

## Features

- Links an Embedded ARM7TDMI™ Core to the Atmel AMBA™ Bus
- Bus Master Granted State Machine
- Bus Interface State Machine
- Fully Scan Testable (up to 96% Fault Coverage)

## Description

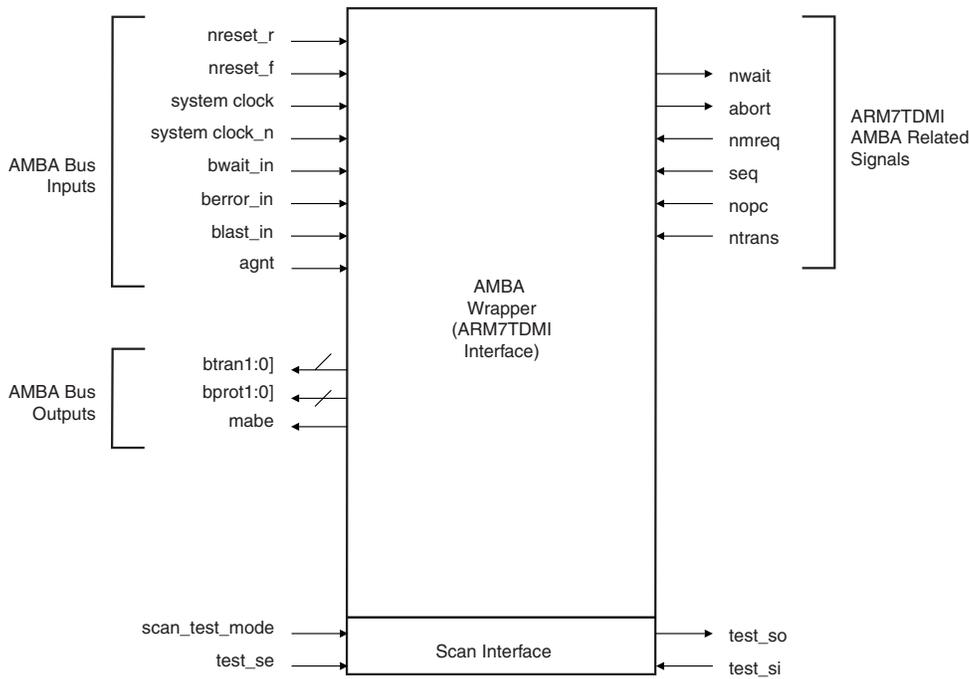
Designed for the Atmel implementation of the AMBA Bus, the ARM7TDMI Bus Interface module enables an ARM7TDMI embedded core to become an AMBA bus master.

The bus interface is designed to link the embedded ARM7TDMI core signals to the AMBA Bus. It includes two state machines. The first state machine determines if the master is currently granted the bus, and the second, more complex, state machine is used to control the bus interface of the master.

This peripheral can only be used with an embedded ARM7TDMI core.

There are no user-programmable registers in this block.

