**HO CHI MINH CITY NATIONAL UNIVERSITY**

HO CHI MINH UNIVERSITY OF TECHNOLOGY

FACULTY OF ELECTRICAL - ELECTRONICS

**DEPARTMENT OF ELECTRONICS**

---------------o0o---------------

****

**CAPSTONE PROJECT 2 REPORT**

**DESIGN AMBA AHB -APB SPECIFICATION**

**COMMUNICATE IN UART**

|  |  |
| --- | --- |
| **Instructor:** | **Dr. Trần Hoàng Linh** |
| **Student’s name:** | **Võ Viết Hưng** |
| **Student’s ID:** | **2051076** |

**Hồ Chí Minh city, July 14th, 2024**

**TABLE CONTENT**

[1. **REVISION** 1](#_Toc501486126)

[2. **PROGRESS OF PROJECT** 5](#_TOPIC’S_CONTENT)

[4. **PLAN FOR NEXT TWO WEEKS WORKING** 5](#_TOPIC’S_CONTENT)

[4. **IMPLEMENTATION PLAN** 5](#_TOPIC’S_CONTENT)

[5. **REFERENCES** 13](#_4._REFERENCES)

# 1. REVISION

During weeks 29-30, as I have failed to report (unsaved file before sending), I also completed the AHB-APB Bridge block. This task involved ensuring the proper bridging between the AHB and APB buses, allowing for seamless data transfer between the two protocols. The completion of this bridge was crucial for the overall functionality of the system.

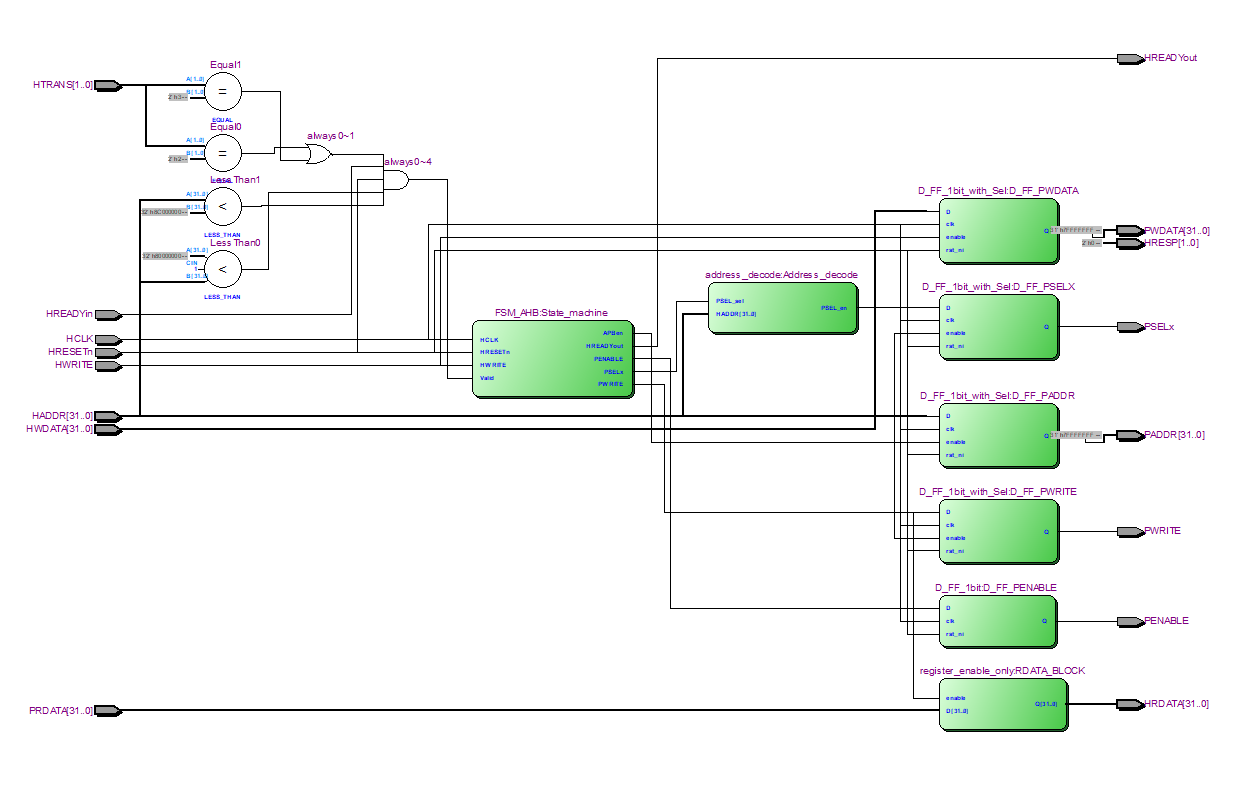
Next, over the next two weeks (31-32), I have focused on developing and finalizing the APB UART block. Initially, I reoutlined the work plan for implementing the APB UART and the UART baud rate generator. Then, after being verification, the baud rate generator block encountered a functional issue (I have used wrong mux selection and also used the baud rate divisor two large -> lead to the output of this block is just capable for only divisor path and the output clock is too slow (by divided in to 8 only, it made the clock out need to reach nearly 5200 # of rising edge of the default clock in the baud rate of 9600)), causing the output signals to not perform as its functionality. To address this, I reanalyzed its operation and did some improvement included with critical components of the APB UART blocks, such as the FSM RX-TX, RX-TX FIFO, and their SHIFT register.

