AMBA Test Interface Controller

Data Sheet

ARM®

DDI 0043E

AMBA Test Interface Controller
Data Sheet

Release Information

The following changes have been made to this book.

Change History

Date	Issue	Confidentiality	Change
December 1995	A	Confidential - Restricted	First issue
January 1996	B	Confidential - Restricted	Minor revisions
March 1996	C	Confidential - Restricted	Major revision
May 1996	D	Confidential - Restricted	Minor revisions
April 1997	E	Confidential - Restricted	Minor revisions

Proprietary Notice

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Product Status

The information in this document is final, that is for a developed product.

Web Address

`http://www.arm.com`

Contents
AMBA Test Interface Controller Data Sheet

Chapter 1
AMBA Test Interface Controller

The Test Interface Controller is a state machine that provides an AMBA bus master for system test. It controls the External Bus Interface (EBI) to sample or drive the ASB data bus **BD**.

This manual contains the following sections:

1.1 Overview

The AMBA Test Interface Controller (TIC) is an ASB bus master that accepts test vectors from the external test bus (the 32-bit external data bus, if available) and initiates bus transfers. The TIC latches address vectors from the test bus and drives the ASB address bus.

Typically, the TIC is the highest priority AMBA bus master to ensure test access under all conditions.

This TIC model does not support control vector/address incrementing: that means that to change the address bus value, a specific address vector needs to be issued.

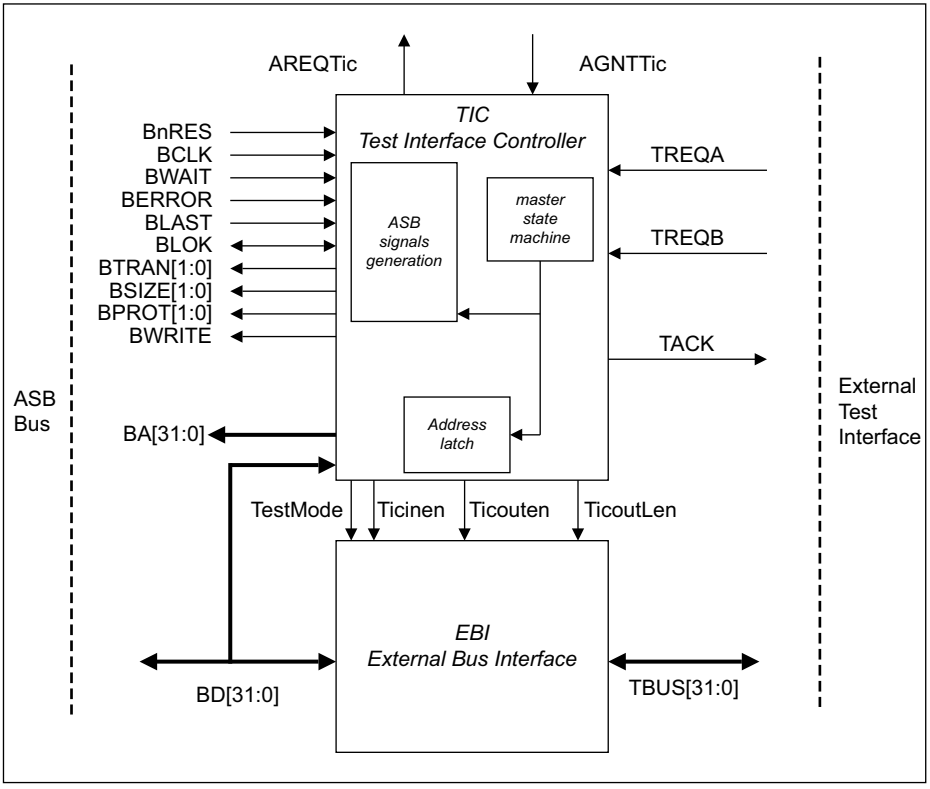

Figure 1-1 Test interface controller block diagram

Figure 1-1 on page 1-2 represents a TIC block in a system where the external 32-bit data bus becomes the test bus when performing test mode accesses. 16-bit and 8-bit data bus systems require, for example, 16 or 24 address lines to be reconfigured as bidirectional test port signals for test mode access. Such systems would use the **TestMode** signal to force the EBI into this state.

