APB bus

NXSOL-OJT-2024

Exported on 2024-01-17 17:52:44

Table of Contents

1 APB 3

2 Prior knowledge 4

2.1 Timing diagram conventions 4

2.2 APB signal descriptions 4

2.3 2.1 Write Transfers 4

2.3.1 2.1.1 With no wait states 5

2.3.2 2.1.2 With wait states 5

2.4 2.2 Read transfers(same as write) 6

2.4.1 2.2.1 With no wait states 6

2.4.2 2.2.2 With wait states 6

2.5 2.3 Error response 6

2.5.1 2.3.1 Failing write transfer 7

2.5.2 2.3.2 Failing read transfer(same as above) 7

2.5.3 2.3.3 Mapping of PSLVERR(note) 7

2.6 3.1 Operating States 8

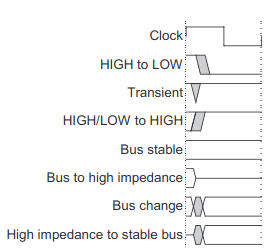
# APB

The APB (Advanced Peripheral Bus) is part of the AMBA (Advanced Microcontroller Bus Architecture) suite of protocols.  
AMBA is a widely used interconnection standard for connecting functional blocks (like CPUs, memory units, and peripherals).

* **Purpose**: APB is designed for low-bandwidth control accesses, for example, peripheral device control. Its primary use is to connect simple peripherals to the primary bus, which could be an AHB (Advanced High-performance Bus) or AXI (Advanced eXtensible Interface) bus in the AMBA standard.
* **Simplicity and Efficiency**: The APB is simpler compared to other buses in the AMBA family. It is optimized for minimal power consumption and reduced complexity of peripherals. This makes it ideal for simple, lower-speed tasks where high throughput is not a critical requirement.
* **Use Cases**: APB is commonly used for interfacing with low-speed peripherals like keyboards, mouse devices, UARTs, SPI (Serial Peripheral Interface) controllers(control signals don't require high speed).

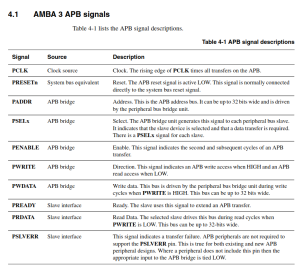
# Prior knowledge

## Timing diagram conventions



* **Bus change**: This indicates that multiple bits on the bus are changing simultaneously aside from just a single binary digit changing
* **Transient**: This is a temporary, intermediate state during the transition of a signal from one stable state to another. It's a period where the signal is neither high nor low
* **Bus to high impedance**: This transition shows the bus moving to a high impedance state, which is effectively an electrical disconnection. This state is used when multiple devices can drive the bus, but only one should do so at a time to avoid conflicts.

## APB signal descriptions

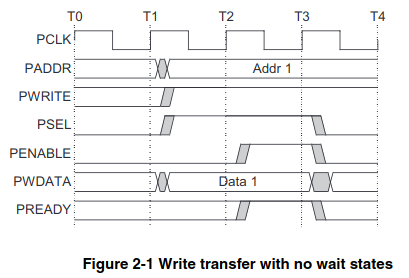


## 2.1 Write Transfers

Write transfers can be categorized into two types:

* With no wait states
* With wait states

### 2.1.1 With no wait states

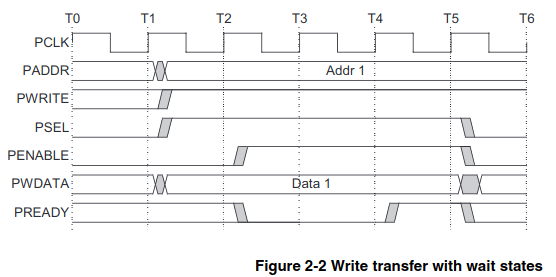


This figure illustrates a basic write transfer process without wait states. The process follows a sequence of clock cycles (T0 to T4), where signals are asserted or deasserted at the rising edge of the clock.

* **At T0**: The process is idle, waiting for the next clock cycle.
* **T1**: The address (PADDR) and write data (PRDATA) become valid, and the write signal (PWRITE) and select signal (PSEL) are asserted.
* **T2-T3**: The PENABLE signal is asserted, indicating the Access phase is in progress, during which the address and data remain valid.
* **T4**: The PENABLE signal is deasserted, marking the end of the transfer. If another transfer is not immediately following, PSELX goes LOW.

The transfer completes at the end of this cycle, with all signals remaining valid throughout the Access phase.

### 2.1.2 With wait states



This figure shows a write transfer that includes wait states, extending the process to T6.  
The PREADY signal can extend the transfer by driving PREADY LOW when PENABLE is HIGH.