The entire process required thorough testing and troubleshooting, particularly to ensure the compatibility and accuracy of the APB output signals during data transmission and reception. I dedicated significant time to verifying and extensively testing all components, ensuring they operate synchronously and reliably. Until this report, both the APB UART block and the AHB-APB Bridge block have nearly achieved the desired stability and performance as I have expected based on references (except for not combine them together yet). So, currently, I am making it ready for the next stages (combine to APB Bridge) of the project.

# 2. PROGRESS OF PROJECT

## *2.1. AHB – APB Bridge*

During week 29 - 30, I have focused on developing and testing the Advanced Peripheral Bus (APB) bridge, 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.

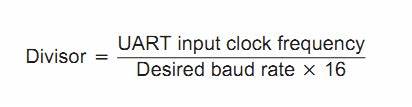
Especially, for the finite state machine of the AHB slave (APB Bridge), I reviewed the related documentation [DDI0170](https://developer.arm.com/documentation/ddi0170/latest/Designer-s-Guide), including its references’ block in document, input signal (**Valid**, HregWrite, **PSELAPBif**, **HTRANS**), 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 2.1: Complete block implementation of AHB – APB bridge module*

## *2.2. APB UART*

***Figure 2.2: General Block Diagram of APB UART***

### ***2.2.1. Baud rate generator***

Baud rate generator takes place as an internal clock generator for this block (named baud tick), so depend on the desired\_baud\_rate signal, it will create the UART clock pulse in compatible.

***Figure 2.2.1.: Baud rate divisor references formula***

So, as I have explained above, I have regenerated this block for decrease the number of clock divisor need to count up to generate out one clock of APB UART.

|  |  |  |
| --- | --- | --- |
| **No.** | **Desired Baud rate** | **Baud rate divisor** |
| 1. | 600 | 5208 |
| 2. | 4800 | 651 |
| 3. | 9600 | 325 |
| 4. | 19200 | 162 |
| 5. | 28800 | 108 |
| 6. | 38400 | 81 |
| 7. | 57600 | 54 |
| 8. | 115200 | 27 |
| 9. | 230400 | 13 |
| 10. | 460800 | 6 |
| 11. | 921600 | 3 |

***Table 2.2.1: Baud rate divisor references***

### ***2.2.1. RX – TX process***

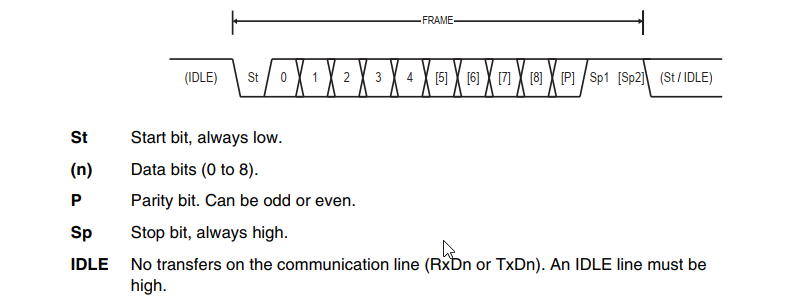
***Figure 2.2.1: General TX FSM & RX FSM operation***

In week 30, I focused on designing and testing the finite state machines (**FSM**) for the transmitter (**TX FSM**) and receiver **(RX FSM**) of the **UART APB**. However, in the previous week (31), after some more testing I have redesign the RX FSM whether this state no requirement to check for the data into it is available or not, just focus on the start, parity and stop bit only. Therefore, with little adjustment, I have reduced the IDLE state one for shorter code length.

