Transmitter Functional Design Specifications

Induction Training

Version 1.0

Block Owner

Si-Vision

Authors

Ayman Mohamed Helal Abozeid

Yomna Ahmed Mohamed Abdallah

This Page is left Blank Intentionally

1 Table of Contents

2 Revision History

Version	Date	Author(s)	Revision Notes	Owner Approval
1.0				

3 Overview

The project is considered as the interface that connect the APB bus and the UART, which is used to convert the UART transactions to APB transactions and vice versa. The device has a MCU that can access a memory of the remote device by sending a write request with the data wanted to be written in the memory of the remote device or a read request and wait for the remote device to response with the required data. The MCU can access the UART using an APB Bus and the UART can access the memory of the local device using another APB bus.

4 Operation and Description

4.1 Digital Interface

4.1.1 Parameters Names

Parameter Name	Default	Description
DATA_WIDTH	32	The width of the data bus
ADDRESS_WIDTH	16	The width of the address bus

4.1.2 Ports Names

Port Name	Port Width	Port Type	Description
uart_clock	1	input	Input clock to UART
uart_reset	1	input	Input reset signal to UART
apb_clock	1	input	Input clock to APB
apb_reset	1	input	Input reset signal to APB
sdata_rx	1	input	Data in to UART RX
sdata_tx	1	ouput	Data out from UART Tx
rdata_master	32	input	Data in to APB master from the APB slave of system Mem
ready_master	1	input	Ready to APB master from the APB slave of system Mem
address_master	16	output	Address to access the system Mem
apb_en_master	1	output	Enable signal to activate the APB slave of system Mem
wdata_master	32	output	Data wanted to be written in the Mem
write_master	1	output	Write signal high when writing and low when reading
sel_master	1	output	Selection signal asserted high to start transitions
rdata_slave	32	output	Data to the APB master of system MCU
ready_slave	1	output	Ready to APB master of system MCU
address_slave	16	input	Address wanted to be accessed by system MCU
apb_en_slave	1	input	Ready to APB master from the APB slave
wdata_slave	32	input	Data wanted to be written in the other device
write_slave	1	input	Write signal high when writing and low when reading
sel_slave	1	input	Selection signal asserted high to start transitions

4.1.3 CDC Table

CDC signal	Source Domain	Destination Domain	Synchronization method
sdata_rx	uart_clock of the remote device	uart_clock	Double flop synchronizer
par_type	apb_clock	uart_clock	Double flop synchronizer
par_en	apb_clock	uart_clock	Double flop synchronizer
err_inj_type	apb_clock	uart_clock	None
err_inj	apb_clock	uart_clock	Double flop synchronizer
rx_wreq	uart_clock	apb_clock	Asy FIFO
rx_rreq	uart_clock	apb_clock	Asy FIFO
rx_rres	uart_clock	apb_clock	Asy FIFO
tx_wreq	apb_clock	uart_clock	Asy FIFO
tx_rreq	apb_clock	uart_clock	Asy FIFO
tx_rres	apb_clock	uart_clock	Asy FIFO
par_count_en	apb_clock	uart_clock	Double flop synchronizer
par_count	uart_clock	apb_clock	Double flop synchronizer with Grey Encoding
err_done	uart_clock	apb_clock	Double flop synchronizer
clk_rate	apb_clock	uart_clock	None

