

# AMBA™ 3 APB Protocol

v1.0

## Specification



# AMBA 3 APB Protocol Specification

Copyright © 2003, 2004. ARM Limited. All rights reserved.

## Release Information

### Change history

Date	Issue	Change
25 September 2003	A	First release for v1.0
17 August 2004	B	Second release for v1.0

## Proprietary Notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM Limited in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM Limited in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM Limited shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

## AMBA Specification License

1. Subject to the provisions of Clauses 2 and 3, ARM hereby grants to LICENSEE a perpetual, non-exclusive, nontransferable, royalty free, worldwide licence to use and copy the AMBA Specification for the purpose of developing, having developed, manufacturing, having manufactured, offering to sell, selling, supplying or otherwise distributing products which comply with the AMBA Specification.

2. THE AMBA SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF SATISFACTORY QUALITY, MERCHANTABILITY, NONINFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE.

3. No licence, express, implied or otherwise, is granted to LICENSEE, under the provisions of Clause 1, to use the ARM tradename, or AMBA trademark in connection with the AMBA Specification or any products based thereon. Nothing in Clause 1 shall be construed as authority for LICENSEE to make any representations on behalf of ARM in respect of the AMBA Specification.

**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 3 APB Protocol Specification

	<b>Preface</b>	
	About this specification .....	x
	Feedback .....	xiii
<b>Chapter 1</b>	<b>Introduction</b>	
	1.1 About the AMBA 3 APB .....	1-2
	1.2 Changes for AMBA 3 APB Protocol Specification v1.0 .....	1-3
<b>Chapter 2</b>	<b>Transfers</b>	
	2.1 Write transfers .....	2-2
	2.2 Read transfers .....	2-4
	2.3 Error response .....	2-6
<b>Chapter 3</b>	<b>Operating States</b>	
	3.1 Operating states .....	3-2
<b>Chapter 4</b>	<b>Signal Descriptions</b>	
	4.1 AMBA 3 APB signals .....	4-2



# List of Figures

## AMBA 3 APB Protocol Specification

	Key to timing diagram conventions .....	xi
Figure 2-1	Write transfer with no wait states .....	2-2
Figure 2-2	Write transfer with wait states .....	2-3
Figure 2-3	Read transfer with no wait states .....	2-4
Figure 2-4	Read transfer with wait states .....	2-5
Figure 2-5	Example failing write transfer .....	2-6
Figure 2-6	Example failing read transfer .....	2-7
Figure 3-1	State diagram .....	3-2





# Preface

This preface introduces the *Advanced Microcontroller Bus Architecture* (AMBA) 3 *Advanced Peripheral Bus* (APB) protocol specification. It contains the following sections:

- *About this specification* on page x
- *Feedback* on page xiii.

## About this specification

This is the specification for the AMBA 3 APB protocol. All references to APB in this manual refer to AMBA 3 (not AMBA 2 or earlier versions).

## Intended audience

This specification is written to help hardware and software engineers to design systems and modules that are compatible with the APB protocol.

## Using this specification

This specification is organized into the following chapters:

### **Chapter 1** *Introduction*

Read this chapter for an overview of the APB protocol.

### **Chapter 2** *Transfers*

Read this chapter for information about the different types of APB transfer.

### **Chapter 3** *Operating States*

Read this chapter for descriptions of the APB operating states.

### **Chapter 4** *Signal Descriptions*

Read this chapter for descriptions of the APB signals.

## Conventions

This section describes the conventions that this specification uses:

- *Typographical*
- *Timing diagrams* on page xi
- *Signals* on page xii.

