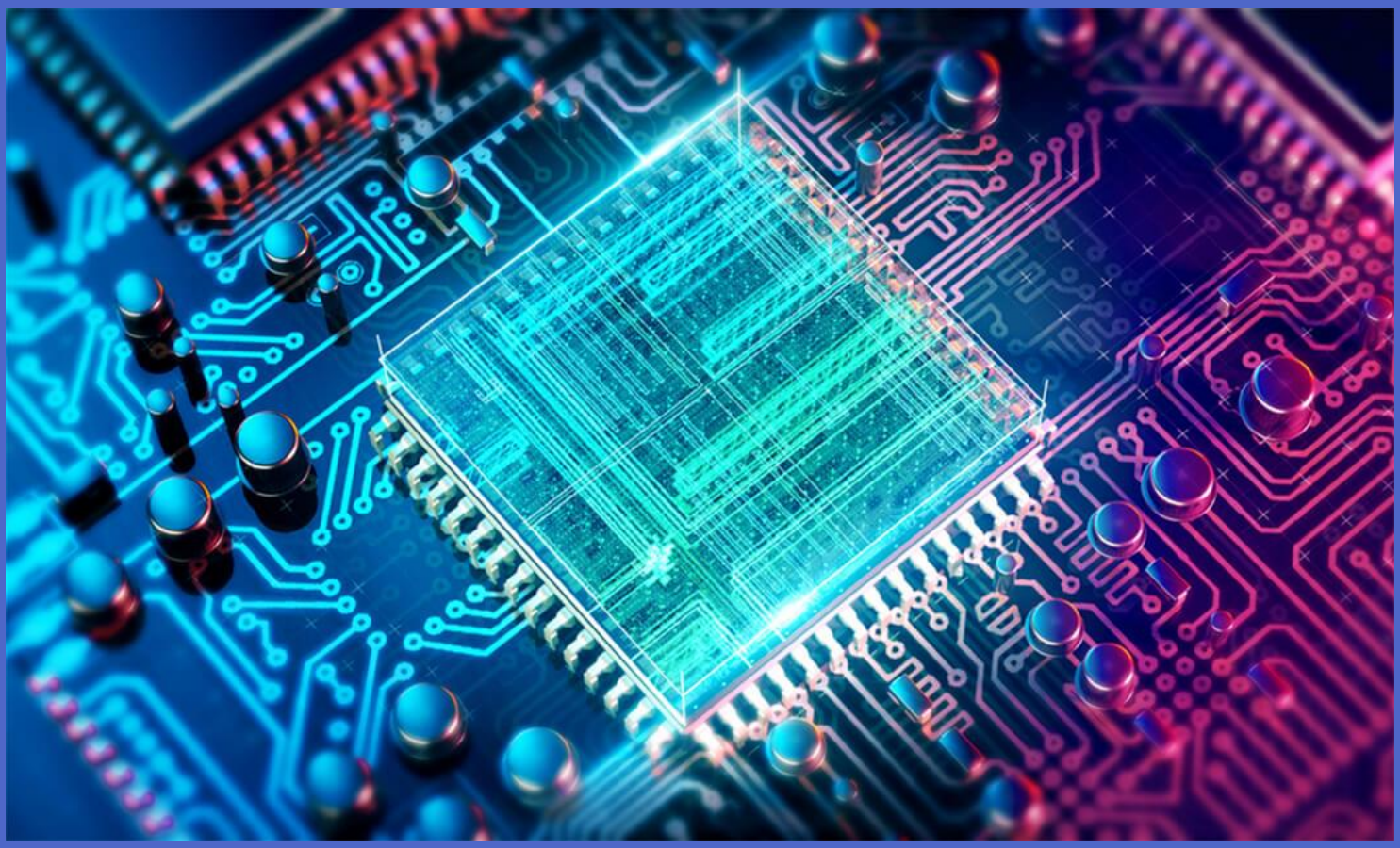
Digital Design and Verification of AMBA3&4 APB Protocol



Abdelrahman Mohamed

Abdelrahmansalby23@gmail.com

LinkedIn

GitHub

Project Link: LINK

CONTENTS

INTRODUCTION

The Advanced Microcontroller Bus Architecture (AMBA) is a widely used on-chip interconnect specification developed by ARM. The AMBA APB (Advanced Peripheral Bus) Protocol is a simple, low-cost interface that is commonly used to connect low-bandwidth peripherals to the high-performance main system bus. The AMBA APB Protocol Version 2.0 is the latest version of this protocol, introducing several enhancements and improvements over the previous version.

