Data sheet for ABMA-APB Interface

Functional Description

The APB bus is part of the *Advanced Microcontroller Bus Architecture* (AMBA) hierarchy of buses and is optimized for minimal power consumption and reduced interface complexity. The APB bus is used to interface to any peripheral device which are low bandwidth and do not require the high performance of a pipelined bus interface.

The APB slave interface acts as a bridge between the APB bus and the peripheral device to which the bus is connected. It receives the APB bus signals and converts them to a form in which is understood by the connected peripheral device.

Most common applications of the APB interface is to read and write registers of the connected device. The Peripheral devices connected to the APB bus could be UART, Timer, Keypad, etc.

Features:

- Compliant with AMBA [Rev 2.0] for easy integration with SOC implementations
- Supports APB bus for a wide range of Frequency (approx 100 MHz)
- Programmable Address and data widths
- Easy integration to any SOC implementation

Typical AMBA-APB System:

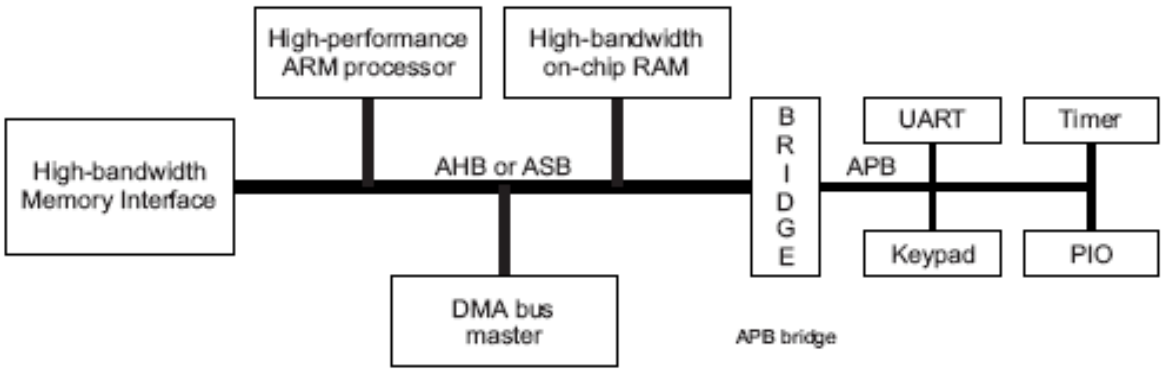


Figure 1: APB in a typical AMBA system

Description:

The Implementation of the APB slave interface is such that it interfaces with the *APB bus* at one end and a *ram* at the other end. It receives the control signals the APB bus and uses them to generated the read and write enable signals for the *ram.* The write data and address signals of the APB bus are forwarded by the APB IF to the ram. Also the data read from the ram is transmitted onto the APB bus.

APB Bus Description:

The APB bus specification can be understood clearly by referring to the state diagram in Figure 2.

Operation of the state machine is through three states described below:

IDLE: This is the default state for the peripheral bus

SETUP: When a transfer is required, the bus moves to this state when the appropriate select signal *PSELx* is asserted. The bus only remains in this state for one clock cycle and will always move to the ENABLE state on the next positive clock edge.

ENABLE: In this state, the enable signal *PENABLE* is asserted. The address, write and select signals should be stable during the transition from **SETUP** to **ENABLE** state.

The **ENABLE** state also lasts for only one clock cycle and the bus will return to **IDLE** state of no further transfers are required. If another transfer is required, the bus will move to **SETUP** state.

APB Slave Description:

The APB slave is very flexible.

Write Transfer : During the write transfer, the *PWRITE* and *PSEL* signal from the APB bus should be asserted. Also the *PADDR* and *PWDATA* buses should be stable before *PENABLE* is asserted. The APB interface generates a *wr_en* signal at the instant when *PSEL, PENABLE* and *PWRITE* are asserted.

The timing diagram is shown in

Figure 3

Read Transfer: During the read transfer, the *PSEL* signal should be asserted and the *PWRITE* signal on the APB bus should be de-asserted. Also the *PADDR* bus should be stable before *PENABLE* is asserted. The APB Interface generates a *rd_en* signal at the instant when *PSEL* and *PENABLE* are asserted and *PWRITE* is de-asserted. The ram will output the data to APBIF which in turn forwards it as *PRDATA* to the APB bus. This *PRDATA* can be sampled by the bus master on the rising edge of the clock at the end of the ENABLE cycle (when *PENABLE* goes low).