### **Typographical**

This specification uses the following typographical conventions:

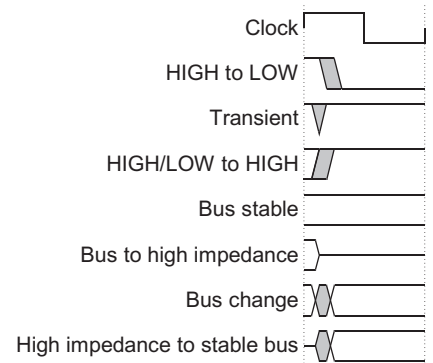
<b><i>italic</i></b>	Highlights important notes, introduces special terminology, denotes internal cross-references, and citations.
<b>bold</b>	Highlights interface elements, such as menu names. Denotes ARM processor signal names. Also used for terms in descriptive lists, where appropriate.

<code>monospace</code>	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.
<u><code>monospace</code></u>	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<i><code>monospace italic</code></i>	Denotes arguments to monospace text where the argument is to be replaced by a specific value.
<b><code>monospace bold</code></b>	denotes language keywords when used outside example code.
<b>&lt; and &gt;</b>	Angle brackets enclose replaceable terms for assembler syntax where they appear in code or code fragments. They appear in normal font in running text. For example: <ul style="list-style-type: none"> <li>MRC p15, 0 &lt;Rd&gt;, &lt;CRn&gt;, &lt;CRm&gt;, &lt;Opcode_2&gt;</li> <li>The Opcode_2 value selects which register is accessed.</li> </ul>

## Timing diagrams

The figure named *Key to timing diagram conventions* explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.



**Key to timing diagram conventions**

## Signals

The signal conventions are:

<b>Signal level</b>	The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means HIGH for active-HIGH signals and LOW for active-LOW signals.
<b>Prefix P</b>	Denotes AMBA 3 APB signals.
<b>Suffix n</b>	Denotes AXI, AHB, and AMBA 3 APB reset signals.

## Further reading

This section lists publications that provide additional information about the AMBA 3 protocol family.

ARM periodically provides updates and corrections to its documentation. See <http://www.arm.com> for current errata sheets, addenda, and the Frequently Asked Questions list.

This document contains information that is specific to the APB interface. See the following documents for other relevant information:

- *AMBA AXI Protocol Specification* (ARM IHI 0022).

## Feedback

ARM Limited welcomes feedback on the APB protocol and its documentation.

### Feedback on the product

If you have any comments or suggestions about this product, contact your supplier giving:

- the product name
- a concise explanation of your comments.

### Feedback on this specification

If you have any comments on this specification, send email to [errata@arm.com](mailto:errata@arm.com) giving:

- the title
- the number
- the relevant page number(s) to which your comments apply
- a concise explanation of your comments.

ARM Limited also welcomes general suggestions for additions and improvements.



# Chapter 1

## Introduction

This chapter provides an overview of the AMBA 3 APB. It contains the following section:

- *About the AMBA 3 APB* on page 1-2
- *Changes for AMBA 3 APB Protocol Specification v1.0* on page 1-3.

## 1.1 About the AMBA 3 APB

The APB is part of the AMBA 3 protocol family. It provides a low-cost interface that is optimized for minimal power consumption and reduced interface complexity. The APB interfaces to any peripherals that are low-bandwidth and do not require the high performance of a pipelined bus interface. The APB has unpipelined protocol.

All signal transitions are only related to the rising edge of the clock to enable the integration of APB peripherals easily into any design flow. Every transfer takes at least two cycles.

The APB can interface with the AMBA *Advanced High-performance Bus Lite* (AHB-Lite) and AMBA *Advanced Extensible Interface* (AXI). You can use it to provide access to the programmable control registers of peripheral devices.



## 1.2 Changes for AMBA 3 APB Protocol Specification v1.0

This version includes:

- A ready signal, **PREADY**, to extend an APB transfer. See Chapter 2 *Transfers*.
- An error signal, **PSLVERR**, to indicate the failure of a transfer. See *Error response* on page 2-6.



# Chapter 2

## Transfers

This chapter describes typical AMBA 3 APB write and read transfers, and the error response. It contains the following sections:

- *Write transfers* on page 2-2
- *Read transfers* on page 2-4
- *Error response* on page 2-6.