Figure 1. ARM7TDMI Bus Interface Pin Configuration



## ARM7TDMI™ Bus Interface

## 32-bit Embedded Core Interface





**Table 1.** Bus Interface Pin Description

Name	Direction	Source/Destination	Description
<b>AMBA Bus Inputs</b>			
nreset_r	Input	Reset Controller	System reset for flip-flop working on the rising edge of the System Clock (nreset_r synchronous to nclock). Active low.
nreset_f	Input	Reset Controller	System reset for flip-flop working on the rising edge of the inverted System Clock (nreset_f synchronous to clock). Active low.
clock	Input		System Clock
nclock	Input		Inverted System Clock
bwait_in	Input	Slaves + decoder	Multiplexed BWAIT response from all the peripherals. Active high.
berror_in	Input	Slaves + decoder	Multiplexed BERROR response from all the peripherals. Active high.
blast_in	Input	Slaves + decoder	Multiplexed BLAST response from all the peripherals. Active high.
agnt	Input	Arbiter	Bus Grant: indicates that the bus master (ARM7TDMI) will be granted the bus when BWAIT is low.
<b>AMBA Bus Outputs</b>			
btran[1:0]	Output	Current bus master/decoder, slaves	Transfer type: type of the next transaction (Address-only, Non-sequential or Sequential). To be connected to the tri-state bus BTRAN[1:0] of the ASB.
bprot[1:0]	Output	Current bus master	Protection control: indicates if the transfer is an opcode fetch or data access, as well as if the transfer is a supervisor mode access or a user mode access. To be connected to the tri-state bus BPROT[1:0] of the ASB.
mabe	Output		Master address bus enable: indicates when BA bus can be taken into account. Active high.
<b>ARM7TDMI AMBA Related Signals</b>			
nwait	Output	ARM7TDMI	Not wait: used when accessing slow peripherals to make the processor wait. Active Low.
abort	Output	ARM7TDMI	Memory abort: tells the processor that a requested access is not allowed. Active High.
nMREQ	Input	ARM7TDMI	Not memory request: memory access requested by the processor. Active Low.
seq	Input	ARM7TDMI	Sequential address: next address (memory access) will be related to the last memory access. Active High.
nopc	Input	ARM7TDMI	Not op-code fetch: indicates that the processor is fetching an instruction from the memory. Active Low.
ntrans	Input	ARM7TDMI	Not memory translate: indicates that the processor is in user mode. Active Low.
<b>Scan Test</b>			
scan_test_mode	Input		Must be tied to 1 during scan test Must be tied to 0 in functional mode
test_se	Input		Test scan shift enabled when tied to 1
test_si <sup>(1)</sup>	Input		Test scan input (input of the scan chain)
test_so <sup>(1)</sup>	Output		Test scan output (output of the scan chain)

Note: 1. The scan chain uses the clock BCLK.

## Scan Test Configuration

The coverage is maximum if all non-scan inputs can be controlled and all non-scan outputs can be observed. In order to achieve this, the ATPG vectors must be generated on the entire circuit (top level) which includes the ARM7TDMI Bus Interface or all ARM7TDMI Bus Interface I/Os must have a top level access and ATPG vectors must be applied to these pins.