1.2 Signal Description

Table 1-1 Signal descriptions

Name	Type	Description
AREQTic	Out	Request from the TIC, indicating that this master requires the bus. This signal must be set up to the falling edge of BCLK. The Arbiter should treat this signal as the highest priority request line (over and above any complex arbitration scheme it might support).
AGNTTic	In	Grant signal that grants the bus to the test controller.
BnRES	In	This active LOW signal indicates the reset status of the bus and is driven by the reset controller.
BCLK	In	System (bus) clock. This clock times all bus transfers. The clock has two distinct phases—phase 1 in which BCLK is LOW and phase 2 in which BCLK is HIGH. In the example system this clock also operates in test mode as TCLK.
BWAIT	In	This signal indicates when current transfer will complete. This signal is valid on the rising edge of BCLK. It must be used in concert with the other slave response lines BERROR and BLAST.
BERROR	In	This signal is used with BWAIT and BLAST to form the retract term in the bus master state machine.
BLAST	In	This signal is used with BERROR and BWAIT to form the retract term.
BLOK	In	Bus lock signal. The TIC does not support locked transfers. Since the TIC should be the highest priority master, its transfers should never be interrupted. Driven LOW when the TIC is granted.
BTRAN[1:0]	Out	These signals indicate the type of the next transaction, which in this master may be Address-only or Sequential. They are valid during the HIGH phase before the transfer to which they refer.
BSIZE[1:0]	Out	These signals indicate the size of the transfer, which for this master is always word (32 bits). These signals have the same timing as the address bus.
BPROT[1:0]	Out	These signals deal with address location access protection control. When generated by the TIC, these signals indicate supervisor mode transfers. They have the same timing as the address bus.
BA[31:0]	Out	System address bus. The addresses become valid during the HIGH phase before the transfer to which they refer and remain valid until the last HIGH phase of the transfer.

Table 1-1 Signal descriptions (continued)

Name	Type	Description
BD[31:0]	In	In test mode, the data bus is used to load values into the TIC's address latch during "address vectors". The TIC then drives the system address bus BA with this value during subsequent single/burst read-write vectors.
BWRITE	Out	When HIGH, this signal indicates a write transfer and when LOW, a read. This signal has the same timing as the address bus.
TREQA	In	Test request A. This signal is used, in combination with TREQB, to control access to the system bus from the test bus.
TREQB	In	Test request B. This signal is used, in combination with TREQA, to control access to the system bus from the test bus.
TACK	Out	Test Acknowledge. This signal is used to indicate that the test interface has been granted access to the system bus. It is also used to indicate transfer delays (ie. transfers with wait cycles).
Ticinen	Out	This active LOW signal indicates that the EBI should drive TBUS onto **DB**.
Ticouten	Out	This active LOW signal indicates that the EBI should drive its latched version of **BD** onto the external TBUS.
TicoutLen	Out	When low the BD latch in the EBI should be transparent.
TestMode	Out	Indicates that the test controller has taken control of the bus. It should be used to enable the external 32-bit test bus (for EBIs that need the TBUS to be specifically enabled) and to select the system clock source for test (TCLK).

1.3 Functional Description

The Test Interface Controller has two fundamental modes of operation:

- default bus master.
- system test.

The default bus master is selected during reset, when an AMBA system is in low power mode (if supported) or when no other masters are requesting the bus. When granted as the default master the TIC performs no test functions but keeps the AMBA bus in a state such that:

- No data transfers occurs (**BTRAN** = A-TRAN).
- The bus is available to be granted to another master when requested (**BLOK** = LOW).