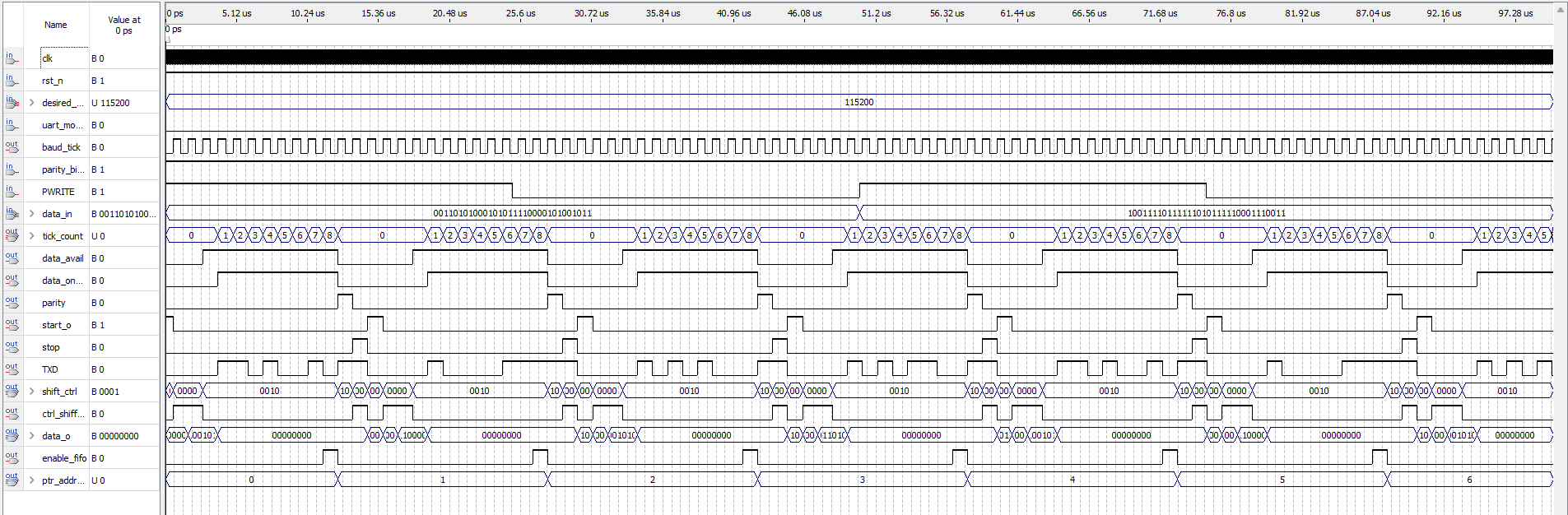
To be more specific, firstly, for the **TX FSM** & **RX FSM**, I identified their main states: **IDLE**, **START**, **DATA[0:7]**, **PARITY**, **STOP** (except for IDLE in RX FSM). I constructed a detailed states’ diagram and defined the state transition conditions based on control signals and the status of the **UART**. After finalizing the diagram, I wrote the **RTL** code and ensuring the synchronization and accuracy of each state.

After completing the design, I set up a simulation environment to test both the TX FSM and RX FSM under various operating conditions. I created test such as inputting randomly data to check whether the data come out is correct or not, assuming no parity\_bit, start\_bit, stop\_bit to ensure that the FSMs functioned correctly as designed. During this process, I debugged and optimized the code by generating more necessary signal to send back the signal come out from FSM for further combination with other process. Then, based on simulation and test results, I able to ensure that the FSMs operated efficiently and reliably.

