Design and Synthesis of an Energy-Efficient In-Order Core Using AMBA AHB-Lite and AXI4-Lite Slave Interface

Gagan R(251090420018), B S Manoj Reddy(251090420010)
Dept. of Electronics and Communication Engineering (Microelectronics) Manipal Institute of Technology (MIT)
Manipal, India
gaganrg17@gmail.com, satyamanojreddy@gmail.com

***Abstract*—This paper compares the AHB Lite and AXI Lite bus architectures using Verilog HDL and Cadence Genus Synthesis Tool under identical conditions. The study evaluates timing, power, and area performance of both slave designs. Results show that AHB Lite achieves lower power and smaller area, while AXI Lite provides better data handling through separate read and write channels. Both meet timing closure with zero violations. The findings highlight the balance between AHB Lite efficiency and AXI Lite scalability for system on chip applications.**

Index Terms*—AHB-Lite, AXI-Lite, Verilog HDL, AMBA Bus, SoC Design, Synthesis, Power Analysis, Timing Closure, Cadence Genus.

I. INTRODUCTION

Modern semiconductor design, system on chip technology integrates multiple functional modules such as processors, memory units, and peripheral controllers on a single silicon substrate. Effective communication between these components is essential to achieve high performance, low power consumption, and predictable timing. To standardize this communication, ARM introduced the Advanced Microcontroller Bus Architecture, which provides a family of on chip communication protocols widely adopted in the semiconductor industry.

Among the AMBA protocols, AHB Lite and AXI Lite are two lightweight interfaces commonly used in small and medium scale system on chip designs. Both are intended for connecting peripheral and control modules to the main processor or memory system, offering simple and reliable communication. However, they differ in architectural structure and operational behaviour, which directly influence synthesis results in terms of area, timing, and power.

The AHB Lite protocol is a simplified version of the Advanced High-Performance Bus and supports a single master with multiple slaves. It uses a shared address and data bus to handle transactions in a sequential manner. This design offers predictable timing, low logic complexity, and minimal area overhead, which makes it suitable for low power embedded applications and microcontroller-based systems.

The AXI Lite protocol is a subset of the Advanced Extensible Interface and introduces separate read and write channels to allow concurrent data transfers. Although this structure requires more control logic and interconnect resources, it improves performance by reducing data access latency and supporting better modularity. This makes AXI Lite more appropriate for systems requiring higher throughput and scalable interconnect designs.

The main objective of this study is to perform a synthesis-based comparison of AHB Lite and AXI Lite slave designs under identical conditions. Both modules are implemented using Verilog hardware description language and synthesized using the Cadence Genus tool with the same process, voltage, and temperature settings. The analysis focuses on timing closure, power efficiency, and silicon area utilization to evaluate their suitability for different system level requirements.

The findings from this work help designers choose the appropriate bus protocol based on the target application. AHB Lite provides a compact and energy efficient solution for simple control interfaces, while AXI Lite offers greater flexibility and performance scalability for complex system architectures. This comparison demonstrates how architectural choices in bus design directly impact post synthesis results and overall system optimization.

II. METHODOLOGY

Both AHB Lite and AXI Lite slave interfaces were designed in Verilog HDL and verified using functional simulation before synthesis. The Cadence Genus Synthesis Tool was used under identical PVT conditions to ensure a fair comparison. The methodology included defining design specifications, developing RTL models, performing functional simulation in ModelSim, and synthesizing the designs with consistent timing and voltage constraints. Post-synthesis reports were analyzed to compare timing, area, and power characteristics, highlighting the performance trade-offs between the two AMBA bus architectures.

A. Operation of AXI4-Lite Slave Module

The AXI4-Lite Slave module is a simple and fully synthesizable peripheral interface that allows communication with an AXI4-Lite master through independent read and write channels. It includes two 32-bit internal registers, reg0 and reg1, that can be accessed via specific address offsets. The design follows the AMBA AXI4-Lite protocol and ensures clean timing closure suitable for synthesis tools such as Cadence Genus.

The slave communicates with the master through five independent channels: Write Address, Write Data, Write Response, Read Address, and Read Data as shown in Figure 1. Each channel uses a VALID–READY handshake mechanism to ensure data transfer synchronization between master and slave.