Table 1-2 shows that for system test an external tester must assert the external pins **TREQA** and **TREQB**:

Table 1-2 Basic TIC operation

Inputs		Outputs	
TREQA	**TREQB**	**TACK**	**TIC mode**
0	0	0	Acting as default bus master
1	0	0	Entering test mode
1	1	1	Test mode entered, ADDRESS vector.
X	X	X	During test
0	0	1	Leaving test mode

Note

During test at least one of **TREQA** and **TREQB** must be HIGH. If both are LOW this indicates end of test.

When entering test mode the TIC requests the bus. Once it is granted as the current bus master test mode is entered and the **TACK** signal is pulled HIGH.

1.3.1 Clocking issues

When not in test mode an AMBA system is clocked by **BCLK**. The source of **BCLK** is not defined in the AMBA standard and may be from an off-chip source or more likely an on-chip PLL for low power consumption. In both cases **TREQA** is synchronised in the TIC by the rising edge of **BCLK**.

When test mode is entered the AMBA system will be clocked by **TCLK**, generated from an external source. Then **TREQA** and **TREQB** signals should be setup to the rising edge of **TCLK**.

The switch over from an internal **BCLK** source to an external **TCLK** source is not handled by the TIC. If an external **BCLK** source is used then **BCLK** and **TCLK** can be identical.

Note

The Example AMBA System uses an externally generated **BCLK** (ie. **BCLK=TCLK**).

This data sheet details one form of TIC that assumes an external clock can be used to drive the system during test. If this is not feasible it is possible for a TIC to run vectors at a much slower rate than **BCLK**. This is not considered further in this document.

1.3.2 Test Mode

In test mode the signal **TestMode** is driven HIGH. This forces **BCLK** to be driven from an external source (**TCLK**) and the External Bus Interface to provide a 32-bit bidirectional channel through which values can be read and written to **BD**. This 32-bit channel is referred to as **TBUS**, though in systems with a 32-bit external data bus **TBUS** will be identical to the external data bus. In AMBA systems which do not have a full 32-bit external data bus, address pins may have special test functionality (ie. connecting bidirectional PAD's that can act as inputs during system test) to provide a **TBUS** connection.

The **TREQA** and **TREQB** signals should both be high on entering test mode. They are used to control the TIC which allows values to be read or written to any address location inside the microcontroller (it can not perform read or writes to external memory as the external bus is occupied by the test bus, **TBUS**). This is done by applying TIC vectors, of which there are three basic types; READ, WRITE and ADDRESS. Before a READ or WRITE vector can access a location the appropriate address must have been loaded. Thus at the beginning of testing the first vector should be an ADDRESS type.

Table 1-3 TIC vectors

TREQA	TREQB	Vector
1	1	ADDRESS. Must be first and last vector in a test sequence, also used as a turn around cycle after a write vector.
1	0	WRITE. Must be followed by a turn around cycle.
0	1	READ.
0	0	End of test. Must be proceeded by an ADDRESS vector.

Vectors are applied in a pipelined fashion. In Figure 1-2 vector 2 is applied while the AMBA transfer triggered by vector 1 is occurring. Then in the following cycle the transfer for vector 2 occurs (the diagram shows an extremely simplified case).

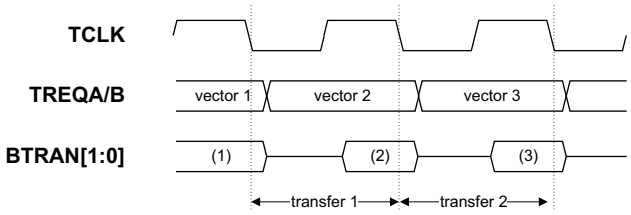

Figure 1-2 TIC vectors and AMBA transfers

When the vectors are applied the **TACK** signal should be monitored. This will normally be HIGH. However, if the transfer initiated by the previous vector is not complete, then **TACK** will go LOW off the falling edge of **TCLK**. When this occurs, the next vector should be held until **TACK** goes high. This is shown in Figure 1-3 on page 1-9. Vector 1 starts a transfer with one wait cycle; vector 2 is held while the transfer completes.