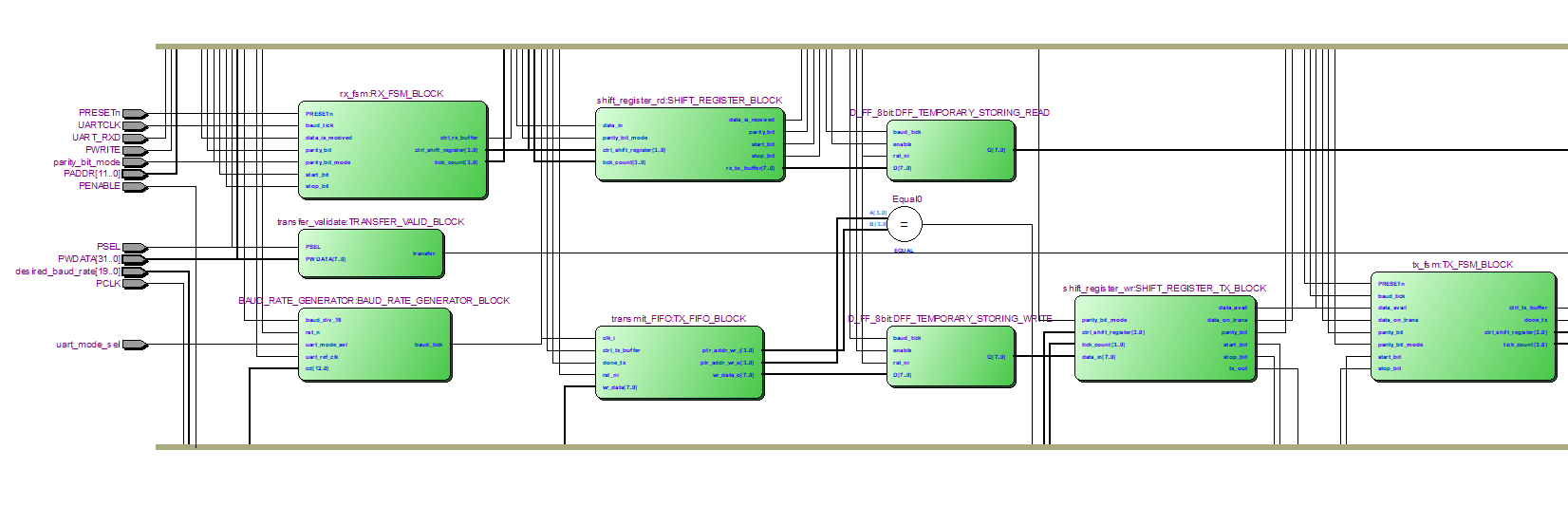
**Here is the data frame signal I currently apply on my design:**

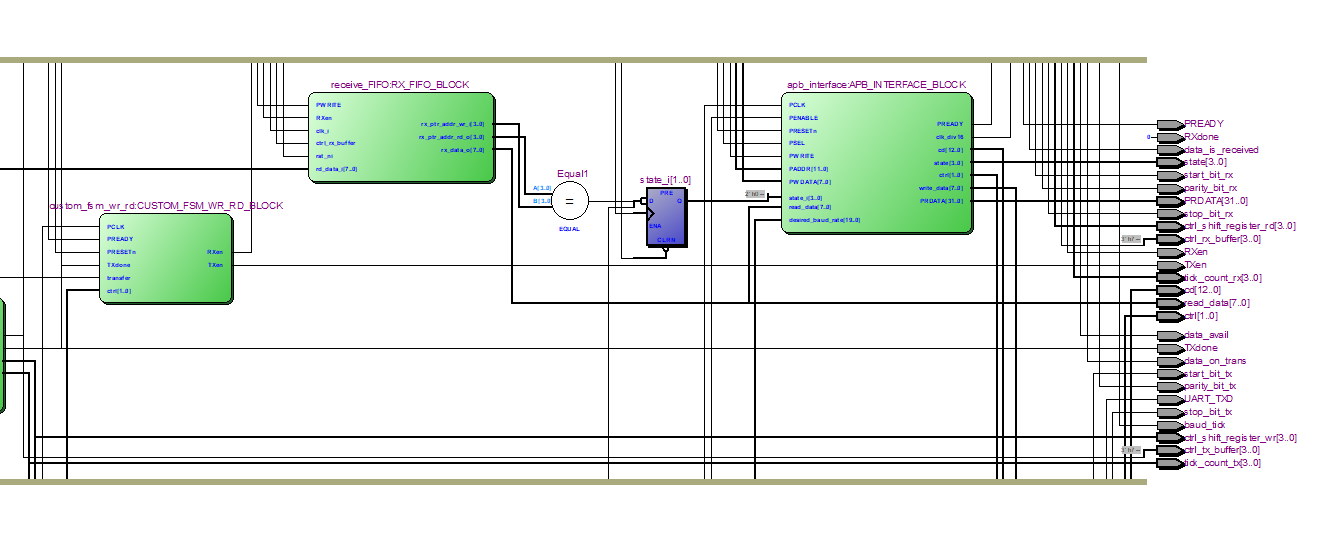
Based on other design methodologies above ([Reference idea for TX and RX FSM](https://vn.images.search.yahoo.com/search/images;_ylt=Awrx_yPb3qdmvCkachdrUwx.;_ylu=Y29sbwNzZzMEcG9zAzEEdnRpZAMEc2VjA3BpdnM-?p=TX+FSM+UART&fr2=piv-web&type=E210VN91215G0&fr=mcafee#id=3&iurl=https%3A%2F%2Fuser-images.githubusercontent.com%2F106643865%2F199658638-16d30460-33a9-4ffe-82d4-aa939696fca9.jpg&action=click)) and UART’s operation in transmit and receive data frame below, I have concluded and regenerated my own FSM of TX and RX related to my capability.