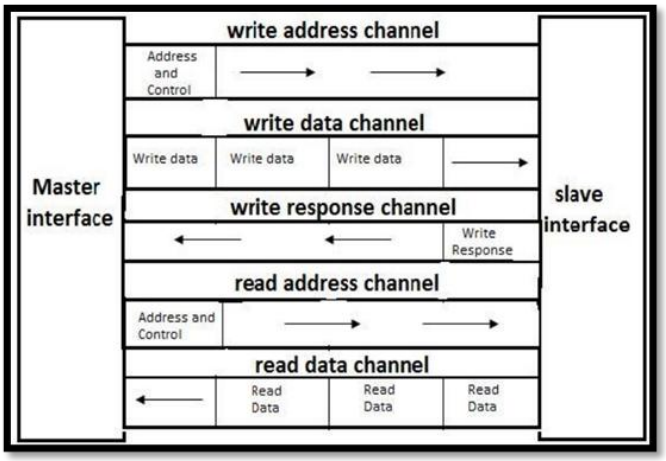

Figure 1: AMBA AXI4-Lite bus protocol

Write Operation Flow:

1. The master initiates a write transaction by asserting AWVALID = 1 with a valid AWADDR. The slave responds by asserting AWREADY = 1 to indicate readiness.
2. The master transmits the data (WDATA) with WVALID = 1. The slave acknowledges by asserting WREADY = 1.
3. When both address and data are valid, the slave decodes AWADDR [3:2] and writes WDATA into the corresponding register (reg0 or reg1).
4. Upon completion, the slave asserts BVALID = 1 and sends BRESP = 2'b00 (OKAY response).
5. The master asserts BREADY = 1, acknowledging the response. The slave then deasserts BVALID, completing the write transaction.

Read Operation Flow:

1. The master starts a read transaction by asserting ARVALID = 1 with ARADDR. The slave asserts ARREADY = 1 when it accepts the address.
2. Based on ARADDR[3:2], the slave retrieves data from reg0 or reg1 and places it on RDATA.
3. The slave asserts RVALID = 1 and sets RRESP = 2'b00 (OKAY response).
4. When the master asserts RREADY = 1, the data transfer is complete. The slave then deasserts RVALID, concluding the read operation.

Table 1: AXI4-lite operation

Channel	Master Signal	Slave Signal	Function Description
Write Address	AWVALID	AWREADY	Address handshake
Write Data	WVALID	WREADY	Data handshake
Write Response	BREADY	BVALID	Write completion acknowledgment
Read Address	ARVALID	ARREADY	Read address handshake
Read Data	RREADY	RVALID	Read data handshake

Advantages AXI4-Lite:

1. The design is fully synthesizable and timing-clean, compatible with Cadence Genus and similar EDA tools.
2. It complies with the AXI4-Lite protocol, ensuring interoperability within AXI-based systems. Independent read and write channels enable parallel data flow and better performance.
3. It is easily extendable to multiple registers or application-specific peripherals and is ideal for FPGA or SoC integration, providing a lightweight interface for control registers.

Applications of AXI4-Lite:

1. This module can be used as a configuration interface for on-chip peripherals such as timers, GPIOs, and ADCs.
2. It is suitable for inclusion in AXI-based SoC bus architectures for lightweight control and status communication.
3. It is also useful for educational purposes to study and verify AMBA AXI protocol behavior. The design can serve as a control interface in hardware accelerators or IP cores integrated into larger systems.
4. Additionally, it can be used as a base model for extension to AXI4-Full or AXI4-Stream protocols in advanced designs.

B. Operation of AHB-Lite Slave Module

The AHB-Lite Slave module implements a simple two-register peripheral interface compatible with the AMBA AHB-Lite protocol. The design supports both read and write transactions and maintains protocol compliance with basic handshake and transfer control signals. Two 32-bit registers (REG0 and REG1) are used for data storage and retrieval based on address decoding. The testbench verifies correct functionality through sequential write and read operations.