4.2 Functional Description

- Top module

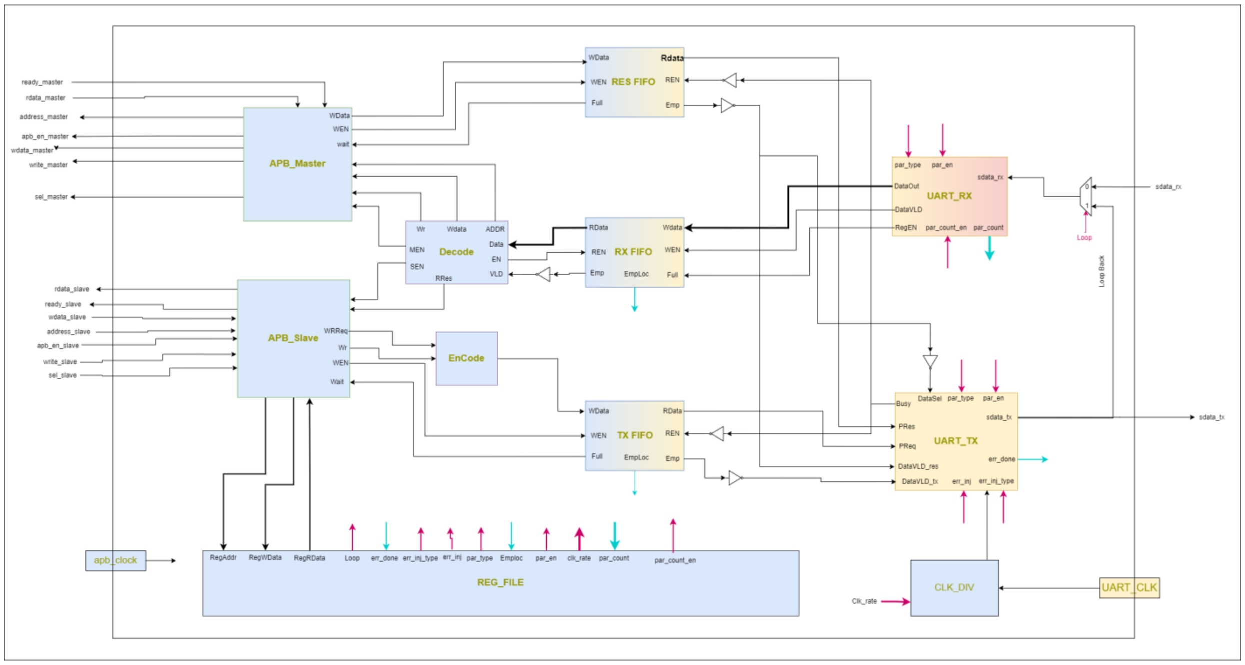

Figure 1 top module architecture

- <u>At Receiving</u>

- Serial data comes from the UART_tx of the remote device on the UART_rx of the local device which takes this data and check the parity error if there is no error the UART_rx will forward the 7 bytes data to FIFO2 else it will discard this frame and increase the counter of the parity error.
- FIFO2 handles the CDC issue which appears due to data travelling from UART clock domain to the APB clock domain.
- FIFO2 will forward the data to a decoder to decode the received data and understand the command type.
- The APB master will be the way to access the device memory and the APB slave will be the way to deal with the MCU of the device.
- The received data are 3 types write request, read request, and read response the write and read requests are forwarded to the memory of the device and the read response are forwarded to the MCU of the device.

- At transmitting

- The data wanted to be transmitted are 3 types write request, read request, and read response the write and read requests are sent from the MCU of the device and stored at FIFO3 and the read request are sent from the memory of the system and stored FIFO1.
- Data stored at FIFO1 is higher priority than data at FIFO3.
- The read and write request are forwarded to a encoder to encode the write request command and the read request command.
- Then the data are forwarded to the UART_tx to transmit it.

- Par_type signal is asserted to “1” if odd parity and asserted to “0” if even parity.