## Operating in an AMBA Bus System

**Figure 2.** An Example of ARM7TDMI Core and Bus Interface Implementation

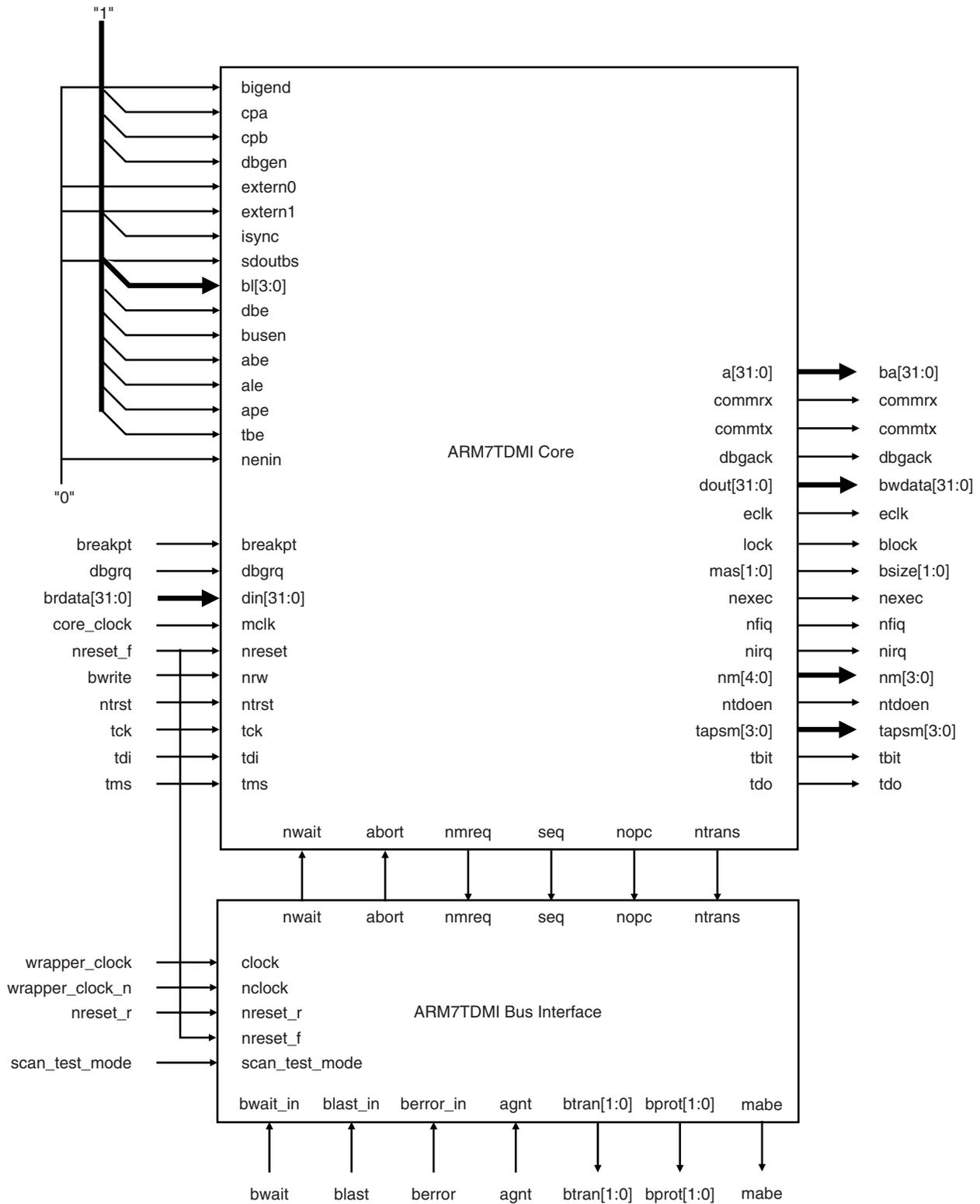
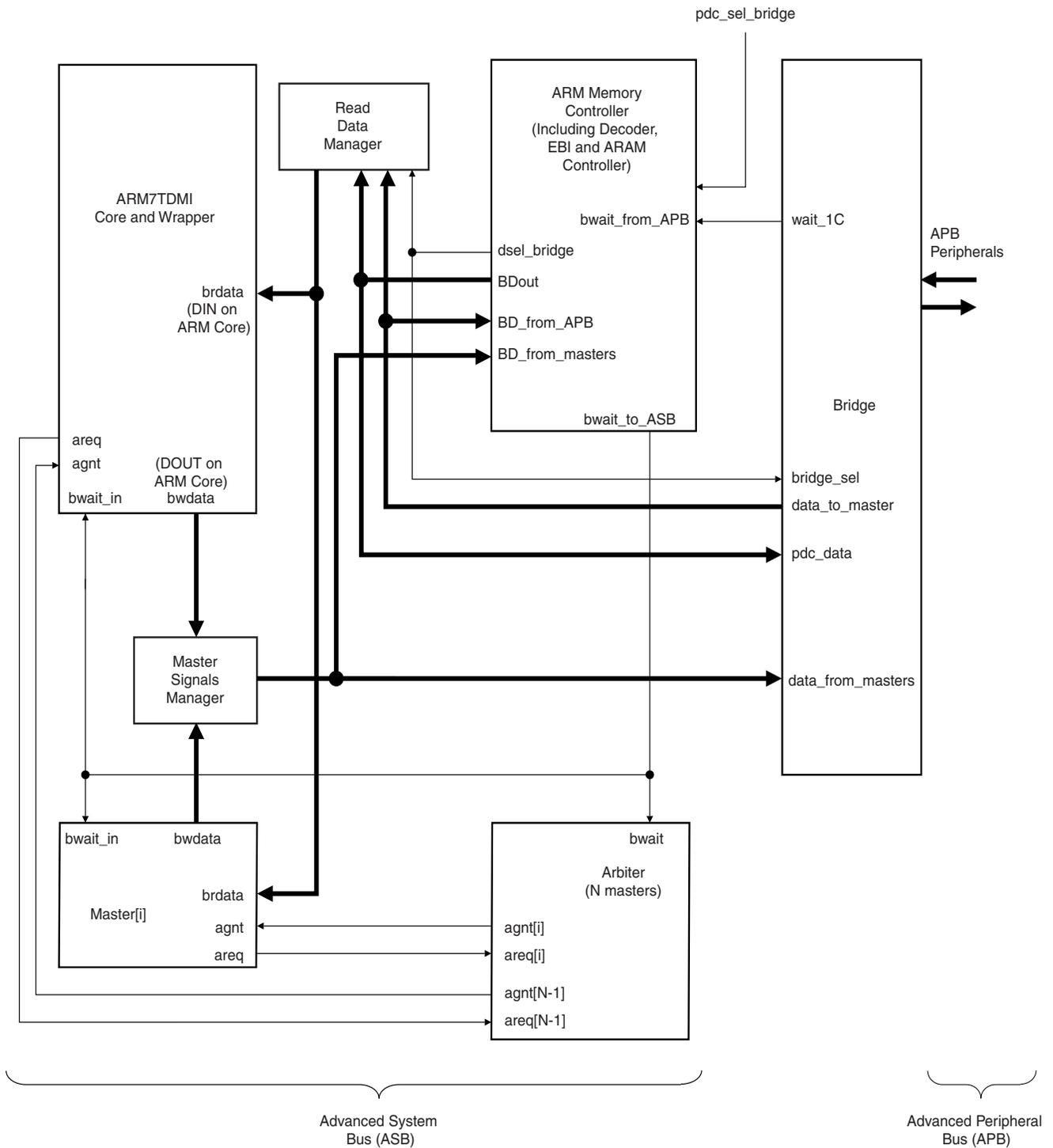