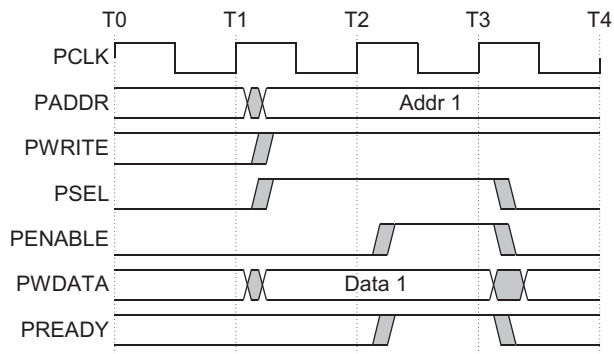
## 2.1 Write transfers

Two types of write transfer are described in this section:

- *With no wait states*
- *With wait states.*

### 2.1.1 With no wait states

Figure 2-1 shows a basic write transfer with no wait states.



**Figure 2-1 Write transfer with no wait states**

The write transfer starts with the address, write data, write signal and select signal all changing after the rising edge of the clock. The first clock cycle of the transfer is called the Setup phase. After the following clock edge the enable signal is asserted, **PENABLE**, and this indicates that the Access phase is taking place. The address, data and control signals all remain valid throughout the Access phase. The transfer completes at the end of this cycle.

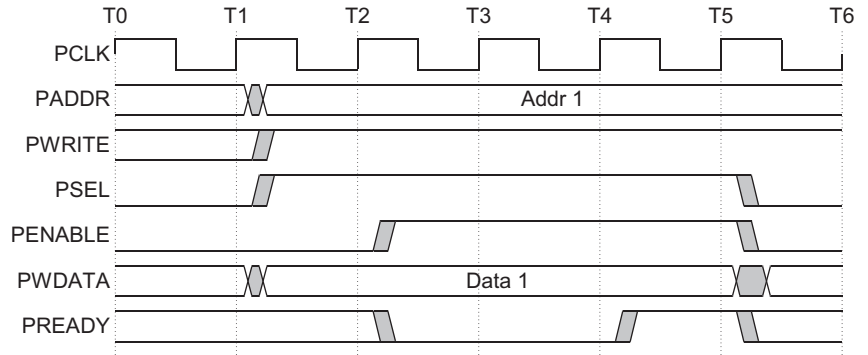
The enable signal, **PENABLE**, is deasserted at the end of the transfer. The select signal, **PSELx**, also goes LOW unless the transfer is to be followed immediately by another transfer to the same peripheral.

### 2.1.2 With wait states

Figure 2-2 on page 2-3 shows how the **PREADY** signal from the slave can extend the transfer. During an Access phase, when **PENABLE** is HIGH, the transfer can be extended by driving **PREADY** LOW. The following signals remain unchanged for the additional cycles:

- address, **PADDR**
- write signal, **PWRITE**

- select signal, **PSEL**
- enable signal, **PENABLE**
- write data, **PWDATA**.



**Figure 2-2 Write transfer with wait states**

**PREADY** can take any value when **PENABLE** is LOW. This ensures that peripherals that have a fixed two cycle access can tie **PREADY** HIGH.

**Note**

It is recommended that the address and write signals are not changed immediately after a transfer but remain stable until another access occurs. This reduces power consumption.

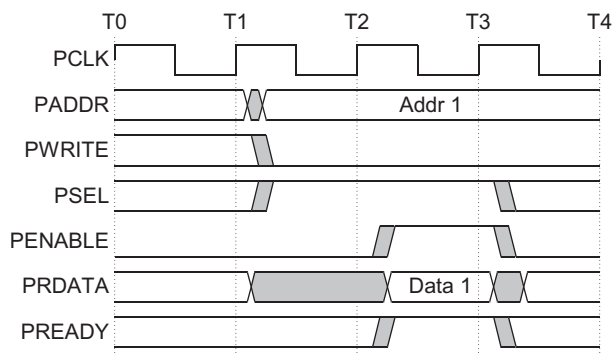
## 2.2 Read transfers

Two types of read transfer are described in this section:

- *With no wait states*
- *With wait states.*

### 2.2.1 With no wait states

Figure 2-3 shows a read transfer. The timing of the address, write, select, and enable signals are as described in *Write transfers* on page 2-2. The slave must provide the data before the end of the read transfer.