- UART_tx

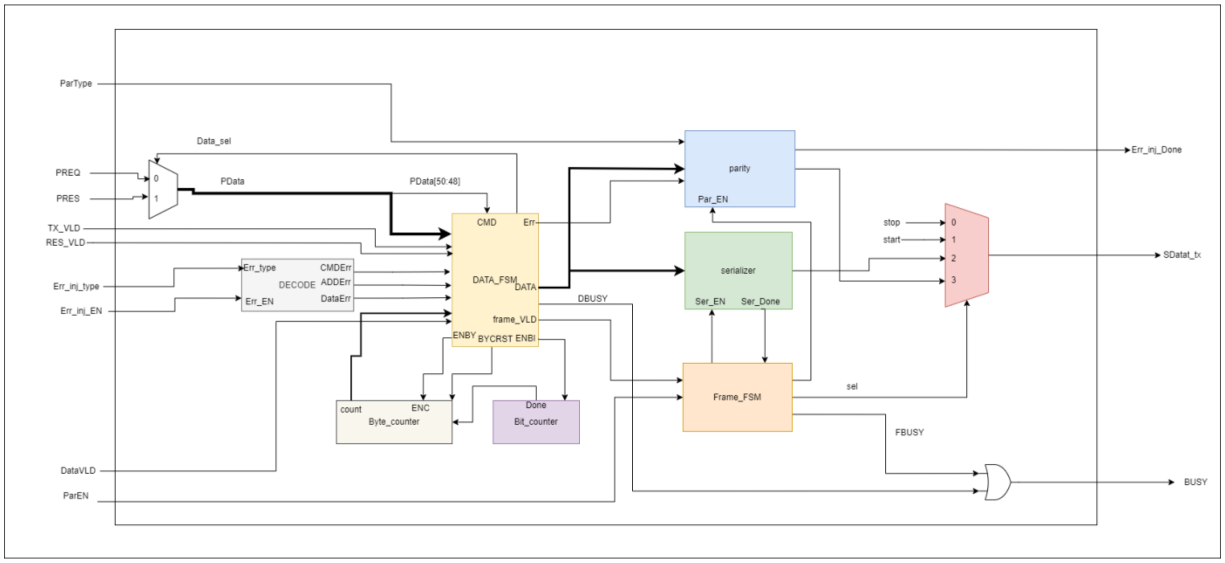

Figure 2 UART_TX architecture

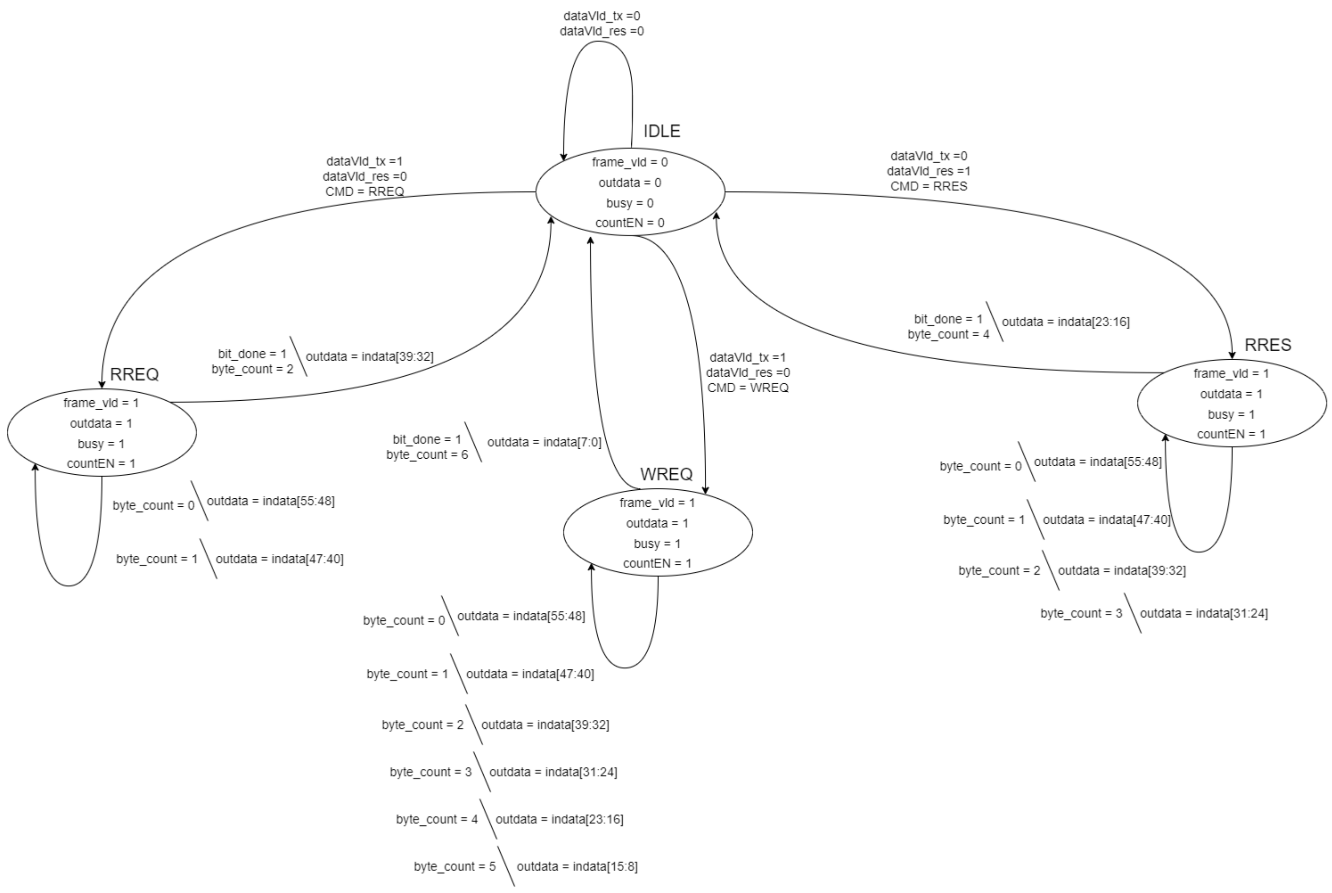

Figure 3 state diagram of the data finite state machine

- The parallel data enters to a finite state machine DATA_FSM to divide the command to bytes of data to be transmitted and to handle the Error injection option.
- Then the data will be forwarded to the sterilizer to convert the data from parallel to serial.
- The transmitted frame is controlled by the frame FSM.

