DESIGN DOCUMENT

**Advance Peripheral Bus 3**

## 

**INTRODUCTION**:

The Advanced Peripheral Bus protocol is a widely used interface protocol for connecting low-speed peripherals to a system bus. The APB is a part of the AMBA 3 protocol family. This document outlines the implementation details of the APB3 protocol for integration into a digital system's design.

**PROTOCOL OVERVIEW:**

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 high performance of a pipelined bus interface. The APB has an unpipelined protocol.

It is a synchronous protocol and hence, all signal transitions are only related to the rising edge of the clock. It requires a minimum of 2 clock cycles to complete a data transfer.

**DESIGN SPECIFICATIONS:**

Here, 8-bit address buses are taken so 256 address locations are accessible. A slave will contain 256 address locations. The data bus will be of 8 bit. Only one slave select signal will be taken because only one slave will be implemented. These signals can be changed as per further scope.

While performing any transfer, conditions for signals to remain stable are as follows:

1. When PSEL is low (design in ideal state), PADDR, PWRITE, PSLVERR and PREADY signal can have any value and design will not have any effect of these signals.
2. When PSEL is asserted (set-up state), PADDR, PWDATA and PWRITE should become stable as per the transfer required (PWRITE remains 0 for read operation and 1 for write operation).
3. PENABLE should get asserted on next clock cycle of PSEL getting asserted.
4. Whenever PENABLE is asserted in access state, PREADY should get asserted if slave is ready to complete the transfer or should be de-asserted if wait state is generated.
5. Data transfer should be performed by the slave as soon as it asserts the PREADY signal.
6. Whenever PREADY is asserted, PSLVERR should become 0. If no error generated else should be 1 for any error.
7. Whenever a transfer is being completed, PENABLE should be de-asserted and PSEL should remain high if there is any further transfer or else should get de-asserted.

When continuous transfer is required, following steps should be performed:

1. For continuous transfer, a counter is taken containing the number of transfers required. Master will decide which type of transfer (read/write) should be performed and how many times. After completion of each transfer, the counter will be decreased by 1.
   1. So, when counter == 0 (when all transfers are completed), design goes to IDLE state.
   2. If PSLVERR is asserted when PSEL, PENABLE, PREADY are HIGH then PSLVERR should be valid. It is only considered at the last transfer of the cycle.

**SIGNAL DESCRIPTION:**

**PCLK:**

SOURCE:Clock Source

Every transfer takes place on the positive edge of the clock. Design works at 10 MHz frequency.

**PRESETn:**

SOURCE:Master interface

Here, suffix n suggests that reset signal is active low.

**PADDR:**

SOURCE:Master interface

This is the APB address bus. Here, the address width is 8 bits wide.

**PSELx:**

SOURCE:Master interface

The Master interface 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 are x PSELx signals for x slaves. Here, only one slave select signal will be there.

**PENABLE:**

SOURCE:Master interface

This signal indicates the second and subsequent cycles of an APB transfer.

**PWRITE:**

SOURCE:Master interface

This signal indicates an APB write access when HIGH and an APB read access when LOW.

**PWDATA:**

SOURCE:Master interface

This bus is driven during write cycles when PWRITE is HIGH. Here, the bus will be 8 bits wide.

**PREADY:**

SOURCE:Slave interface

It will be driven by the slave whenever it is ready to transfer.

**PRDATA:**

SOURCE:Slave interface

The selected slave drives this bus during read cycles when PWRITE is LOW. Here, the bus will be 8 bits wide.

**PSLVERR:**

SOURCE:Slave interface

This signal indicates a transfer failure.

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

**OPERATING STATES:**

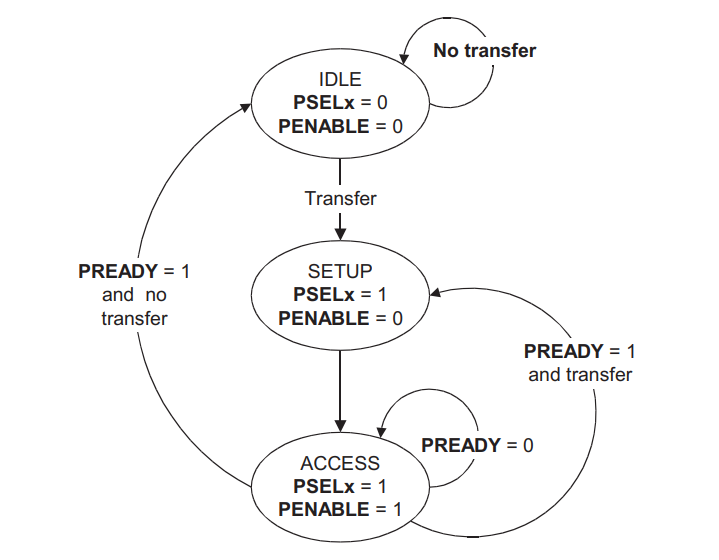


Figure1

The state machine operates through the following states.

**IDLE:**

This is the default state of the APB. Here PSELx and PENABLE are both LOW.

**SETUP:**

When a transfer is required the bus moves into the SETUP state, where the appropriate select signal PSELx 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.

**ACCESS:**

The enable signal PENABLE 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. Exit from the ACCESS state is controlled by the PREADY signal from the slave. If PREADY is held LOW by the slave, then the peripheral bus remains in the ACCESS state. 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.

**READ TRANSFERS WITH NO WAIT STATE:**

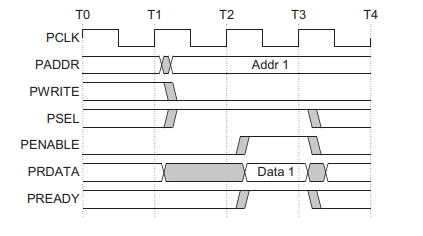


Figure 2

Here, as soon as PSEL is asserted, PADDR and PWRITE should become stable. When PREADY is asserted, data is written on the bus by the slave and PSLVERR will also made stable (0 if no error else 1).