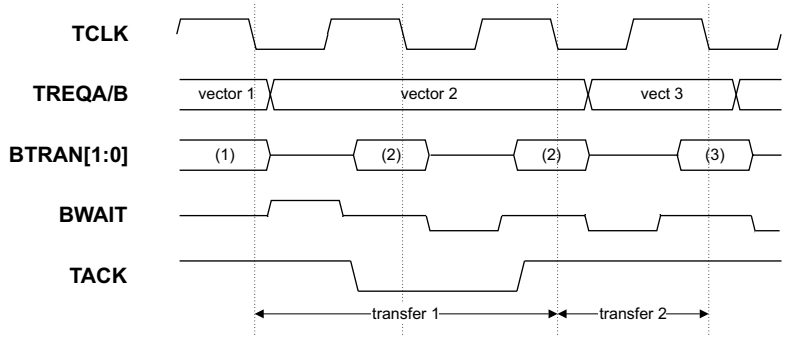

Figure 1-3 Vectors and waited transfers

Although all vector types are applied as above, the timing characteristics for **TBUS** and the number of each vector type applied is vector dependant.

ADDRESS vector

Only one ADDRESS vector is required for an address transfer. The **TBUS** must be driven with the address value required during the ADDRESS vector transfer (ie. transfer 2 shown above). This value is latched from **BD** by the TIC, driven on to **BA** by the end of the ADDRESS vector transfer and is held for subsequent READ or WRITE vectors.

During an ADDRESS vector the TIC drives **BTRAN** as A-TRAN, thus no locations are accessed in the microcontroller.

WRITE vector

Only one WRITE vector is required for each write transfer. As with the ADDRESS vector; the value to be written should be driven on **TBUS** during the transfer for the WRITE vector. The TIC drives **BTRAN** as S-TRAN and asserts **BWRITE**, causing a write transfer to the address location setup by the last ADDRESS vector.

READ vector

Unlike ADDRESS or WRITE vectors the READ vector **TBUS** activity does not take place during its corresponding AMBA transfer. During the read transfer period the **TBUS** should be undriven. The value read by the READ vector is driven out on **TBUS** in the cycle following the transfer.

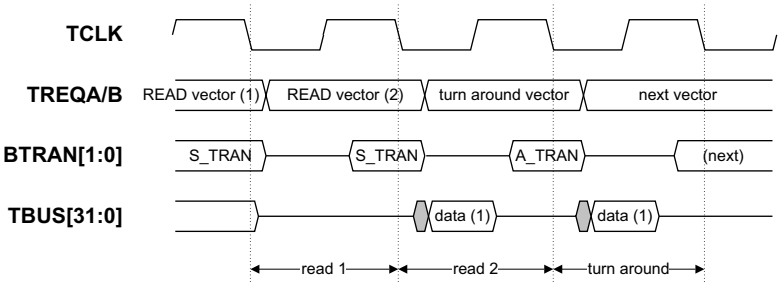

Figure 1-4 Read vectors and turn around

Here two reads are done (from the same address location). Since **TBUS** is driven in the cycle following the read 2 transfer; the READ vector 2 can not be followed by a WRITE or and ADDRESS vector (this would require **TBUS** to be driven by the tester in the cycle following the read 2 transfer). Thus a 'turn around vector' is needed. Turn around is indicated by **TREQA**=1, **TREQB**=1, which is identical to an ADDRESS vector. However, no address change occurs since **TBUS** is not driven and **BD** is not latched onto **BA**. The turn around vector may be followed by ADDRESS vectors (or any other type of vector) in which case the address will change in subsequent cycles.