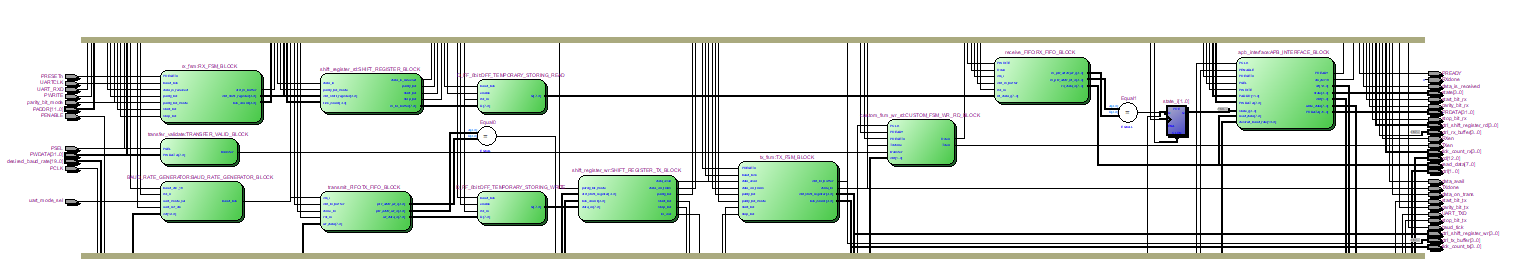
***Figure 2.2.2: TX & RX data frame references***

**Here is an example of TX simulation:**

***Figure 2.2.3: Testing Operation of TX UART***

And following is the nearly final figure of APB UART that I have done and on checking:

***Figure 2.2.4: The first part of RTL view of APB UART Block design reference***

***Figure 2.2.5: The part of RTL view of APB UART Block design reference***

***Figure 2.2.6: The full view of RTL of APB UART Block design reference***

# 3. PLAN FOR NEXT TWO WEEKS WORKING

Over the next two weeks, my primary focus will be on integrating two critical blocks: the APB UART and the AHB-APB Bridge. This integration is a crucial step to ensure that the system can smoothly transmit data between protocols and that all components operate in sync without errors. This is a complex task that requires careful testing and troubleshooting to ensure that the output signals from the UART block, through the APB Bridge, are accurate and stable as they reach other components.

In parallel with the integration work, I also plan to meet with Mr. Hải for a short discussion about my Capstone Project progress. The goal of this meeting is to address the remaining issues that arise during the integration process and to propose and test specific applications to verify the completeness of the design.

# 4. IMPLEMENTATION PLAN

* ***Dissertation time***: 2 months
* ***Detail Process Plan:***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Content** | **July** | **August** |
| 1 | ***Explore possible designs*** | ✓ |  |
| 2 | ***Design implementation*** | ✓ |  |
| 3 | ***Design implementation*** | ✓ |  |
| 4 | ***Design implementation*** | ✓ |  |
| 5 | ***Unification design*** |  | ✓ |
| 6 | ***Complete design*** |  | More than ½ ✓ |
| 7 | ***Insert Application check*** |  |  |
| 8 | ***Report*** |  |  |

# 5. REFERENCES

1. ***AMBA AHB Protocol Specification*** <https://developer.arm.com/documentation/ihi0033/latest/>
2. ***AHB Example AMBA System Technical Reference Manual*** <https://developer.arm.com/documentation/ddi0170/a/I967114>
3. ***DirectCore Advanced Microcontroller Bus Architecture - Bus Functional Model*** <https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocuments/UserGuides/CoreAMBA_BFM_UG.pdf>
4. ***Effective Design and Implementation of AMBA AHB Bus Protocol using Verilog*** <https://www.researchgate.net/publication/337510558_Effective_Design_and_Implementation_of_AMBA_AHB_Bus_Protocol_using_Verilog>
5. ***AMBA AHB Protocol*** <https://fr.scribd.com/doc/41197279/AMBA-AHB-Protocol-Presentation>
6. ***PrimeCell UART (PL011) Technical Reference Manual*** <https://developer.arm.com/documentation/ddi0183/g>
7. ***Design and FPGA Implementation of UART Using Microprogrammed Controller***

<https://www.researchgate.net/publication/282030059_Design_and_FPGA_Implementation_of_UART_Using_Microprogrammed_Controller>