Figure 3. Data Buses





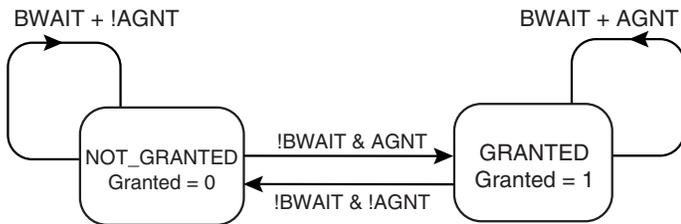
## Bus Master Interface Description

The bus master interface consists of a state machine that is used to determine if the master is currently granted the bus, and to control the bus interface of the master.

### Granted State Machine

The granted state machine is used to determine whether or not the bus master has been granted the bus. It runs on the rising edge of the system clock and has only two states GRANTED and NOT\_GRANTED. The state diagram is shown below.

**Figure 5.** Bus Master Granted State Machine



The output from the state machine is the internal **Granted** signal, which is used in the main bus master state machine. It is important to note that the **agnt** signal may be asserted for a number of clock cycles, but it is only when **agnt** is asserted and **bwait** is low that the bus master actually becomes GRANTED.

An important design consideration is that the state machine may be asynchronously reset into either state depending on the value of the **agnt** signal. During reset one bus master in the system is set as the default bus master, as indicated by **agnt** being asserted during reset, and will be reset in to the GRANTED state. All other bus masters will be reset into the NOT\_GRANTED state.

## Bus Interface State Machine

The main bus interface state machine is falling edge triggered and contains six states. The entire state diagram, as shown in Figure 6, is quite complex but can be considered in four quadrants:

