Capstone Project 2

Week 3 – Semester 233

Student: Vo Viet Hung – 2051076

**Implementation of AHB-APB and APB-UART bridges**

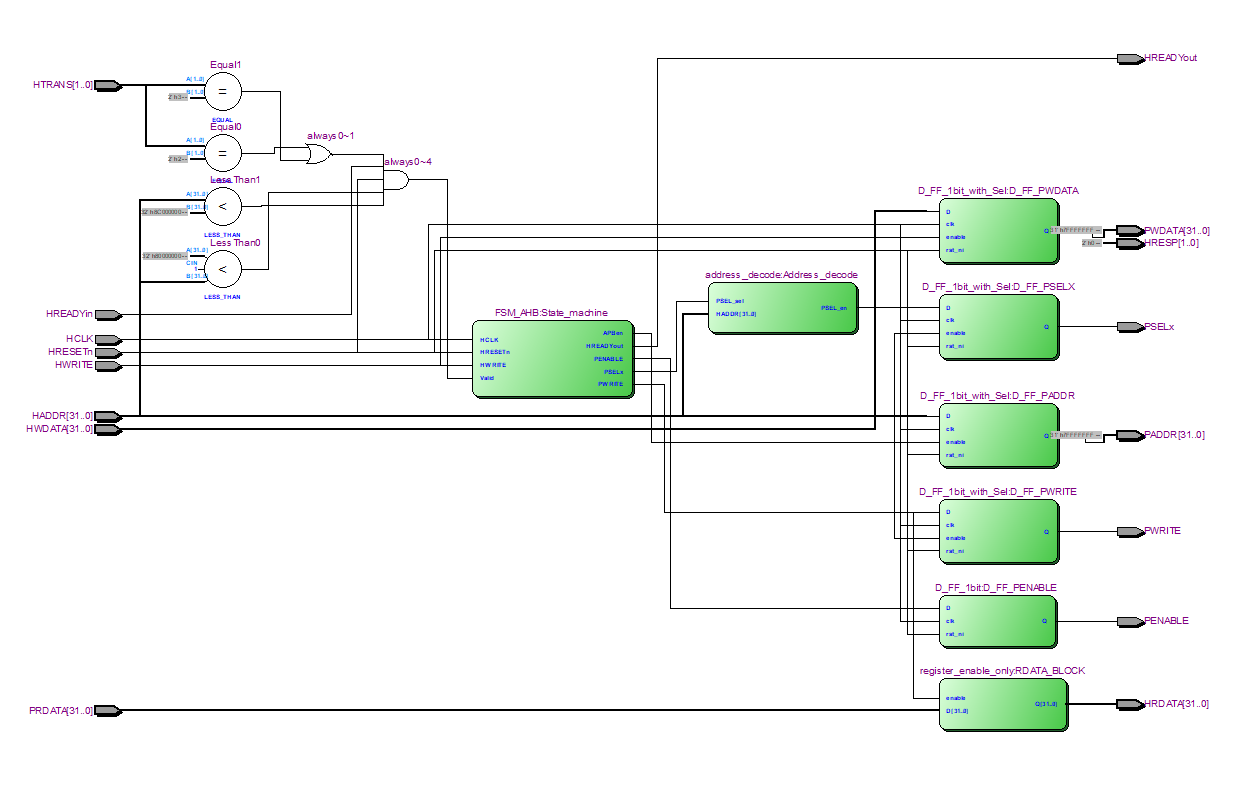
# The first problem: APB Bridge

# ***Figure 1: Block diagram of bridge module***

## Summary

During week 30, I have focused on developing and testing the Advanced Peripheral Bus (APB) bridge for part of my project. I started by reviewing technical documents related to structure and operational FSM of APB. Next, I reimplemented the schematic by add additional sub-signal to make sure that the conversion from Advanced High-performance Bus (AHB) to APB operate as similar as given information of documents. This process involved adjusting the signal during each state inside state-machine of APB Bridge and data format to meet APB output signal requirements. I also conducted several tests to ensure the bridge worked correctly and efficiently, including signal testing and states’ generating. The results showed that the APB bridge operated stably and met the technical requirements effectively.