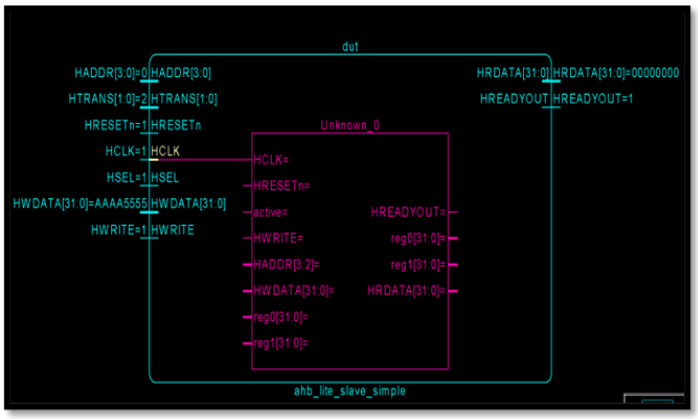

Figure 2: AMBA AHB-Lite bus protocol

Algorithm Steps:

1. Apply reset and initialize registers to zero.
2. Set HREADYOUT = 1 and HRESP = 0.
3. Check if transfer is valid (HSEL=1, HREADYIN=1, HTRANS[1] =1).
4. If HWRITE=1, write HWDATA to REG0 or REG1 based on HADDR[3:2].
5. If HWRITE=0, read data from REG0 or REG1 and place it on HRDATA.
6. For invalid address, output HRDATA = 0xDEADBEEF.
7. Repeat for every valid transaction.

Advantages of AHB-Lite:

8. Fully synthesizable and timing-clean (Genus compatible).
9. Complies with AHB-Lite standard protocol.
10. Single-cycle operation with no wait states.
11. Compact and easy to integrate in FPGA/SoC systems.
12. Independent read/write handling for better performance.
13. Easily extendable for more registers or features.

Applications of AHB-Lite:

1. Control/configuration interface for peripherals (GPIO, ADC, timers).
2. Used in AHB-based SoC architectures.
3. Educational use for AMBA protocol learning.
4. Register interface for IP cores or accelerators.
5. Can be upgraded for full AHB or AXI protocol support.

III. LIMITATIONS

1. Both AXI4-Lite and AHB-Lite designs are simplified interfaces with limited functionality.
2. They support only basic single transactions without burst or advanced features and are restricted to two 32-bit registers.
3. They lack error handling, protection mechanisms, and power-saving features such as clock gating or low-power modes.

IV. RESULTS AND DISCUSSION

Clock-driven READ and WRITE operations synchronized with the rising and falling edges of HCLK. After RESET deassertion as shown in the Figure 3 gives, the master asserts HSEL and valid HTRANS signals during WRITE cycles to store data (0xAAAA1111 and 0xBBBB2222) into two registers. The slave responds with HREADYOUT = 1 and HRESP = 0 (OKAY), indicating successful transfers. During READ cycles, HWRITE = 0, and the slave drives HRDATA with the stored values aligned to the next clock edge. Overall, the waveform confirms proper handshaking, address decoding, and data transfer synchronization with the CLOCK.

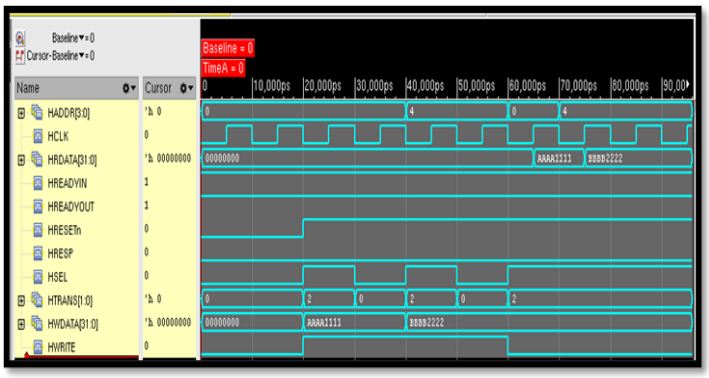

Figure 3: waveform of Simple AHB-Lite Slave Interface

The cell area report of the Simple AHB-Lite Slave Interface shown in the Figure 4 showing the total utilized area and distribution of logic cells in the design. Figure 5 represents the power report, indicating both dynamic and leakage power consumption under normal operating conditions. Figure 6

represents the timing report, confirming that the design meets setup and hold requirements for reliable operation. Figure 7 represents the simulation waveform of the AHB-Lite Slave Interface, illustrating correct read and write transactions synchronized with the clock.

```
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Nov 02 2025  04:15:11 pm
  Module:                 ahb_lite_slave
  Technology libraries:   slow
                          physical_cells
  Operating conditions:   slow
  Interconnect mode:      global
  Area mode:              physical library
=============================================================

   Instance    Module  Cell Count  Cell Area  Net Area   Total Area
-------------------------------------------------------------------
ahb_lite_slave              176    2668.073   727.552     3395.625
```