The timing diagram is shown in Figure 4

State diagram for APB Transfer:

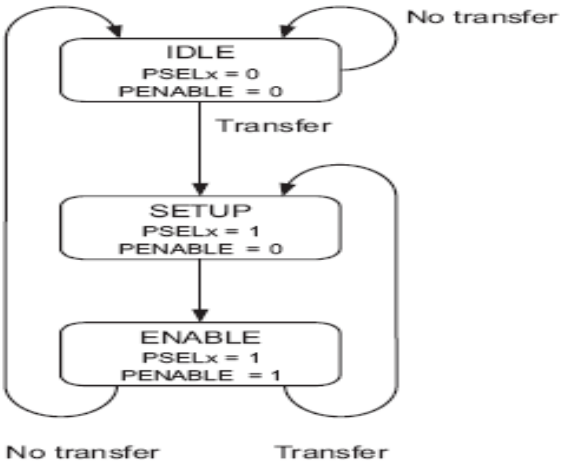


Figure 2: State diagram for APB transfer

Timing Diagrams:

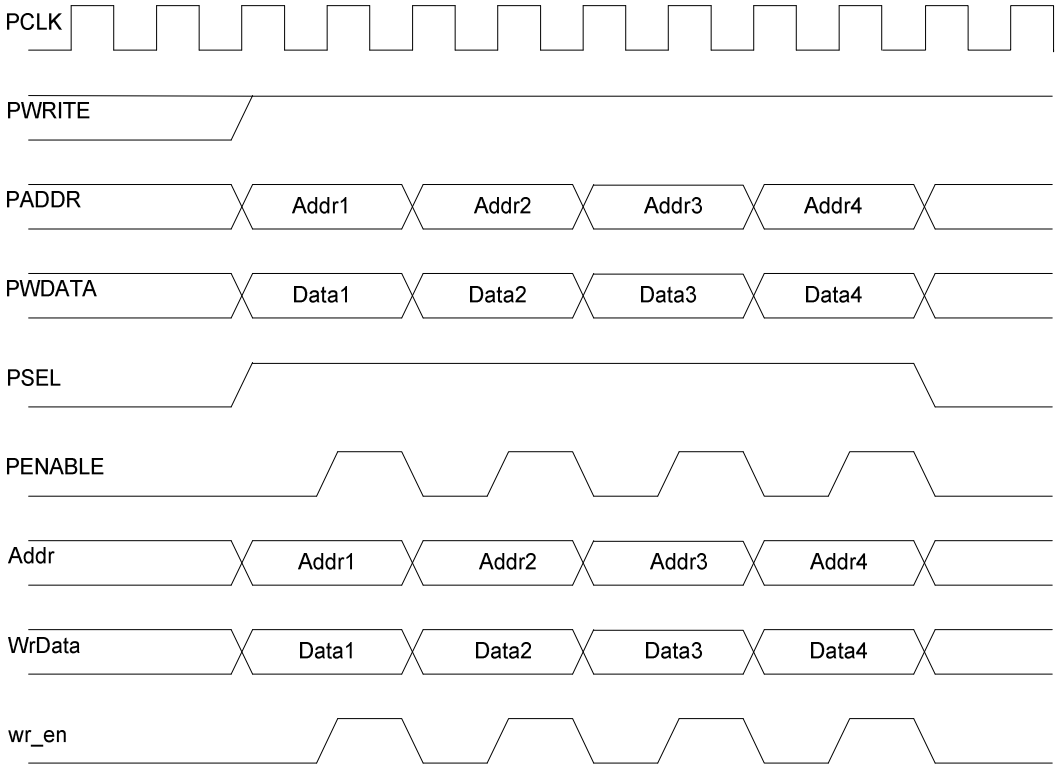
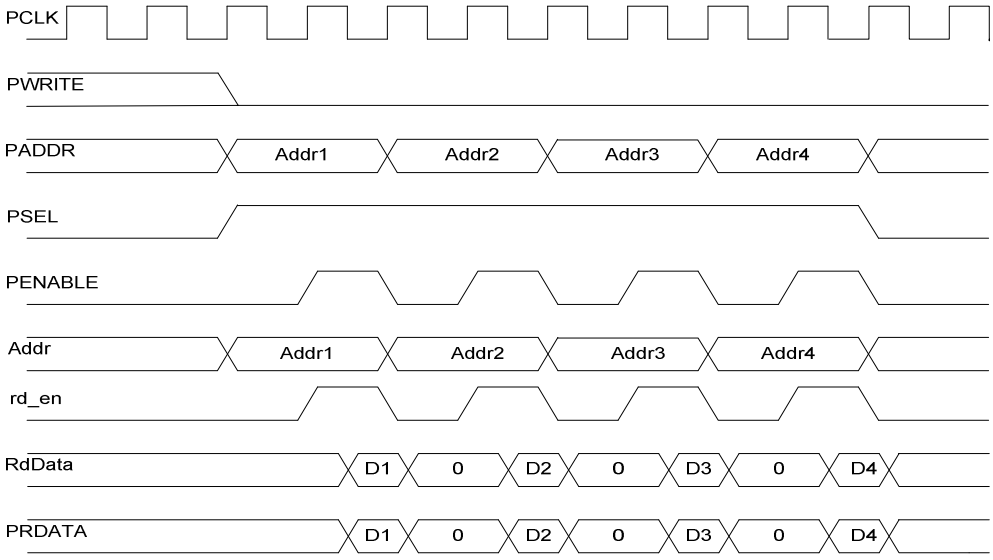