## Solution

For the finite state machine of the AHB slave (APB Bridge), I reviewed the related documentation [DDI0170](https://developer.arm.com/documentation/ddi0170/latest/Designer-s-Guide), including its references’ Block in document, Input signal (Valid, HregWrite, PSELAPBif, HTRANS), and Output requirements and others to ensure that state transitions do not alter the behavior of certain flags, such as the PWRITE, PSEL, PENABLE. Regarding the D - FF blocks, which help synchronize the input signals of the AHB and the output signals of the APB, I adjusted the signals (ENABLE) within the state diagram to produce the correct signals as mentioned in the documentation. Additionally, I did spent additionally times with implemented FSM with its generated case, signals to ensure data from AHB side to APB accurate outputs. Finally, to me, the operate of this intermediate feature is fine.

# ***Figure 1.2: Block diagram implementation of bridge module***

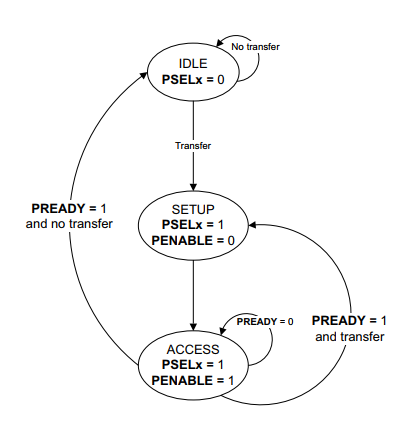
# The second problem: APB UART

***Figure 2: APB UART general block diagram***

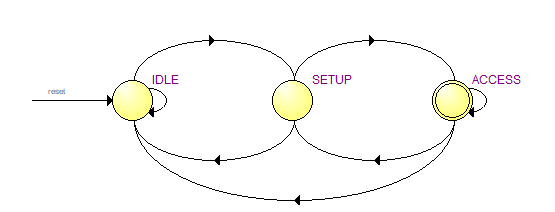
## Summary

***Figure 2.1.1: APB UART buffering***

The figure 2.1.1 above is the overall operating methodology of APB UART inside the block diagram.

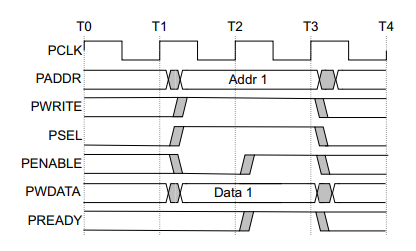
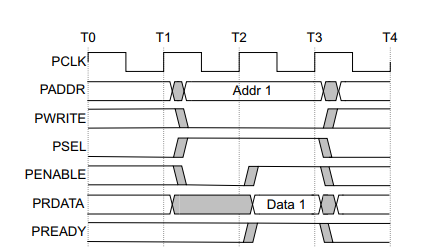
During this week, I focused on integrating the APB with the UART module. I started by reviewing the technical specifications [AMBA APB Protocol Specification](https://developer.arm.com/documentation/ihi0024/latest/) and requirements for the UART interface [APB UART Programmers model](https://developer.arm.com/documentation/ddi0479/d/apb-components/apb-uart) . Therefore, I come with this state diagram:

***Figure 2.1.2: State diagram of APB interface***

And generated this implementation version based on it:

***Figure 2.1.3: State diagram of APB interface implementation***

Where: I have adjusted PENABLE and PSEL signals are generated from Master (APB Bridge), and PREADY is returned if PENABLE and PSEL is asserted. For implement below operational:

***Figure 2.1.4: Write transfer with no wait states***

***Figure 2.1.5: Read transfer with no wait states***

Then, I designed the necessary logic to handle the communication between the APB and UART, ensuring proper data formatting and timing of two write and read transfer operation.

## Solution

### Solution I didn’t choose

For writing and reading transfer operation, there are two types: with no wait and with wait states. For the reason that why I didn’t choose operations with wait states is that this operation will make the system more complexity (additional signal, operating states I guess), which made it slower as they cannot complete in a single cycle; Moreover, by adding external signal for making wait states, this method also affect bus utilization and overall system performance (require more adjust).