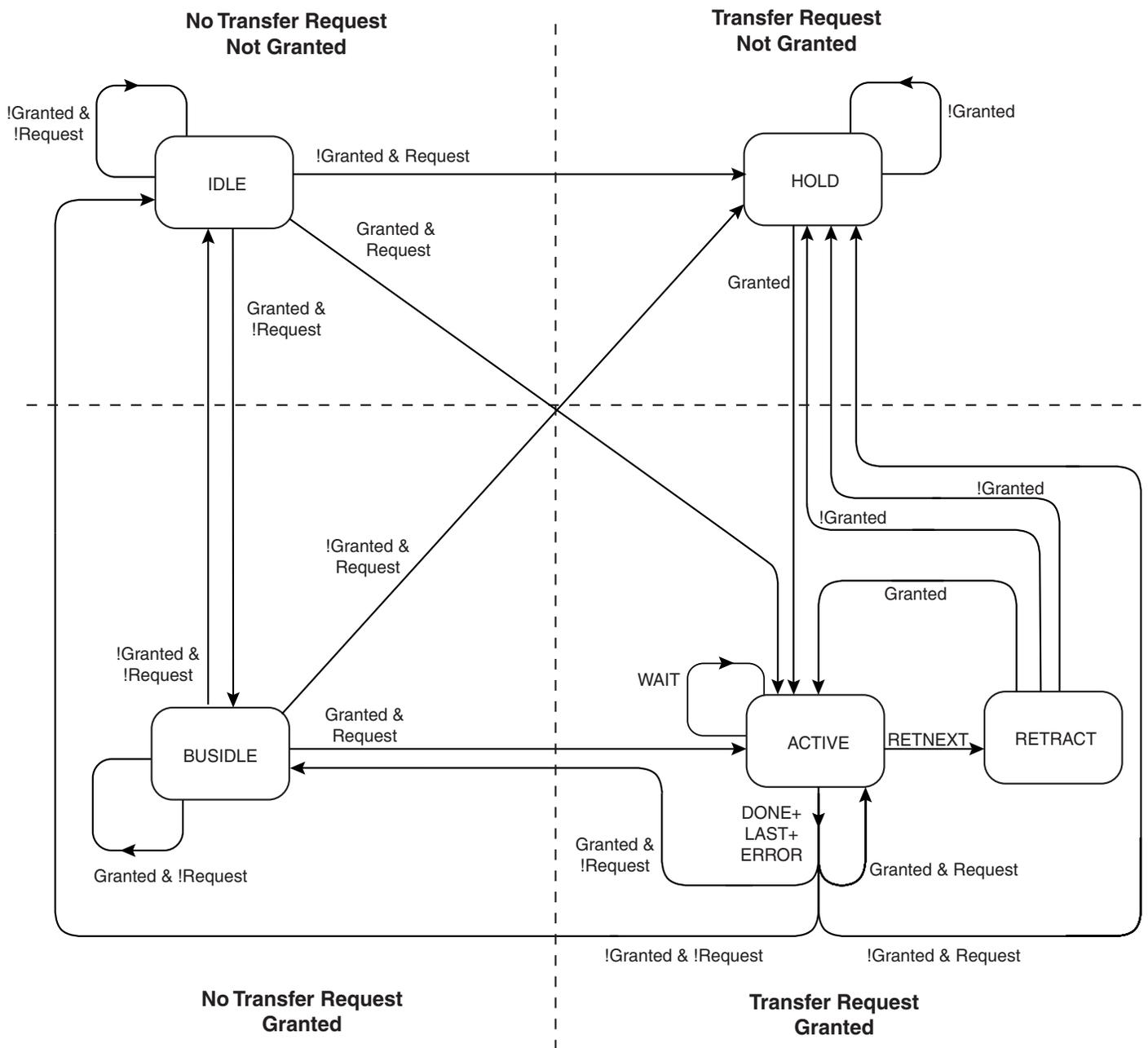
No Transfer Request Not Granted	Transfer Request Not Granted
No Transfer Request Granted	Transfer Request Granted

The “Transfer Request, Granted” quadrant contains three states, which handle bus turn around and the retract operation.

The two internal bus master signals **Granted** and **Request** control the majority of the transitions around the state diagram. **Granted** is generated from the simpler state machine described above and **Request** is generated directly by the bus master. **Request** is asserted high when the bus master requires a transfer on the bus and is low when the bus master does not need access to the bus.

The only time when a transition around the state diagram is not controlled by **Granted** and **Request** is when the bus master is in the ACTIVE state. In this state the transition to the next state is determined by the transfer response that is received. WAIT, DONE, LAST, ERROR and RETNEXT shown in the diagram correspond to the encoding of the transfer response signals.

Figure 6. Bus Master Main State Machine



Note that the state diagram assumes that once the bus master has made a request for a transfer, as indicated by **Request**, then **Request** remains asserted until the bus master has performed a transfer.

As the main bus master state machine operates from the falling edge of the clock, it is necessary to use latched versions of the transfer response signals **bwait**, **berror** and **blast** to control the exit from the ACTIVE state.

The reset conditions are not shown on the state diagram and, in a similar manner to the granted state machine, the main bus master state machine has a complex reset term.

If **agnt** is asserted during reset, when **nreset** is low, the bus master is the default bus master and enters the IDLE state. However, if **agnt** is not asserted during reset then the bus master enters the IDLE state.