**READ TRANSFER WITH WAIT STATE:**

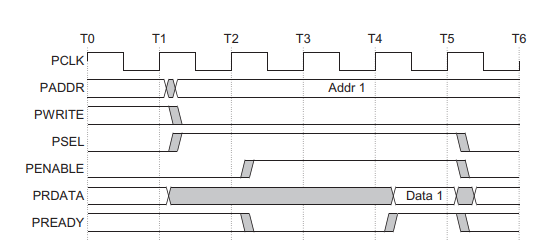


Figure 3

Here, as soon as PSEL is asserted, PADDR and PWRITE should become stable. As soon as PENABLE is asserted, PREADY remains low and read transfer goes into wait state. When PREADY is asserted, data is written on the bus by the slave and PSLVERR will also become low as no error will be generated.

**READ TRANSFER WITH FAILING STATE:**

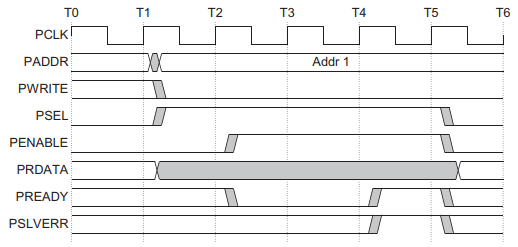


Figure 4

Here, as soon as PSEL is asserted, PADDR and PWRITE should become stable. As soon as PENABLE is asserted, PREADY remains low and read transfer goes into wait state. When PREADY is asserted, data is written on the bus by the slave and PSLVERR will also become high showing error is generated.

**WRITE TRANSFER WITH NO WAIT STATE:**

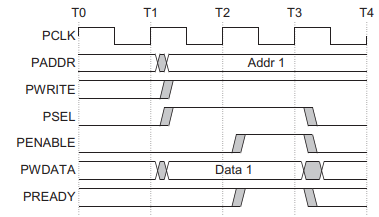


Figure 5

Here, as soon as PENABLE is asserted, PREADY is also asserted by slave in access state and data is written by the slave at the respective address.

**WRITE TRANSFER WITH WAIT STATE:**

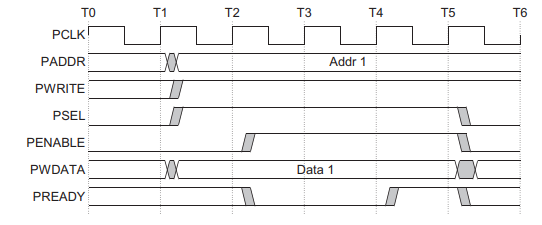


Figure 6

Here, PENABLE is asserted but PREADY is not asserted by the slave at that moment the write transfer goes into wait state.

**WRITE TRANSFER WITH FAILING STATE:**

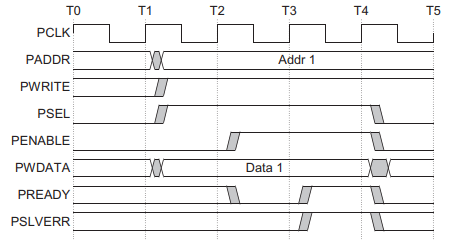


Figure 7

Here, PSLVERR is asserted during the write transfer by the slave to indicate an error scenario.

**FAILURE CASES WITH RESPECT TO STATES:**

**SETUP:**

* If PSEL is de-asserted go back to ideal state. (Error message will generated and master will go back to IDEAL state).
* If PENABLE is not asserted just after one clock edge of PSEL (Protocol is violated so an error message will be generated and will go to IDEAL state)
* If data, address or write signal is not stable before assertion of PENABLE, error message will be displayed, and transfer will be completed.

**ACCESS:**

* After completion of data transfer if PENABLE is not de-asserted. (DUT will throw an error message and master will go back to IDEAL state).
* If any PSEL, PENABLE or both are de-asserted while waiting for wait state. (Protocol is violated so error message will be generated, and if PSEL is de-asserted then design will go back to IDEAL state and if only PENABLE is de-asserted then design will go back to SETUP state).
* When waiting for PREADY for ‘n’ time duration and still PREADY is not asserted (error message will be generated and design will go back to IDEAL state). Here, n is 10 clock cycle.
* If data, address or write signal is not stable before assertion of PREADY while in wait state, error message will be displayed, and transfer will be completed.

**Slave Error Cases:**

* Out of range address operation.
* Writing operation at a read only location.
* Reading operation at a write only location.

**TESTCASES:**

1. Initial reset to bring the design into known state.
2. Write operation at boundary addresses.
3. Read operation at boundary addresses.
4. Write operation with wait state at random address.
5. Read operation with wait state at random address.
6. Back-to-back write and read operations.
7. Multiple write operations to check that data is overwritten or not.
8. Multiple read operations.
9. In between reset.
10. Writing at read only address and reading again to see if data is written or not.
11. Reading from write only address.
12. If PREADY is not asserted while in wait state during write/read transfer.
13. While in wait state, PENABLE is de-asserted.
14. While in wait state, only PSEL is de-asserted.
15. While in wait state, address is changed or PWRITE is toggled.
16. De-asserting PSEL when in SETUP state.
17. Content of PADDR is changed or PWRITE is toggled while in SETUP state.
18. While continuous transfer, after one transfer is done, PENABLE is not de-asserted.
19. PENABLE is not asserted exactly after one clock cycle when PSEL is asserted.