APB revisions:

1. AMBA 2 APB Specification
2. AMBA 3 APB Protocol Specification v1.0

   The AMBA 3 APB Protocol Specification v1.0 defines the following additional functionality:

   - Wait states
     - **PREADY**: A ready signal to indicate completion of an APB transfer.
   - Error reporting
     - PSLVERR An error signal to indicate the failure of a transfer.

   This version of the specification is referred to as APB3, which is supported in my design.
3. AMBA APB Protocol Specification v2.0

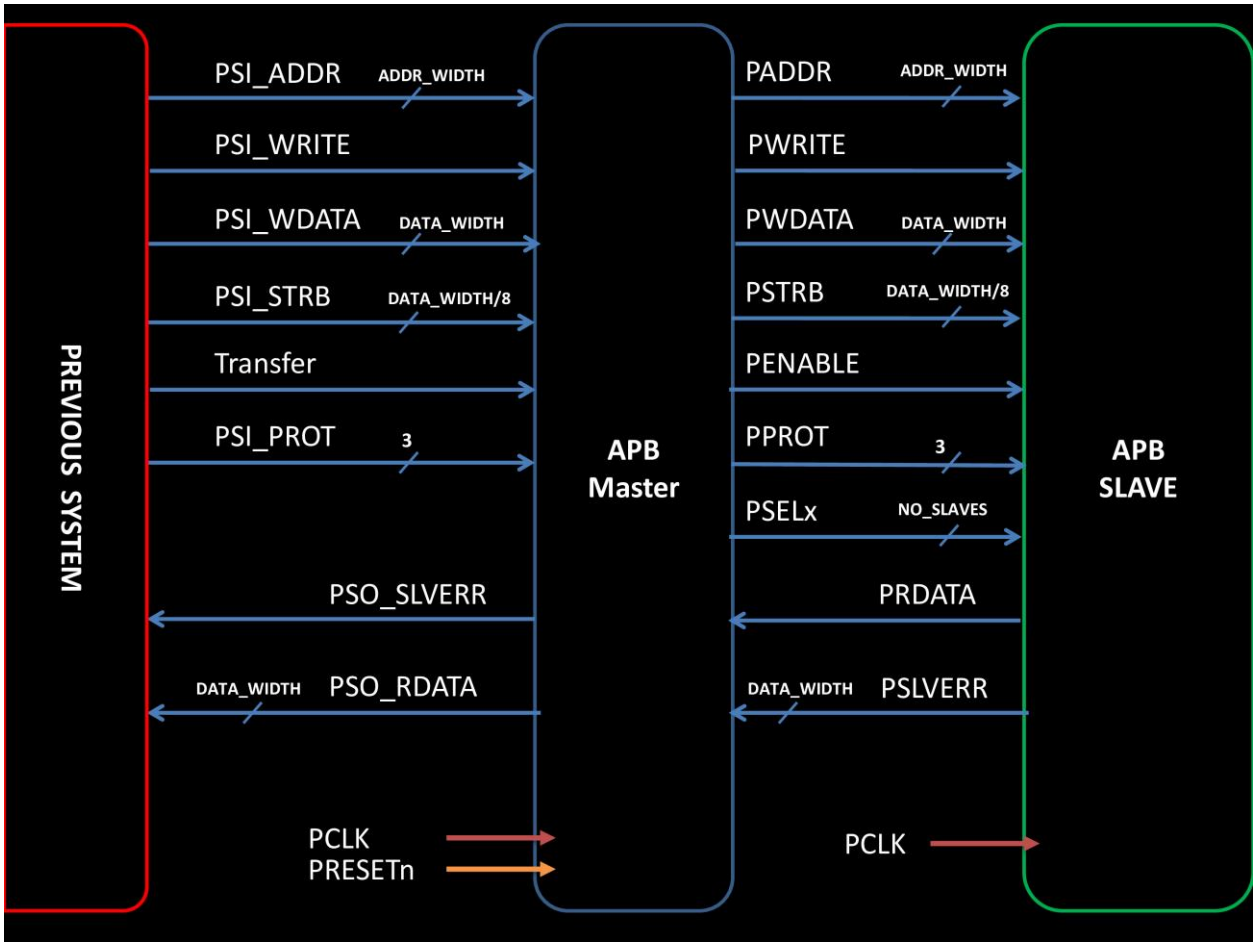
   The AMBA APB Protocol Specification v2.0 defines the following additional functionality:

   • Transaction protection. See Protection unit support on page 3-8.
     - **PPROT** A protection signal to support both non-secure and secure transactions on APB.