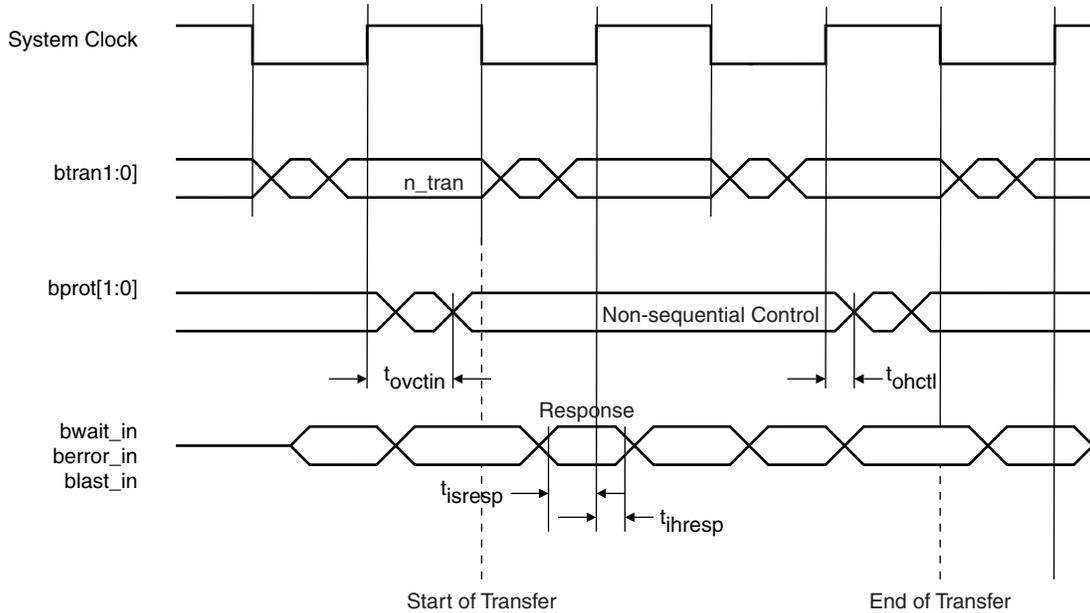
The following table indicates the actions that must occur in each state.

Name	Description	Actions
IDLE	The master does not require the bus and it is not granted.	Internal btran is Address-only Master clock is enabled Master address bus is tri-stated
BUSIDLE	The master does not require the bus, but it has been granted anyway.	Internal btran as indicated by master Master clock is enabled Master address bus enable is generated from Granted signal
HOLD	The master requires the bus, but it has not been granted.	Internal btran is Address-only Master clock is disabled Master address bus is tri-stated
ACTIVE	Active state when data transfers occur. Exiting this state is dependent on the transfer response.	Internal btran as indicated by master Master clock enable is derived from BWAIT Master address bus enable is generated from Granted signal
RETRACT	Retract state, where the rest of the elements in the system see the transfer finish, but the bus master is not advanced.	Internal btran is Address-only Master clock is disabled Master address bus enable is generated from Granted signal

## Timing Diagrams

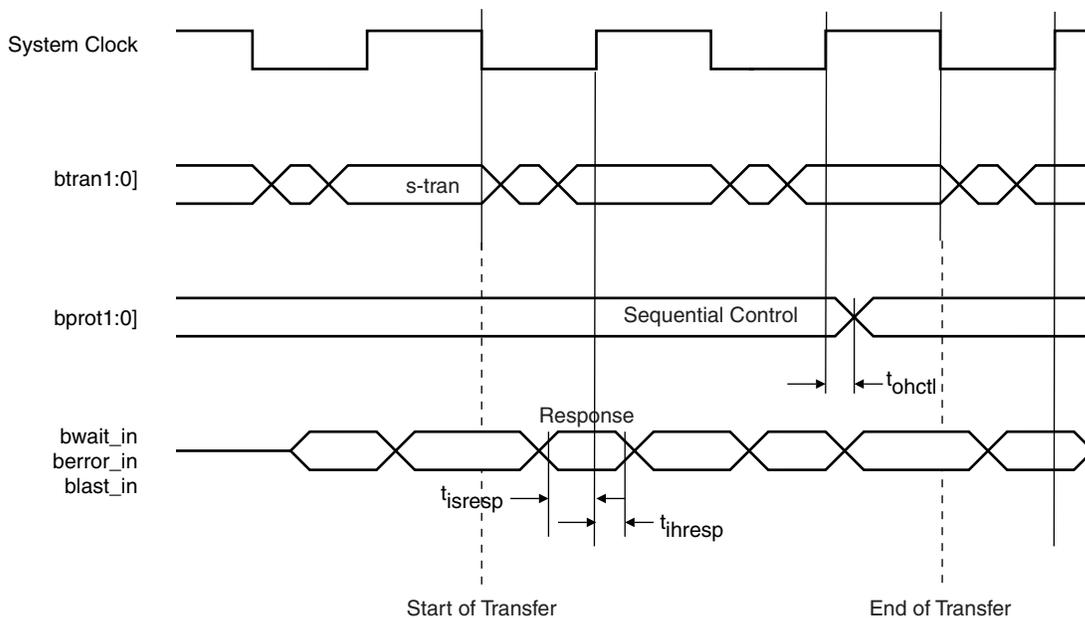
The following diagrams show the timings for the parameters described in this datasheet. For complete timing diagrams, see the "System Architecture" datasheet.

