WIT – NXP BATCH-III

SYNOPSIS SUBMISSION

BATCH NAME	NXP WIT BATCH-III(GROUP – 8)
NAME OF STUDENTS	1. Radhika Garg 2. Ridhima 3. Ridhima Choudhary 4. Riya Singla 5. Sachu Meghana 6. Sai Tanmayee P 7. Sakshi Saxena
TITLE OF THE PROJECT/SYNOPSIS	RTL Design of APB Slave Module in Verilog-HDL
GUIDE NAME	Mr.Shashikant Sharma, Mr.Sanjay Saini
SUBMISSION DATE	22ND SEPTEMBER 2025

Abstract / Synopsis:

PROBLEM STATEMENT

Efficient communication between a processor and peripherals is necessary to achieve high performance in SoC designs. However, peripheral devices often operate with lower complexity and require a lightweight communication protocol. Traditional bus systems introduce overhead in terms of latency, complexity, or power consumption.

Thus, the need arises to design and verify an APB slave module that:

- Implements the APB protocol faithfully.
- Provides reliable register-level read/write transactions.
- Can be easily integrated into SoC peripheral subsystems.

AIM

To design and implement a Register Transfer Level (RTL) model of an APB Slave Module using Verilog-HDL, ensuring strict compliance with the AMBA APB protocol, and to validate its correctness, efficiency, and scalability through simulation and verification.

OBJECTIVES

1. To study the APB protocol signals and timing as per ARM AMBA specifications.
2. To design a synthesizable APB slave RTL module in Verilog HDL.

3. To implement read/write functionality with memory-mapped registers.
4. To verify the correctness of the design using testbenches and simulations.

To provide a modular and reusable RTL block that can be extended for multiple peripherals.

SCOPE OF WORK

Implementation of a basic APB slave with memory-mapped registers.
Simulation only (no FPGA synthesis since EDA Playground is simulator-based).
Testbenches for:

- Single read/write operation.
- Continuous transfers.
- Invalid/edge cases

Waveform verification to confirm protocol compliance.

APB PROTOCOL SIGNALS

The APB protocol defines the following key signals:

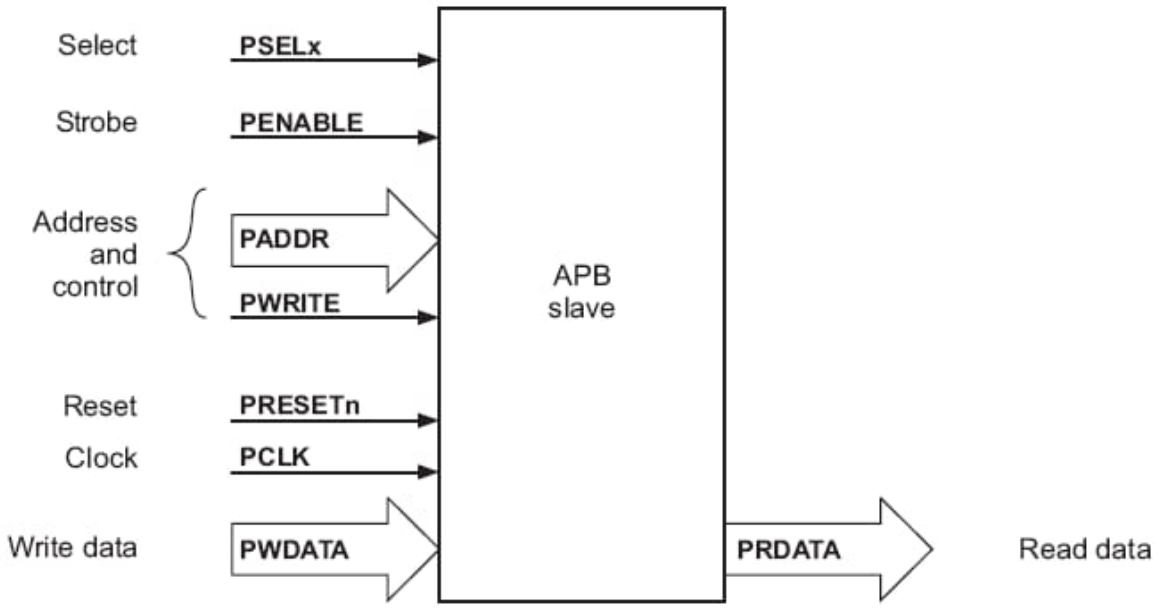
- **PCLK:** Clock signal.
- **PRESETn:** Reset (active low).
- **PADDR:** Address bus.
- **PWDATA:** Write data bus.
- **PRDATA:** Read data bus.
- **PWRITE:** Indicates read (0) or write (1).
- **PSEL:** Slave select signal.
- **PENABLE:** Indicates second phase of transaction.
- **PREADY:** Slave response indicating completion.
- **PSLVERR:** Error response (optional).