Figure 4: cell area report of Simple AHB-Lite Slave Interface

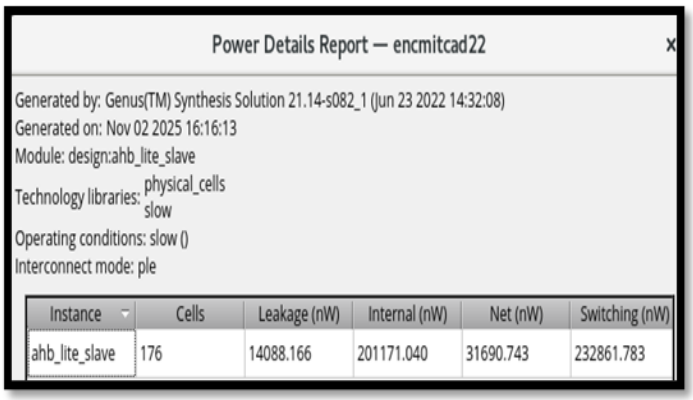

Figure 5 : Power report of Simple AHB-Lite Slave Interface

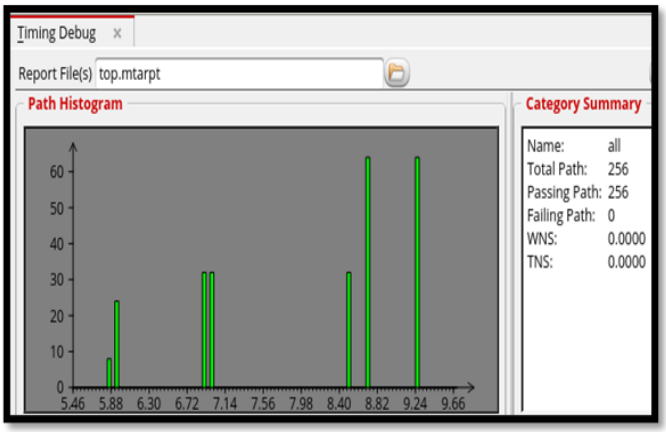

Figure 6 : Timing report of Simple AHB-Lite Slave Interface

Proper protocol handshaking and data flow during both write and read transactions as shown in the Figure 7 . After the reset is deasserted, the master initiates a write operation by asserting AWVALID and WVALID along with the address and data. When the slave responds with AWREADY and WREADY, the address and data are successfully accepted, and shortly after, the slave asserts BVALID to signal the completion of the write. Once the master acknowledges with BREADY, the write response is cleared. In the read phase, the master asserts ARVALID with the desired address, and the slave responds with ARREADY. Following this, the slave outputs the corresponding data on RDATA and asserts RVALID. The master then acknowledges with RREADY, completing the read transfer. Throughout the waveform, all transactions are synchronized with the rising edges of ACLK, showing smooth, single-cycle operations and correct AXI4-Lite handshake sequences without wait states.