**Figure 7.** ASB Bus Master Non-sequential Transfer



For the non-sequential transfer shown above, the address and control signals become valid in the system clock high phase before the start of the transfer. An important feature of the AMBA protocol is that it allows for poor output valid times on non-sequential transfers, which is provided through the automatic insertion of a wait state at the start of every non-sequential transfer by the decoder.

**Figure 8.** ASB Bus Master Sequential Transfer

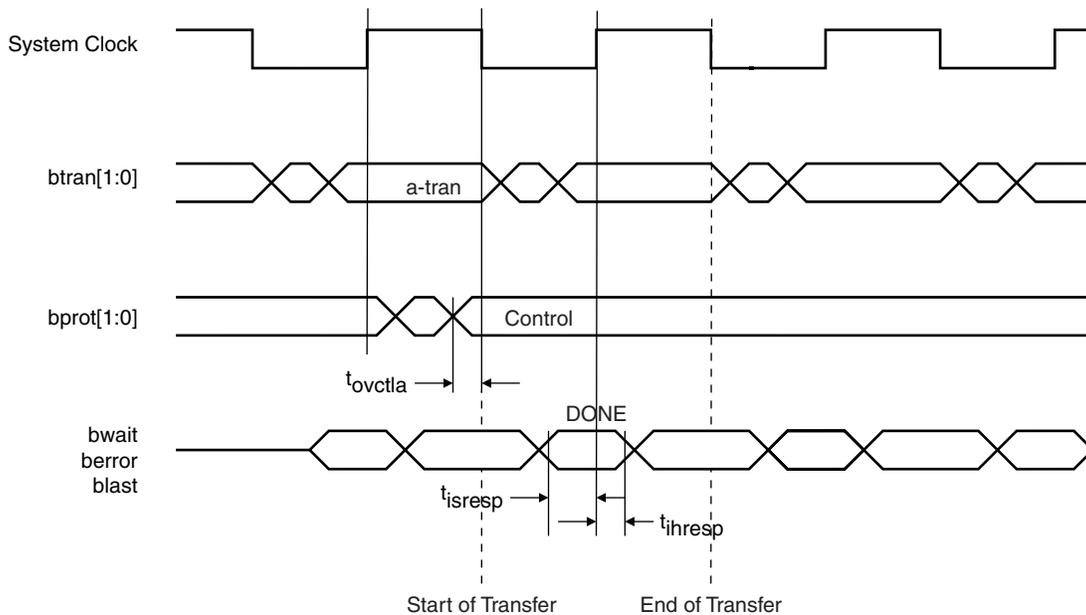


A sequential transfer has different timing parameters for the address and control signal valid times. In a typical bus master, the output valid times for sequential transfers are far better than for non-sequential transfers. The output hold times for address, control and data are identical and independent of the transfer type.

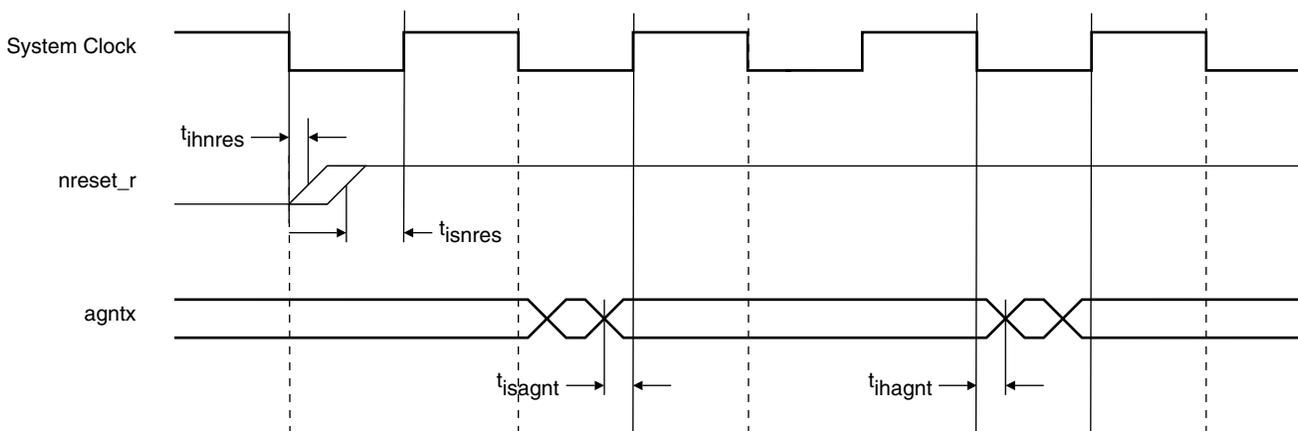
The other difference between the sequential and non-sequential transfers is that during a sequential transfer the data may be driven during the first phase of the transfer and hence the data valid parameter is specified from the falling edge of the system clock.