METHODOLOGY

1. **Design Phase (Verilog RTL on EDA Playground):**
    - Implement APB slave using **FSM** (IDLE → SETUP → ACCESS).
    - Add a register file (e.g., 4 × 8-bit registers).

2. **Testbench Development:**
    - Write a Verilog testbench that acts as an **APB master**.
    - Generate read/write transactions by driving APB signals.
3. **Simulation & Verification:**
    - Use **EDA Playground simulator (Icarus Verilog / Verilator / Aldec RivieraPro / ModelSim)**.
    - View and analyze **waveforms** in the integrated viewer (EPWave).
4. **Documentation:**
    - Capture **waveform screenshots** for read/write operations.

BLOCK DIAGRAM



THEORY

The Advanced Peripheral Bus (APB) is a fundamental part of the AMBA (Advanced Microcontroller Bus Architecture) protocol, widely adopted in System-on-Chip (SoC) designs. Unlike its high-performance counterparts (AXI, AHB), APB is specifically optimized for low-power, low-complexity, and latency-insensitive peripherals such as GPIOs, UARTs, timers, and control/status registers.

The APB protocol follows a two-phase operation:

1. Setup Phase → Master asserts PSEL, provides address (PADDR) and control signals.

2. Access Phase → With PENABLE asserted, data transfer occurs either as read (PRDATA) or write (PWDATA). The slave then responds with PREADY to indicate transaction completion or PSLVERR for error signaling.

The APB slave module is the backbone of this communication, responsible for:

- Address Decoding → Identifying target registers.
- Data Handling → Executing accurate read and write transactions.
- Synchronization → Ensuring timing compliance with the system clock (PCLK).
- Response Generation → Providing handshake and error signals.

By designing this slave in Verilog-HDL at the RTL level, the module becomes synthesizable, reusable, and seamlessly integrable into SoC environments. Simulation ensures protocol correctness and timing accuracy, while synthesis validates hardware feasibility. The outcome is a scalable and energy-efficient communication module, suitable for next-generation embedded and SoC designs.

EXPECTED OUTCOMES

- RTL model of APB Slave in Verilog HDL.
- Testbench with simulation results (waveforms of read/write).
- Verified compliance with AMBA APB protocol.
- A reusable, modular design ready for integration in SoC designs.

TOOLS AND TECHNOLOGIES

- **Coding Language:** Verilog HDL
- **Platform:** EDA Playground (https://edaplayground.com/)
- **Simulator:** Icarus Verilog / ModelSim (default on EDA Playground)
- **Waveform Viewer:** EPWave (integrated in EDA Playground)

APPLICATIONS

- **Embedded Systems:** Communication between CPU and peripherals.
- **SoC Design:** Integration of low-power peripherals.
- **Education/Research:** Understanding and practicing bus protocol design.

PROJECT REPOSITORY

The complete Verilog source code, testbench files, and simulation results for the APB Slave Module can be accessed at the following link:

GitHub Repository LINK: https://github.com/Team-No-8/APB-Slave-Design