   • Sparse data transfer. See Write strobes on page 3-4. The following interface
     - **PSTRB:** A write strobe signal to enable sparse data transfer on the write data bus.

   This version of the specification is referred to as APB4, which is supported in my design.

DESIGN ARCHITECTURE



KEY FEATURES

The key features of the AMBA APB Protocol Version 2.0 include:

- Synchronous operation
- Simple, low-cost interface
- Non-pipelined bus transactions
- Support for 8-bit, 16-bit, and 32-bit data transfers
- Efficient power management through clock gating

SIGNAL DESCRIPTION

Input Signal

signal	Source	Description
Global Inputs		
PCLK	Clock source	Clock. The rising edge of PCLK 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.
Inputs from Previous system to APB Master		
PSI_ADDR	Previous System	Previous System Address bus for read and write, up to 32 bits
PSI_WRITE	Previous System	Direction. This signal indicates an APB write access when HIGH and an APB read access when LOW.
PSI_WDATA	Previous System	Write data. Driven by the Previous System bus when PWRITE is HIGH. This bus can be up to 32 bits wide.
PSI_STRB	Previous System	Write strobes. This signal indicates which byte lanes to update during a write transfer. There is one write strobe for each eight bits of the write data bus. Therefore, PSTRB[n] corresponds to PWDATA[(8n + 7):(8n)]. Write strobes must not be active during a read transfer.
Transfer	Previous System	This signal indicate to Transfer from IDLE State to SETUP State
PSI_PROT	Previous System	Protection type. This signal indicates the normal, privileged, or secure protection level of the transaction.
Inputs from Slave to APB Master		
PRDATA	Slave	Read Data. The selected slave drives this bus when PWRITE is LOW. up to 32-bits wide.
PSLVERR	Slave	This signal indicates a transfer failure. APB peripherals are not required to support the PSLVERR pin. This is true for both existing and new APB peripheral designs.

SIGNAL DESCRIPTION

Output Signal

signal	Source	Description
APB Master Outputs to Slave		
PADDR	Master	Address. This is the APB Master address bus, up to 32 bits
PWRITE	Master	Direction. This signal indicates an APB write access when HIGH and an APB read access when LOW.
PWDATA	Master	Write data. This bus is driven by the peripheral bus Master when PWRITE is HIGH. up to 32 bits wide.
PSTRB	Master	Write strobes. This signal indicates which byte lanes to update during a write transfer. There is one write strobe for each eight bits of the write data bus. Therefore, PSTRB[n] corresponds to PWDATA[(8n + 7):(8n)]. Write strobes must not be active during a read transfer.
PSELx	Master	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 PSELx signal for each slave.
PPROT	Master	Protection type. This signal indicates the normal, privileged, or secure protection level of the transaction.
APB Master Outputs to Previous system		
PSO_RDATA	Master	Read Data. Master drives this bus to Previous System when PWRITE is LOW. up to 32-bits wide.
PSO_SLVERR	Master	This signal indicates a transfer failure. APB peripherals are not required to support the PSLVERR pin. This is true for both existing and new APB peripheral designs.

AMBA3 & AMBA4 FEATURES

1. AMBA3

It's also called Issues A and B

a. APB signal **PREADY** added

b. APB signal **PSLVERR** added

2. AMBA4

c. APB signal **PPROT** added

PPROT[2:0]	Protection level
[0]	1 = privileged access 0 = normal access
[1]	1 = nonsecure access 0 = secure access
[2]	1 = instruction access 0 = data access