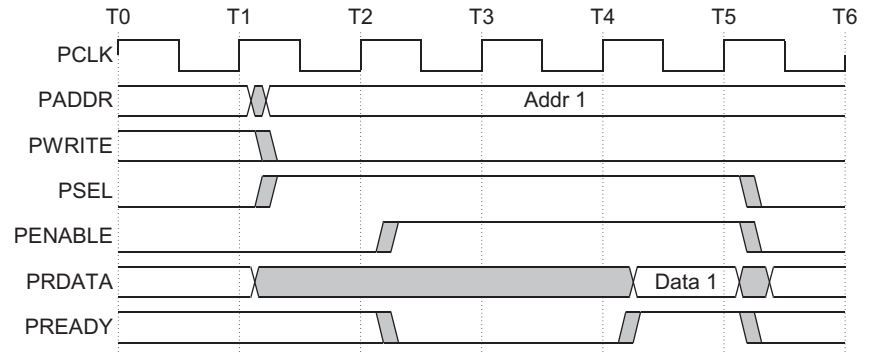
**Figure 2-3 Read transfer with no wait states**

### 2.2.2 With wait states

Figure 2-4 on page 2-5 shows how the **PREADY** signal can extend the transfer. The transfer is extended if **PREADY** is driven LOW during an Access phase. The protocol ensures that the following remain unchanged for the additional cycles:

- address, **PADDR**
- write signal, **PWRITE**
- select signal, **PSEL**
- enable signal, **PENABLE**.

Figure 2-4 on page 2-5 shows that two cycles are added using the **PREADY** signal. However, you can add any number of additional cycles, from zero upwards.

**Figure 2-4 Read transfer with wait states**

## 2.3 Error response

You can use **PSLVERR** to indicate an error condition on an APB transfer. Error conditions can occur on both read and write transactions.

**PSLVERR** is only considered valid during the last cycle of an APB transfer, when **PSEL**, **PENABLE**, and **PREADY** are all HIGH.

It is recommended, but not mandatory, that you drive **PSLVERR** LOW when it is not being sampled. That is, when any of **PSEL**, **PENABLE**, or **PREADY** are LOW.

Transactions that receive an error, might or might not have changed the state of the peripheral. This is peripheral-specific and either is acceptable. When a write transaction receives an error this does not mean that the register within the peripheral has not been updated. Read transactions that receive an error can return invalid data. There is no requirement for the peripheral to drive the data bus to all 0s for a read error.

APB peripherals are not required to support the **PSLVERR** pin. This is true for both existing and new APB peripheral designs. Where a peripheral does not include this pin then the appropriate input to the APB bridge is tied LOW.

### 2.3.1 Write transfer

Figure 2-5 shows an example of a failing write transfer that completes with an error.

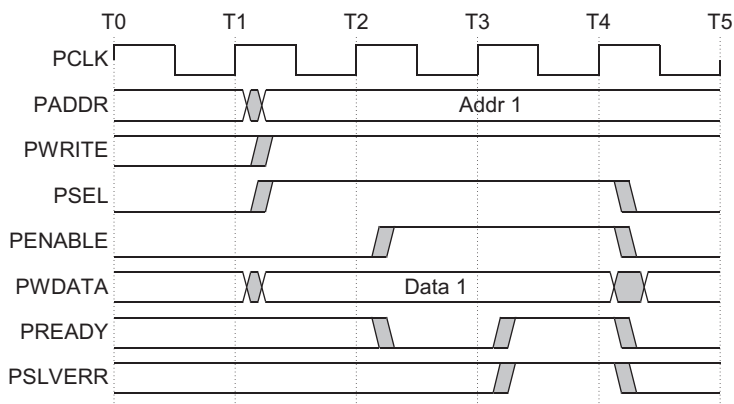


Figure 2-5 Example failing write transfer



### 2.3.2 Read transfer

A read transfer can also complete with an error response, indicating that there is no valid read data available. Figure 2-6 on page 2-7 shows a read transfer completing with an error response.

