte)	Byte #3 (LSByte)
3E	7A