d. APB signal **PSTRB** added

ADDRESS DECODING

This design have 4 slaves which need only two bits to decoding

PSELx is 4 bits

Slave 1 0001

Slave 2 0010

Slave 3 0100

Slave 4 1000

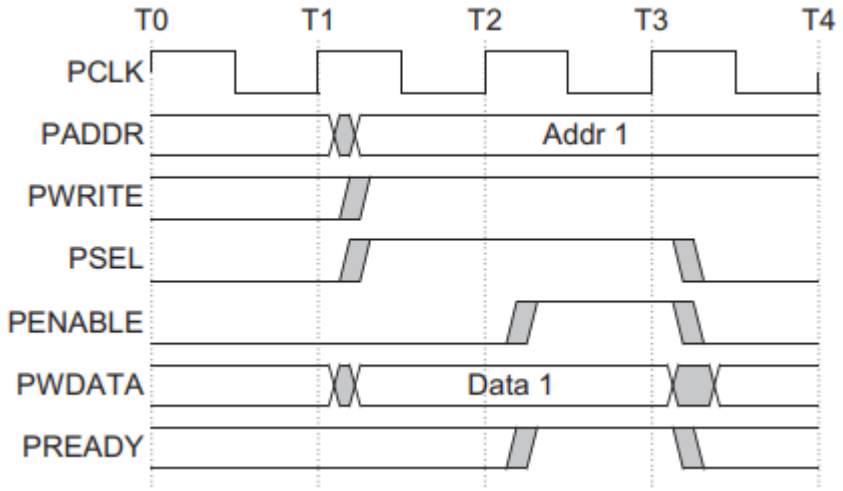
It's decoding using the last tow bits of PADDR ➔ PADDR[31:30];

```
// ADDRESS Decoding
    always @(posedge PCLK or negedge PRESETn) begin
        if (!PRESETn) begin
            PSELx <= 'b0;
        end
        else if (NextState == IDLE) begin
            PSELx <= 'b0;
        end
        else begin
            case (PSI_ADDR[31:30])
                'b00: PSELx <= 4'b0001;
                'b01: PSELx <= 4'b0010;
                'b10: PSELx <= 4'b0100;
                'b11: PSELx <= 4'b1000;
                default: begin
                   PSELx <= 'b0;
                end
            endcase
        end
    end
```

TRANSACTIONS

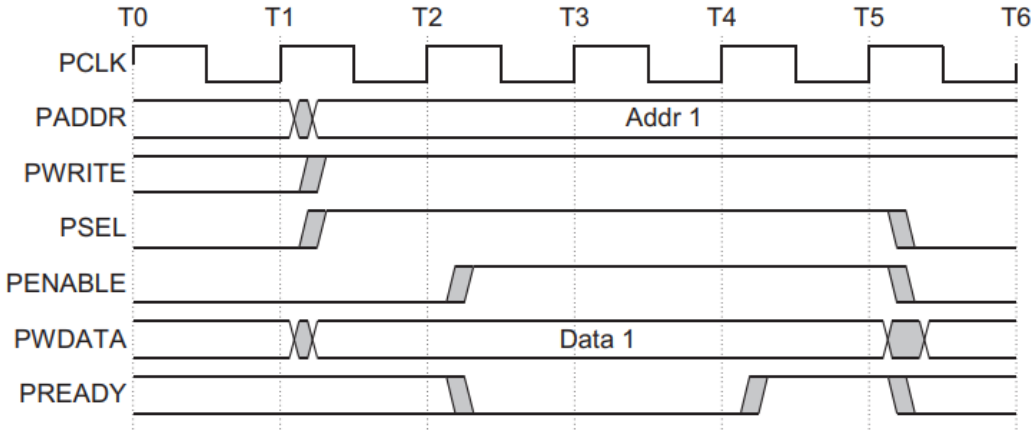
Write Transfer

1. With no wait state



I. At T1, address, data, write, and select signals are registered at PCLK rising edge.
II. At T2, enable and ready signals are registered at PCLK rising, indicating Access phase.
III. Address, data, and control signals remain valid until transfer completes at T3.
IV. Enable signal is deasserted at transfer end, select signal deasserted unless next transfer.
V. This defines the Setup, Access, and completion phases of a standard AMBA write transfer.