For an Address-only transfer the address and control signals may be driven in the clock high phase before the start of the transfer, or in the case of bus master hand-over may only be driven during the clock low phase of the transfer itself. The address and control valid timing parameters are only relevant when the Address-only transfer is followed immediately by a sequential transfer and in this case the address and control signals must be driven such that they are valid during the low phase of the Address-only transfer, which in turn means they are valid throughout the clock high phase that precedes the Sequential transfer.

**Figure 9.** ASB Master Address-only Transfer



**Figure 10.** ASB Bus Master Arbitration and Reset Signals



The **nreset\_r** signal may be asserted asynchronously and so there is no setup and hold parameter relating to the assertion of the signal. The **AGNT** signal, which is returned from the arbiter changes during the low clock phase.

## Timing Parameters

The timing parameters related to an ASB bus master operating in an AMBA system are also shown in textual form in the following two tables. The first details the input signals, while the second details output signals.

**Table 2.** Bus Master Input Timing Parameters

Parameter	Description
$t_{isnres}$	nreset de-asserted setup to rising system clock
$t_{ihnres}$	nreset de-asserted hold after falling system clock
$t_{isresp}$	bwait, berror and blast setup to rising system clock
$t_{ihresp}$	bwait, berror and blast hold after rising system clock
$t_{isagnt}$	agnt setup to rising system clock
$t_{ihagnt}$	agnt hold after falling system clock

**Table 3.** Bus Master Output Timing Parameters

Parameter	Description
$t_{ovctin}$	For Non-sequential transfers, bprot[1:0] valid after rising system clock
$t_{ovctla}$	For Address-only transfers, bprot[1:0] valid after falling system clock
$t_{ohctl}$	bprot[1:0] hold after rising system clock



## Atmel Headquarters

*Corporate Headquarters*  
2325 Orchard Parkway  
San Jose, CA 95131  
TEL (408) 441-0311  
FAX (408) 487-2600

### *Europe*

Atmel U.K., Ltd.  
Coliseum Business Centre  
Riverside Way  
Camberley, Surrey GU15 3YL  
England  
TEL (44) 1276-686-677  
FAX (44) 1276-686-697

### *Asia*

Atmel Asia, Ltd.  
Room 1219  
Chinachem Golden Plaza  
77 Mody Road Tsimhatsui  
East Kowloon  
Hong Kong  
TEL (852) 2721-9778  
FAX (852) 2722-1369

### *Japan*

Atmel Japan K.K.  
9F, Tonetsu Shinkawa Bldg.  
1-24-8 Shinkawa  
Chuo-ku, Tokyo 104-0033  
Japan  
TEL (81) 3-3523-3551  
FAX (81) 3-3523-7581

## Atmel Operations

*Atmel Colorado Springs*  
1150 E. Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906  
TEL (719) 576-3300  
FAX (719) 540-1759

### *Atmel Rousset*

Zone Industrielle  
13106 Rousset Cedex  
France  
TEL (33) 4-4253-6000  
FAX (33) 4-4253-6001

---

### *Fax-on-Demand*

North America:  
1-(800) 292-8635  
International:  
1-(408) 441-0732

### *e-mail*

[literature@atmel.com](mailto:literature@atmel.com)

### *Web Site*

<http://www.atmel.com>

### *BBS*

1-(408) 436-4309

#### © Atmel Corporation 2000.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.



Printed on recycled paper.

1283B-05/00/0M