Capstone Project 2

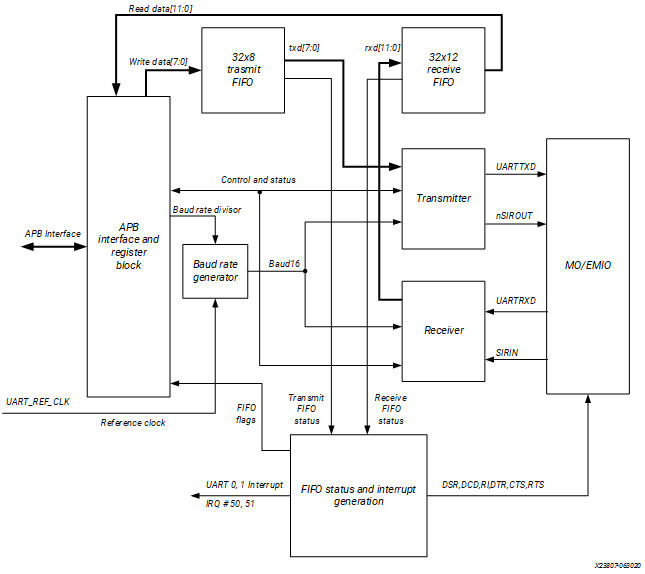
Week 5 – Semester 233

Student: Vo Viet Hung – 2051076

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

# The problem: APB UART

In the past week, I have completed several important tasks related to the APB UART block. First, I have finalized and tested the key components of this block, including the finite state machine (FSM) for data reception (RX) and transmission (TX), the FIFO buffers for RX and TX, and the SHIFT registers. These components have been integrated and thoroughly tested to ensure accurate operation. Additionally, I have performed checks and debugged the necessary output signals of the APB during the debugging process, aiming to ensure the APB UART block operates stably and meets design requirements. This work helps ensure that the UART communication system via the APB operates efficiently and reliably.

* 1. **Summary**

***Figure 1.1: General Block Diagram of APB UART***

In the past week, I focused on addressing critical issues within the APB UART block to enhance its functionality and reliability. Two significant areas of improvement were the baud rate generator and the TX FIFO block.

***Baud Rate Generator Adjustment***:

- Issue: In the previous design, the input signals (***PCLK*** & ***UART\_REF\_CLK***) handling were incorrect, leading to clock signal (or I can ***baudtick***) generated out when the **mode\_uart\_sel (select which clock to use)** input set was wrong (I set up that there will be two types of clock input into baudrate generator due to the design in ***Figure 1.1***).

- Solution: I rearrange the mux to select which clock is used to correctly interpret the input signals, ensuring that when user choose whether clock input is, the baudtick clock signal generation is based on that.

***TX FIFO Block Enhancements:***

- Issue: The initial design of the TX FIFO block accepted a 32-bit input signal and divided it into four 8-bit registers. This design was not well application because I did based on the operation of STM32 MCU UART’s functions in which it only transfer a packet of data in type 8 – 16 bits each transfer, and also register of some board communication only need for 16 bits data or registers for each transfer. Therefore, I have adjusted to which is used only 16 bits data at max (depend on user also so there would be an external signal for selecting transfer mode if transfer 8 bits mode or 16 bits mode). The next thing come out from this block is that I have also taken out external signal such as tx\_add\_ptr\_wr and tx\_addr\_ptr\_rd to check whether the state of TX FIFO incase the incoming data is too fast lead to wrong transmission.