2. With wait state

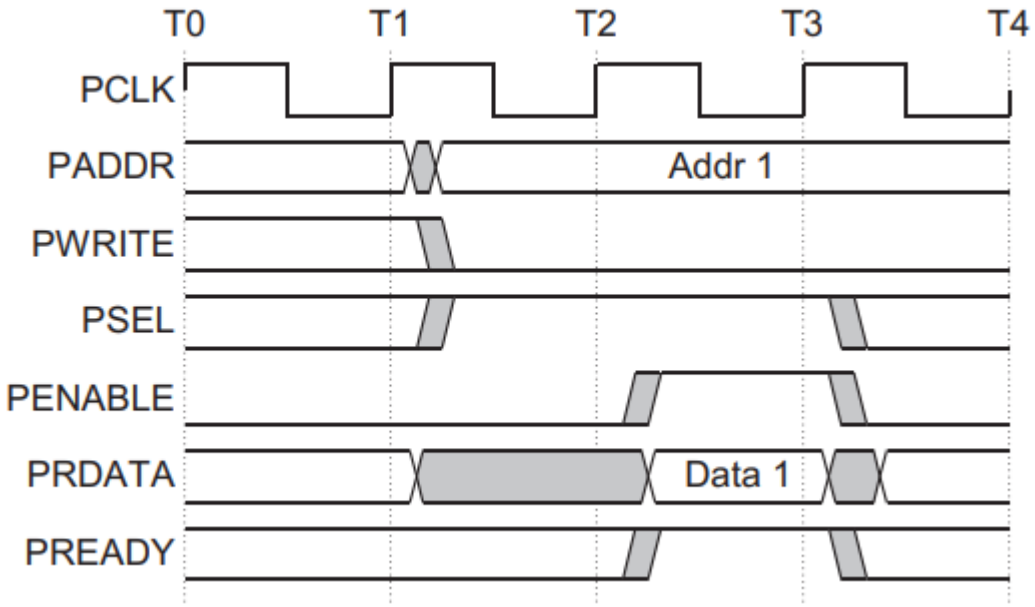


Only transfer done when PREADY is HIGH

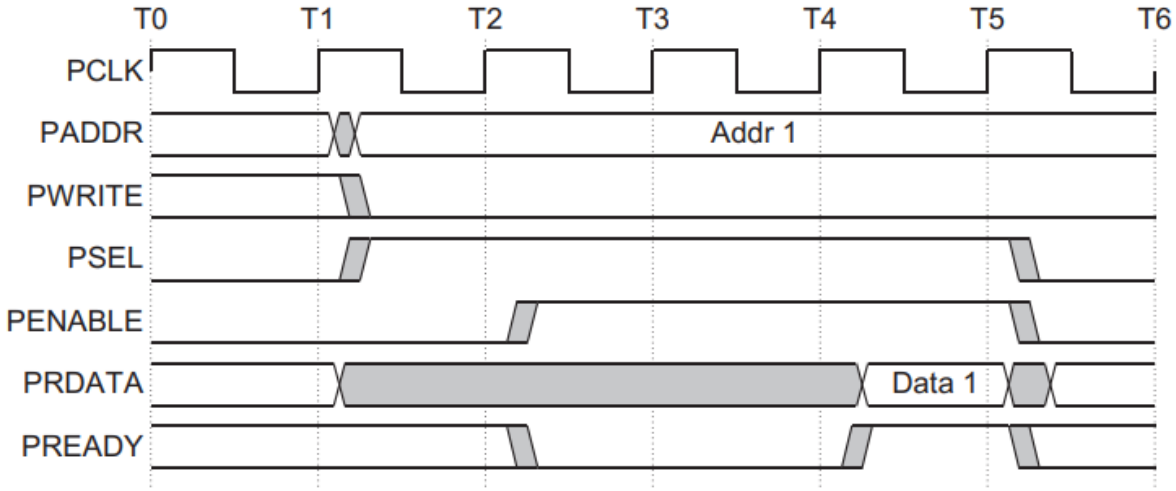
TRANSACTIONS

Read Transfer

a. With no wait state



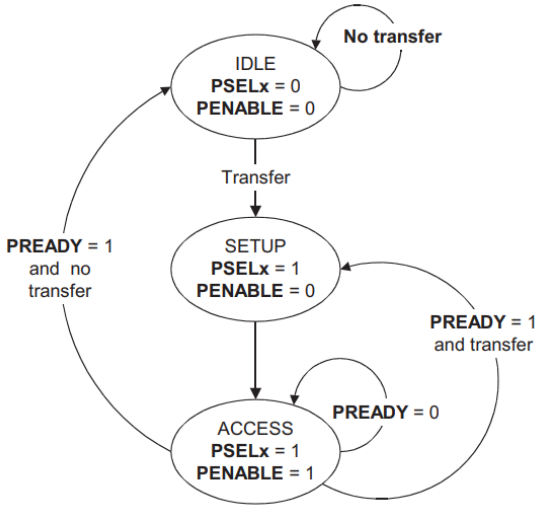
b. With wait state



Only transfer done when PREADY is HIGH

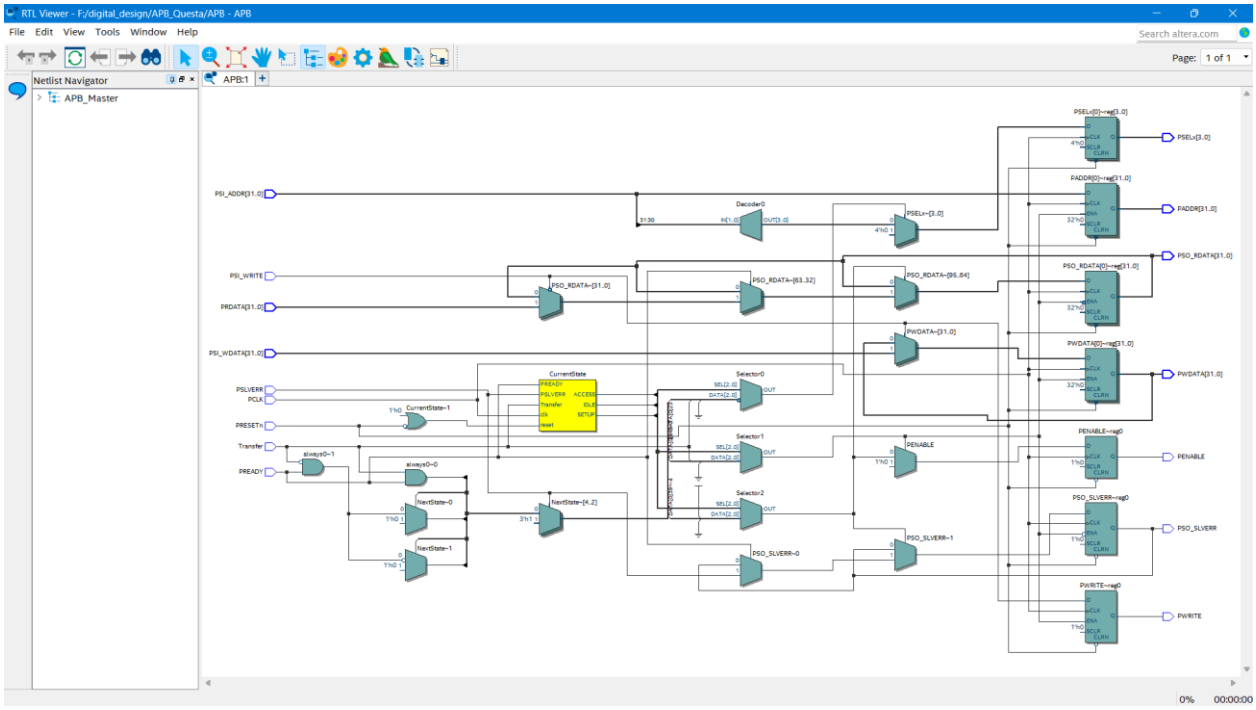
OPERATING STATES

Operating states in ARM Specs pdf

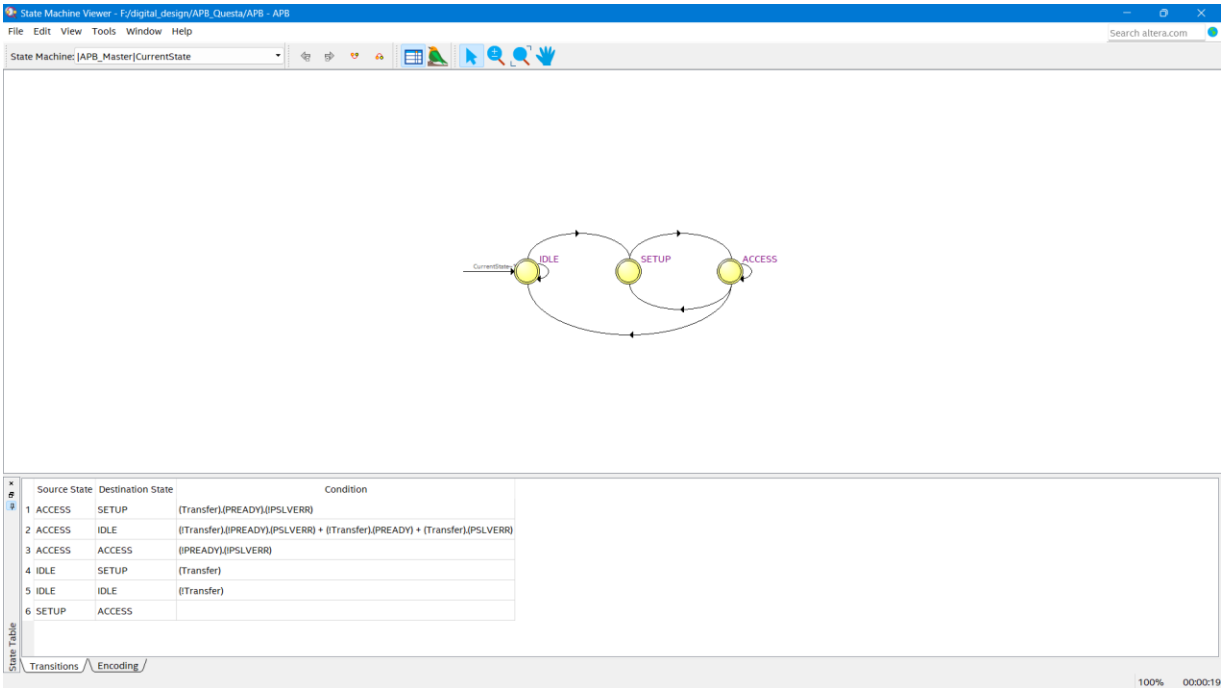


IDLE	This is the default state of the APB.
SETUP	When a transfer is required the bus moves into the SETUP state, where the appropriate select signal, PSELx, is asserted. The bus only remains in the SETUP state for one clock cycle and always moves to the ACCESS state on the next rising edge of the clock
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