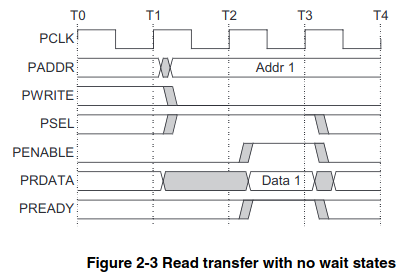
* **T2-T4**: The PENABLE signal is asserted, but the PREADY signal goes LOW, indicating the wait states.

|  |
| --- |
| *Note: It is recommended that the address and write signals remain stable until another access occurs to reduce power consumption.* |

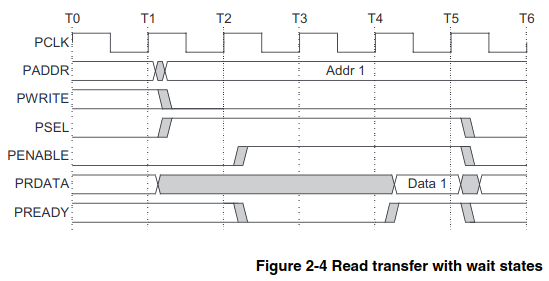
## 2.2 Read transfers(same as write)

The timing of the signals is similar to the write transfer(as described in Write transfers on topic 2.1) but is oriented towards reading data from a slave device.

### 2.2.1 With no wait states



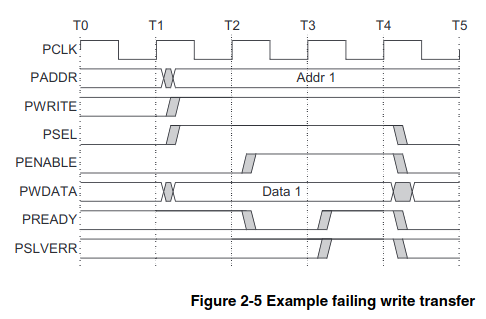
### 2.2.2 With wait states



## 2.3 Error response

* **PSLVERR**: It's a signal used to indicate an error during an APB transfer.
* **Conditions**: Error conditions can occur in both read and write transactions.
* **Validity**: PSLVERR is valid during the last cycle of an APB transfer when PSEL, PENABLE, and PREADY are all HIGH.
* **Recommendation**: It's suggested to drive PSLVERR LOW when not being sampled, i.e., when any of PSEL, PENABLE, or PREADY are LOW.
* **Peripheral behavior**: Whether an error changes the state of the peripheral is peripheral-specific and either behavior is acceptable.
* **Read and write transactions**: In case of an error, write transactions may not change the register within the peripheral, whereas read transactions should return all 0s for a read error.

### 2.3.1 Failing write transfer



The high PSLVERR indicates that there has been an error during the transfer.

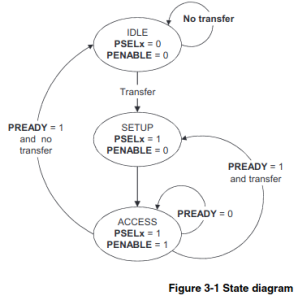
### 2.3.2 Failing read transfer(same as above)

Similar to write transfers, a read transfer can also end with an error.

### 2.3.3 Mapping of PSLVERR(note)

* Bridging AXI to APB: An APB error is mapped back to RESP/BRESP = SLVERR. This is done by mapping PSLVERR to the AXI signals RRESP[1] for reads and BRESP[1] for writes.
* Bridging AHB to APB: PSLVERR is mapped back to HRESP = ERROR for both reads and writes, achieved by mapping PSLVERR to the AHB signal HRESP[0].

## 3.1 Operating States



This diagram visually represents the transitions between the different states of the APB state machine. It shows:

* **IDLE**: The default state of the APB when no transfers are occurring.
  + **To SETUP**: When a transfer is required, the bus moves into the SETUP state.
* **SETUP**: The peripheral select signal PSELx is asserted, indicating which peripheral is being communicated with.
  + **To ACCESS**: The bus only remains in this state for one clock cycle and moves to the ACCESS state on the next clock's rising edge.
* **ACCESS**: The enable signal PENABLE is asserted in the ACCESS state. Stable signals(address, write, select, and write data) are crucial during the transition from SETUP to ACCESS. The exit from the ACCESS state is dependent on the PREADY signal from the peripheral (slave device).
  + **Remaining in ACCESS**: If PREADY is low, the bus remains in the ACCESS state.
  + **Exit from ACCESS**: If high, the bus either returns to the IDLE state if no more transfers are needed ***OR*** moves directly to the SETUP state if another transfer is pending.