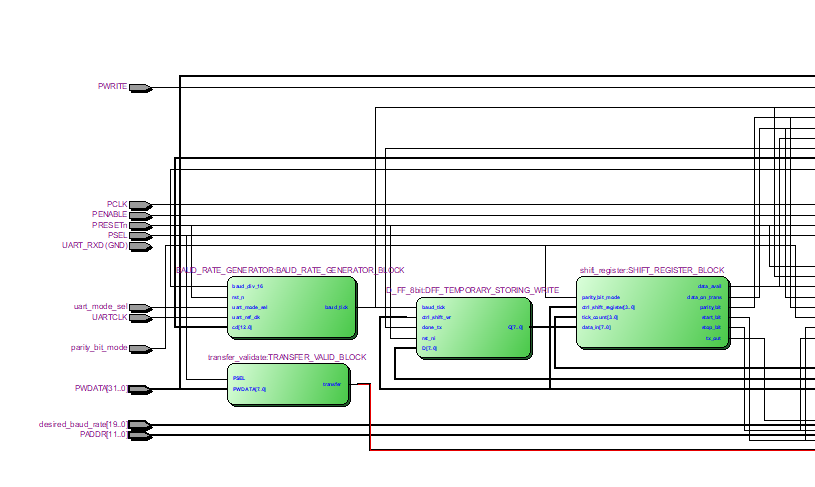
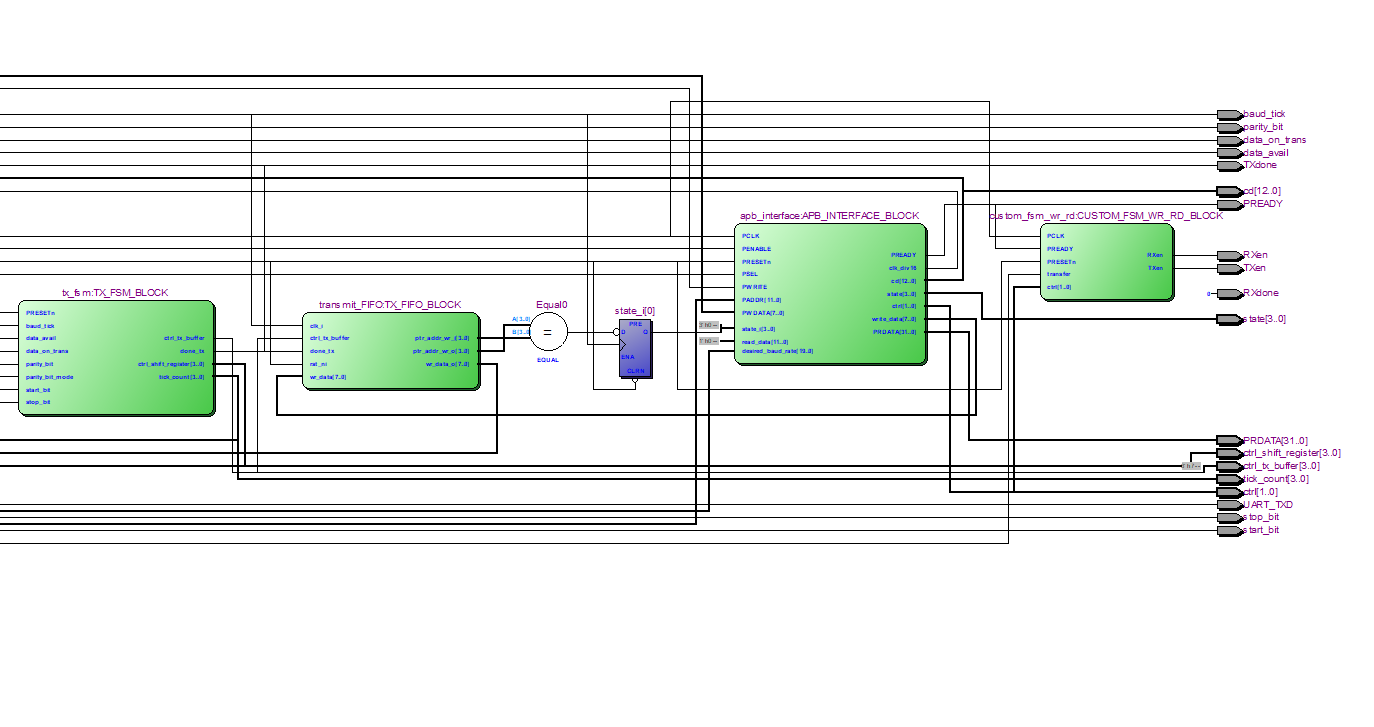
- Solution: I modified the TX FIFO block to accommodate 16-bit or 8-bit input signals directly. This adjustment involved resizing the input handling to ensure proper data segmentation and storage within the FIFO.

***APB Interfacing Block:***

I did some more adjustments related to this block such as the register block for search into which functions, if the base offset (PADDR) is in the range of 0 to 10 it will show the following function such as for address 0x000 this will require for sending or reading based on ***PWRITE*** signal, or for address 0x010 this is used require for setting up baud rate used by internal clock (counter divisor and divisor clock). The updated design was thoroughly tested to verify its functionality with different input sizes. Related to this field, I know more about how a transaction between master (CPU) to other slave component in whole construct of CPU processing.

These improvements contribute to the overall result in which I have succeed in the side of APB UART where I have completed with the side of TXD. However, I did an external mistake is that I have misunderstood of this block operating so I added an APB Interface FSM which led to the wrong operation in the APB Bridge.

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

***Figure 1.1.2: The first part of APB UART Block design reference***

***Figure 1.1.3: The remain part of APB UART Block design reference***

* 1. **Planning for next week working**

For the next week, I have figured out ways for solving my problems, first is that I will go deeply more in fixing the operation of APB UART first, where I continue assuming random PENABLE and PSEL for check data out of TXD and in RXD. Then I will combine the two block APB Bridge and APB UART together for showing some results.

**Thanks for reading my Journal.**

# 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>