RTL ELABORATION



FSM in Questa



QUESTA TIMING ANALYSIS

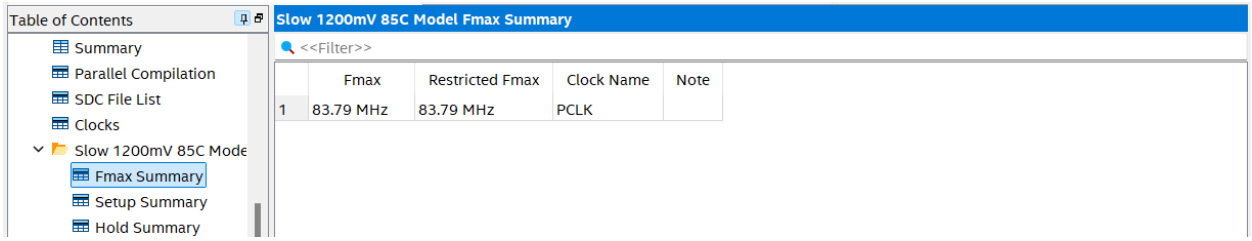
PCLK Period

Clocks

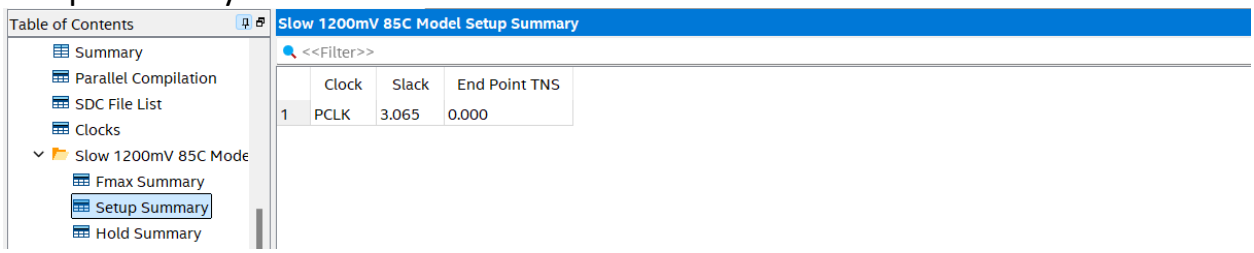
<<Filter>>

	Clock Name	Type	Period	Frequency	Rise	Fall	Duty Cycle	Divide by	Multiply by	Phase	Offset	Edg
1	PCLK	Base	15.000	66.67 MHz	0.000	7.500						

Fmax



setup summary



hold summary

Clocks

<<Filter>>

	Clock Name	Type	Period	Frequency	Rise	Fall	Duty Cycle	Divide by	Multiply by	Phase	Offset	Edg
1	PCLK	Base	15.000	66.67 MHz	0.000	7.500						

VERIFICATION PLANE

1. Asert RESET to verfy the output
2. Write with no wait
    - Assert PSLEVRR with less freq. to verfy the output
3. Write with wait
    - Make waiting period variable to test all cases
    - Assert PSLEVRR with less freq. to verfy the output
4. Read with no wait
    - Check the PSTRB and ensures it’s equal to zero
    - Assert PSLEVRR with less freq. to verfy the output
5. Read with wait
    - Check the PSTRB and ensures it’s equal to zero
    - Assert PSLEVRR with less freq. to verfy the output

VERIFICATION CONSTRAINTS

1. Constraint on **PRESETn** to be active (**RESET_ACTIVATE**%) of the time, which **RESET_ACTIVATE** is a parameter in ***shared_pkg.***

2. Constraint on **PSLVERR** to be active (**PSLVERR_ACTIVATE**%) of the time, which **PSLVERR_ACTIVATE** is a parameter in ***shared_pkg.***

3. Constraint on **PSI_WDATA** to be (10% is MAX, 5% is ZERO and 85% other values), which MAX is parameter in ***shared_pkg*** take maximum value of **PWDATA** and ZEO is parameter take value zero.