Figure 3: Timing Diagram for Write Transfer



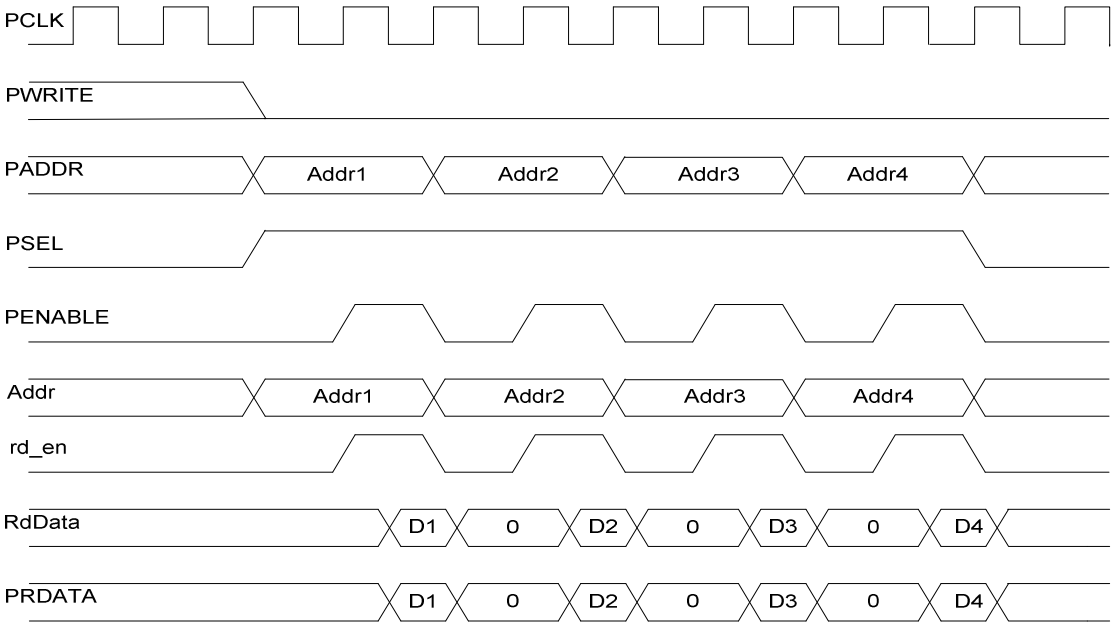
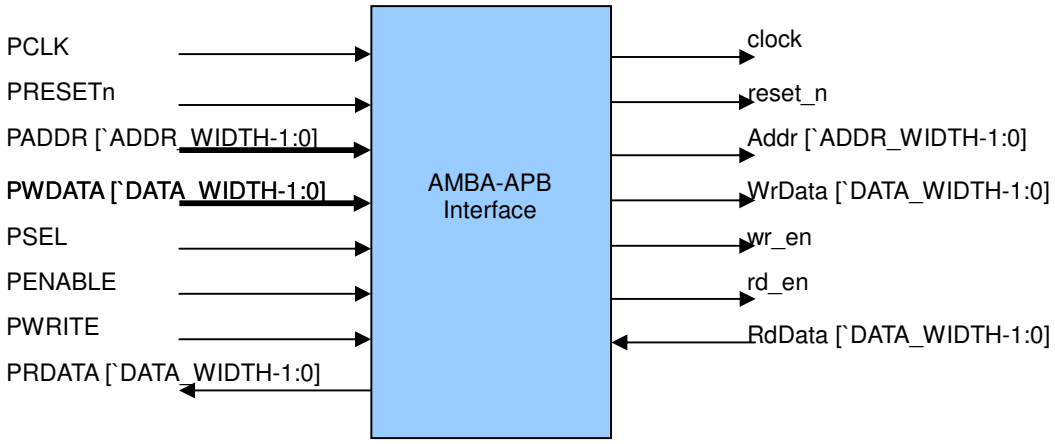


Figure 4: Timing Diagram for Read Transfer

Schematic Symbol



Signal definition table:

Signal	***Direction***	***Description***
PADDR	IN	Input address lines from the APB bus
PWDATA	IN	Input data bus from APB bus
PCLK	IN	Input APB processor clock . This is used as the reference clock for the peripheral device connected to the APB bus
PRESETn	IN	Active low reset for the system. Input from APB end

Signal	*Direction*	*Description*
PWRITE	IN	1'b0 : Read transfer 1'b1 : Write transfer
PSEL	IN	Input from APB. Acts as enable for selecting that particular device (similar to chip select)
PENABLE	IN	Input Enable data line.
PRDATA	OUT	Output data bus to the APB bus.
wr_en	OUT	Enable signal for writing data into the ram connected to APB IF
Addr	OUT	Addr of the ram to which data has tp be written or read from.
WrData	OUT	Data to be written to the connected ram
rd_en	OUT	Enable signal for reading data from the ram connected to APB IF
RdData	IN	Data read from the connected ram
clock	OUT	Clock input the connected ram (same as PCLK)
reset_n	OUT	Active low reset signal (same is PRESETn)

Verification:

The APBIF module has been verified with following approaches:

- All the locations written and read back
- Self checking test benches used
- Prototyped on ML 401 board using LEDs to depict pass condition

Deliverables:

- Verilog RTL source code
- Test benches
- Behavioral model for APB driver