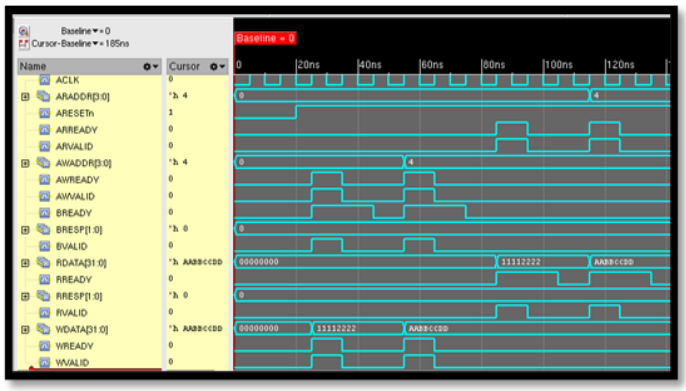

Figure 7: waveform of Simple AXI4-Lite Slave Interface

The waveform of the Simple AXI4-Lite Slave Interface shown in the Figure 7 proper read and write operations synchronized with the clock. Figure 8 represents the cell area report, indicating the total utilized area and logic cell distribution within the design. Figure 9 represents the power report, showing dynamic and leakage power consumption during operation. Figure 10 represents the timing report, confirming that the design meets setup and hold requirements, ensuring stable and reliable performance.

```
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Oct 27 2025  04:00:36 pm
  Module:                 axi_lite_slave_simple
  Technology libraries:   slow
                          physical_cells
  Operating conditions:   slow
  Interconnect mode:      global
  Area mode:              physical library
=============================================================

      Instance       Module  Cell Count  Cell Area  Net Area   Total Area
-------------------------------------------------------------------------
axi_lite_slave_simple               184    2855.784   759.771     3615.555
```

Figure 8: cell area report of Simple AXI4-Lite Slave Interface

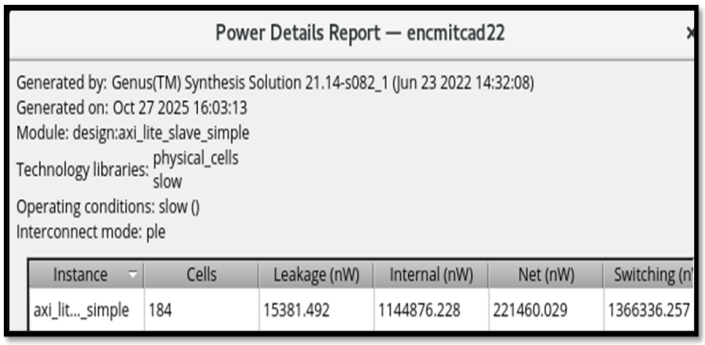

Figure 9 : Power report of Simple AXI4-Lite Slave Interface

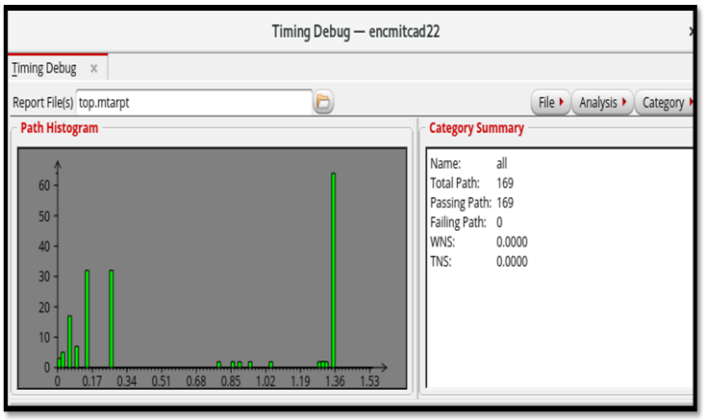

Figure 10 : Timing report of Simple AXI4-Lite Slave Interface

V. Conclusion

Both designs meet timing constraints with zero violations. The AHB-Lite slave's sequential single-channel design ensures predictable timing and lower dynamic power. AXI-Lite achieves higher concurrency through parallel read and write channels, at the cost of increased switching activity and silicon area.

Table 2: Comaparision Table

Metric	AHB-Lite Slave	AXI4-Lite Slave	Observation
Cell Count	176	184	AXI-Lite uses slightly more logic.
Total Area (µm²)	3395.6	3615.5	AXI-Lite occupies ~6% more area.
Total Negative Slack (ns)	0.000 / 0.000	0.000 / 0.000	Both achieve full timing closure.
Total Power (W)	2.47×10^{-4}	1.38×10^{-3}	AXI-Lite consumes ~5× more power.
Leakage Power (W)	1.43×10^{-5}	1.54×10^{-5}	Nearly identical leakage.
Register Power (%)	87.1	82.3	Register logic dominates in both.

References

[1] ARM Ltd., AMBA AHB-Lite Protocol Specification, ARM IHI 0033A, 2010.

[2] ARM Ltd., AMBA AXI-Lite Protocol Specification, ARM IHI 0022E, 2011.

[3] Cadence Design Systems, *Genus™ Synthesis Solution User Guide*, Version 21.14, 2024.

[4] D. Flynn et al., *AMBA System-on-Chip Bus Specification*, ARM Ltd., 2020.