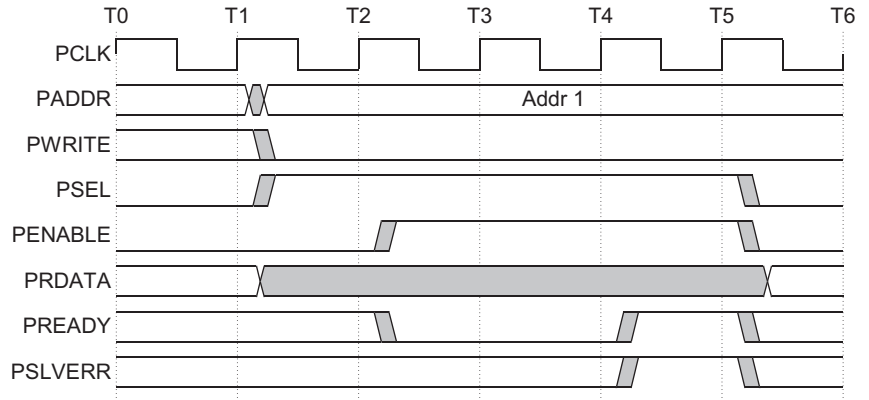


Figure 2-6 Example failing read transfer

### 2.3.3 Mapping of PSLVERR

When bridging:

**From AXI to APB** An APB error is mapped back to **RRESP/BRESP = SLVERR**. This is achieved by mapping **PSLVERR** to the AXI signals **RRESP[1]** for reads and **BRESP[1]** for writes.

**From AHB to APB** **PSLVERR** is mapped back to **HRESP = ERROR** for both reads and writes. This is achieved by mapping **PSLVERR** to the AHB signal **HRESP[0]**.



# Chapter 3

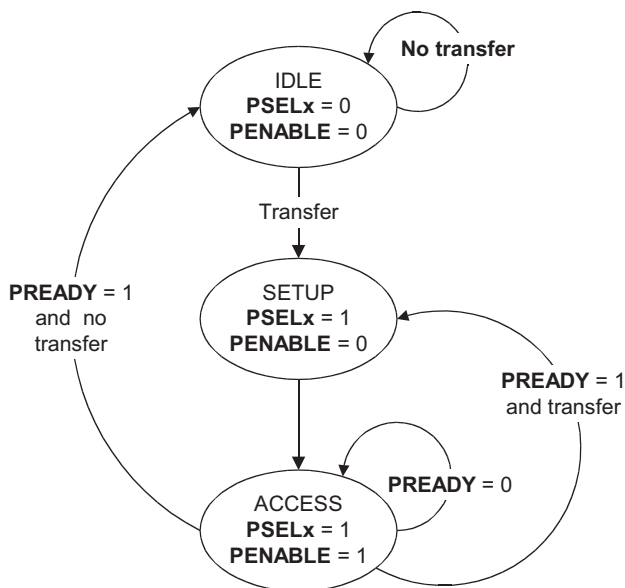
## Operating States

This chapter describes the AMBA 3 APB operating states. It contains the following section:

- *Operating states* on page 3-2.

### 3.1 Operating states

Figure 3-1 shows the operational activity of the APB.