* + 1. **Solution I choose**

I did choose no wait states, therefore, come to FSM of APB interface, each of writing and reading transaction, will enable to FIFO buffer appropriate.

# Problems arise while I’m doing the project

## Problem 1: APB UART Buffer

I have not experienced with TX FSM and RX FSM yet, because firstly, I have lack of time this week due to my internship. Secondly is that I have not found the example design or references for this stage.

### Proposed Solution

However, after spending more time, I have come to this example on [Design and Implementation of High-Speed Universal Asynchronous Receiver and Transmitter (UART)](https://ieeexplore.ieee.org/document/9070856), unfortunately, I don’t have right to access for this, but I do have example feature of it, so during the next week, I will try to do with this state diagram, hope to get some more information with UART.

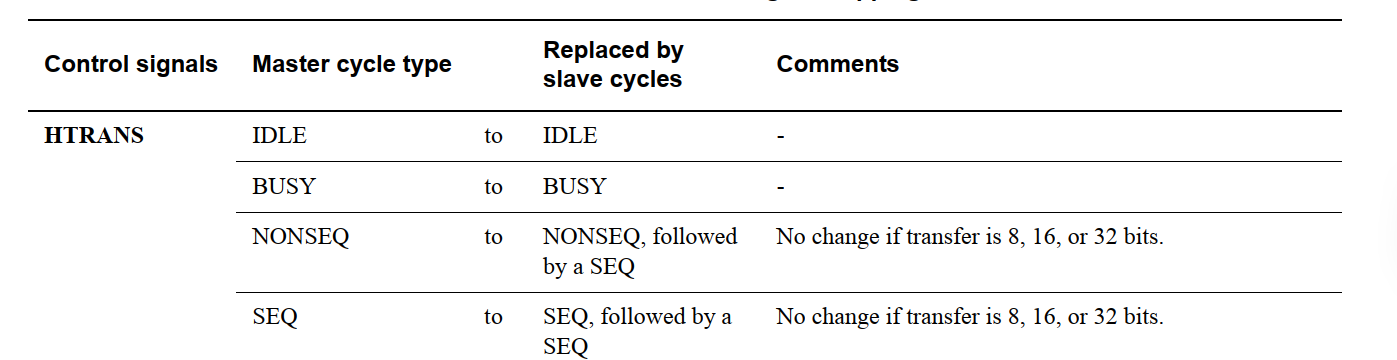
## Problem 2: APB Bridge

Although I have completed with the output signal of the APB Bridge and get some correct results based on testing its operation, I still struggle with an additional signal name HTRANS where there are four stages: NONSEQ, SEQ, BUSY, ERROR indicates the type of the current transfer.

## 3.2.1. Proposed Solution

***Figure 3.2.1: Bus matrix module components***

I have found this BUS matrix for communication between AHB – AHB Slave (APB Bridge) - APB UART. And based on this design, HTRANS will take place at the signal to indicate whether the next transaction is ready or not.

So next week, I will continue with this in order to prepare for complete communication between AHB and APB to UART.

# References

1. *AMBA APB Protocol Specification,* <https://developer.arm.com/documentation/ihi0024/latest/>
2. *Arm Cortex-M System Design Kit Technical Reference Manual r1p1,* <https://developer.arm.com/documentation/ddi0479/d/apb-components/apb-uart>
3. Lakshmisagar H.S1, Sumathi M.S2 1,2Assistant Professors, BMS Institute of Technology, Bangalore, Karnataka, India, *Design and Verification of APB Compliant Quad Channel UART* <https://www.ijraset.com/fileserve.php?FID=9093>
4. A. K. Gupta, A. Raman, +1 author Ravi Ranjan, *Design and Implementation of High-Speed Universal Asynchronous Receiver and Transmitter (UART).* <https://www.semanticscholar.org/paper/Design-and-Implementation-of-High-Speed-Universal-Gupta-Raman/a8796177ab085bddf1836fda479085c66f3a3607>