4. Constraint on **PRDATA** to be (10% is MAX, 5% is ZERO and 85% other values), which MAX is parameter in ***shared_pkg*** take maximum value of **PSI_ADDR** and ZEO is parameter take value zero

5. Constraint on **PSI_ADDR[31:30]** to take all values with equal chances.

6. Constraint on **WaitPeriod** to be between **“MIN_WAIT_PEROPD”** and **“MAX_WAIT_PEROPD”,** which **MIN_WAIT_PEROPD and MAX_WAIT_PEROPD** is parameter in ***shared_pkg.***

For AMBA4:

Constraint on **PSI_STRB** to be zero when **PSI_WRITE** is zero.

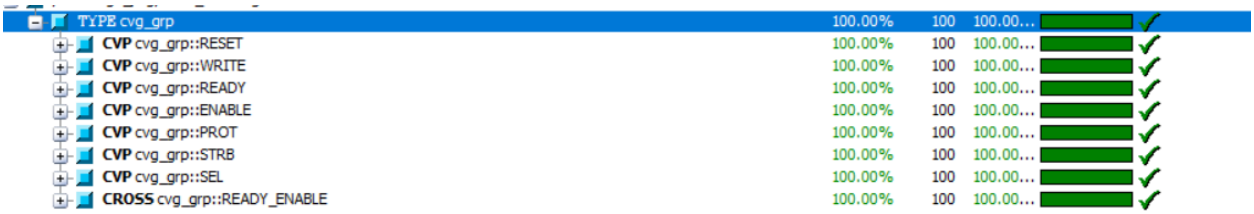
V E R I F I C A T I O N
C O V E R A G E

Bins for the following signals:

- **PRESETn**
- **PWRITE**
- **PREADY**
- **PENABLE**
- **PSELx**

Cros coverage between PREADY and PENABLE.

For AMBA 4 Bins for PPROT



VERIFICATION

ASSERTION

Assertion for the following:

1. When PRESETn is Active
   a. AMBA 3

```
RST_PENABLE     : assert to ensures that PENABLE     is zero When PRESETn is Active
RST_PADDR       : assert to ensures that PADDR       is zero When PRESETn is Active
RST_PWRITE      : assert to ensures that PWRITE      is zero When PRESETn is Active
RST_PSO_RDATA   : assert to ensures that PSO_RDATA   is zero When PRESETn is Active
RST_PSO_SLVERR  : assert to ensures that PSO_SLVERR  is zero When PRESETn is Active
RST_PWDATA      : assert to ensures that PWDATA      is zero When PRESETn is Active
RST_PSELx       : assert to ensures that PSELx       is zero When PRESETn is Active
```

   b. AMBA 4

```
RESET_STRB  : assert to ensures that PSTRB  is zero When PRESETn is Active
RESET_PROT  : assert to ensures that PROT   is zero When PRESETn is Active
```

2. Other assertion
   a. AMBA 3

```
NextState_pslv : assert to ensure when PSLVERR is active and past NextState is ACCESS then NextState is IDLE
selx           : assert to ensure when PSI_ADDR!=0 then one and only one of the bits in PSELx is asserted
ASSERT_ENABLE  : assert to ensure when NextState is ACCESS then PENABLE should be asserted
ADDRESS        : assert to ensure when NextState is SETUP then PADDR = PSI_ADDR
WRITE          : assert to ensure when NextState is SETUP then PWRITE = PSI_WRITE
WRITE_DATA     : assert to ensure when NextState is SETUP and PWRITE HIGH then PWDATA = PSI_WDATA
READ_DATA      : assert to ensure when NextState is ACCESS and PWRITE LOW then PSO_RDATA = PRDATA
SLVERR         : assert to ensure when PREADY is HIGH then PSO_SLVERR = PSLVERR
```

   b. AMBA 4

```
READ_STRB : assert to ensure when PWRITE is LOW PSTRB = 0
STRB      : assert to ensure when NextState is SETUP and PWRITE is HIGH then PSTRB = PSI_STRB
PROT      : assert to ensure when NextState is ACCESS then PPROT is PSI_PROT
```

REFERENCES

AMBA® APB
Protocol: https://www.eecs.umich.edu/courses/eecs373/readings/IHI0024C_amba_apb_protocol_spec.pdf