**Figure 3-1 State diagram**

The state machine operates through the following states:

- |               |   |
|---------------|---|
| <b>IDLE</b>   | This is the default state of the APB.   |
| <b>SETUP</b>  | When a transfer is required the bus moves into the SETUP state, where the appropriate select signal, <b>PSELx</b> , is asserted. The bus only remains in the SETUP state for one clock cycle and always moves to the ACCESS state on the next rising edge of the clock.   |
| <b>ACCESS</b> | <p>The enable signal, <b>PENABLE</b>, is asserted in the ACCESS state. The address, write, select, and write data signals must remain stable during the transition from the SETUP to ACCESS state.</p> <p>Exit from the ACCESS state is controlled by the <b>PREADY</b> signal from the slave:</p> <ul style="list-style-type: none"> <li>If <b>PREADY</b> is held LOW by the slave then the peripheral bus remains in the ACCESS state.</li> </ul> |

- If **PREADY** is driven HIGH by the slave then the ACCESS state is exited and the bus returns to the IDLE state if no more transfers are required. Alternatively, the bus moves directly to the SETUP state if another transfer follows.



# Chapter 4

## Signal Descriptions

This chapter describes the AMBA 3 APB signals. It contains the following section:

- *AMBA 3 APB signals* on page 4-2.

4.1 AMBA 3 APB signals

Table 4-1 lists the APB signal descriptions.

Table 4-1 APB signal descriptions

Signal	Source	Description
PCLK	Clock source	Clock. The rising edge of <b>PCLK</b> times all transfers on the APB.
PRESETn	System bus equivalent	Reset. The APB reset signal is active LOW. This signal is normally connected directly to the system bus reset signal.
PADDR	APB bridge	Address. This is the APB address bus. It can be up to 32 bits wide and is driven by the peripheral bus bridge unit.
PSELx	APB bridge	Select. The APB bridge unit generates this signal to each peripheral bus slave. It indicates that the slave device is selected and that a data transfer is required. There is a <b>PSELx</b> signal for each slave.
PENABLE	APB bridge	Enable. This signal indicates the second and subsequent cycles of an APB transfer.
PWRITE	APB bridge	Direction. This signal indicates an APB write access when HIGH and an APB read access when LOW.
PWDATA	APB bridge	Write data. This bus is driven by the peripheral bus bridge unit during write cycles when <b>PWRITE</b> is HIGH. This bus can be up to 32 bits wide.
PREADY	Slave interface	Ready. The slave uses this signal to extend an APB transfer.
PRDATA	Slave interface	Read Data. The selected slave drives this bus during read cycles when <b>PWRITE</b> is LOW. This bus can be up to 32-bits wide.
PSLVERR	Slave interface	This signal indicates a transfer failure. APB peripherals are not required to support the <b>PSLVERR</b> pin. This is true for both existing and new APB peripheral designs. Where a peripheral does not include this pin then the appropriate input to the APB bridge is tied LOW.



# Index

## A

ACCESS 3-2

## C

Changes for this version 1-3

Conventions

signal naming xii

timing diagram xi

typographical x

## E

Error response 2-6

Error response, read transfer 2-7

Error response, write transfer 2-6

Errors, mapping 2-7

## I

IDLE 3-2

## N

No wait states

read 2-4

write 2-2

## O

Operating states 3-2

ACCESS 3-2

diagram 3-2

IDLE 3-2

SETUP 3-2

Overview 1-2

## P

**PADDR** 4-2

**PCLK** 4-2

**PENABLE** 4-2

**PRDATA** 4-2

**PREADY** 4-2

**PRESETn** 4-2

**PSELx** 4-2

**PSLVERR** 4-2

**PWDATA** 4-2

**PWRITE** 4-2

## R

Read errors 2-7

Read transfers

with no wait states 2-4

with wait states 2-4

## S

SETUP 3-2  
Signal descriptions 4-2  
Signal naming conventions xii  
State machine 3-2

## T

Timing diagram conventions xi  
Typographical conventions x

## W

Wait states  
    read 2-4  
    write 2-2  
Write errors 2-6  
Write transfers  
    with no wait states 2-2  
    with wait states 2-2