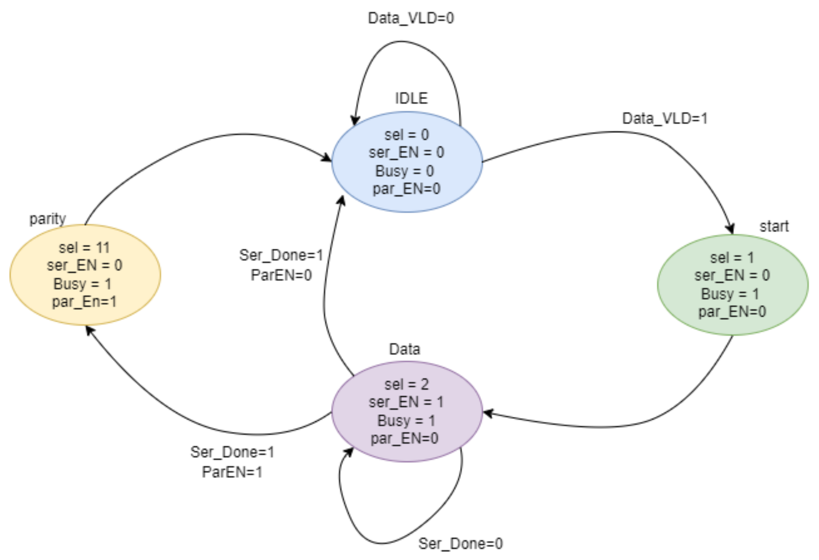

Figure 4 state diagram of the UART_TX frame finite state machine

- UART_rx

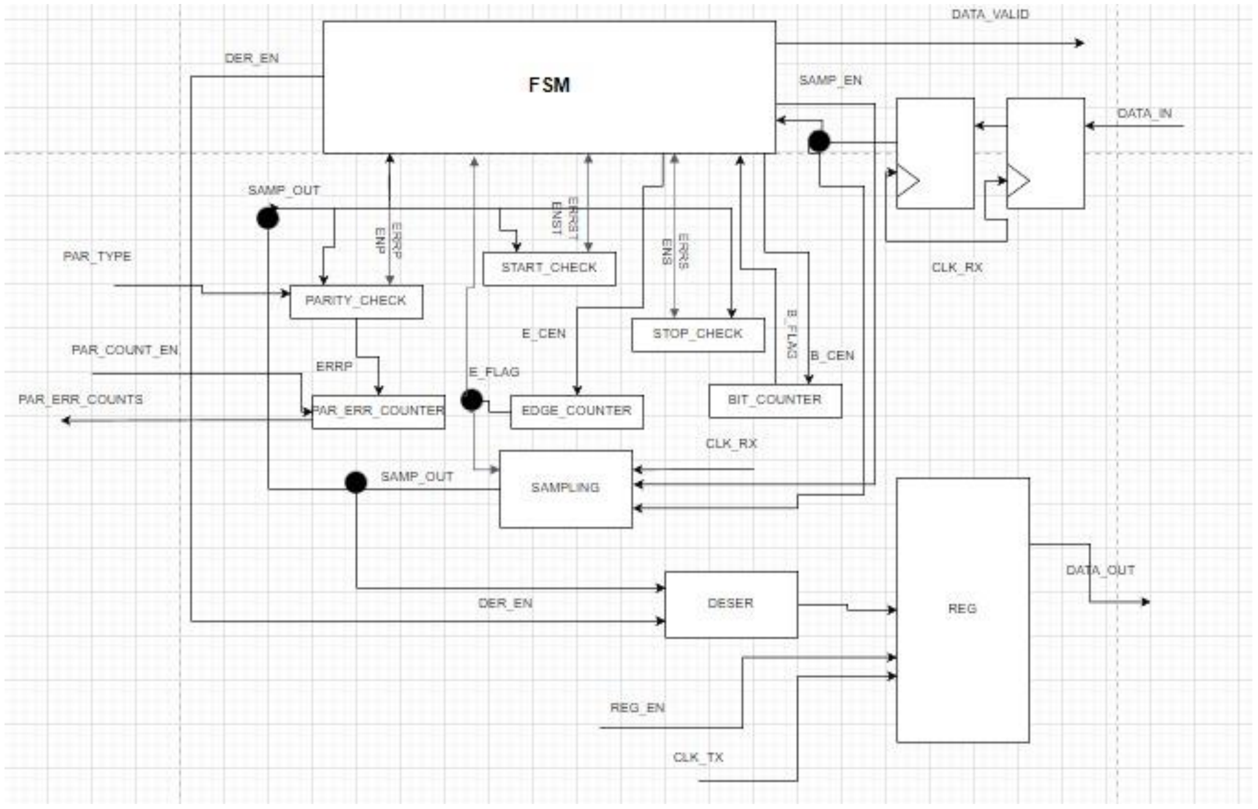

Figure 5 UART_RX architecture

- Serial synchronized data enter the FSM , since FSM detect that this data =0 it enters the next state one following another until the whole byte is transferred , the FSM assert SAMP_EN signal to allow sampling circuit to work and we sample the data at the 3 middle samples and give it to the deserializer and convert it to parallel then collect the whole command in the REG,
  We use counter to count number of errors that cased by the parity checks.

- The MAXIMUM ERROR in sampling will be one clock cycle

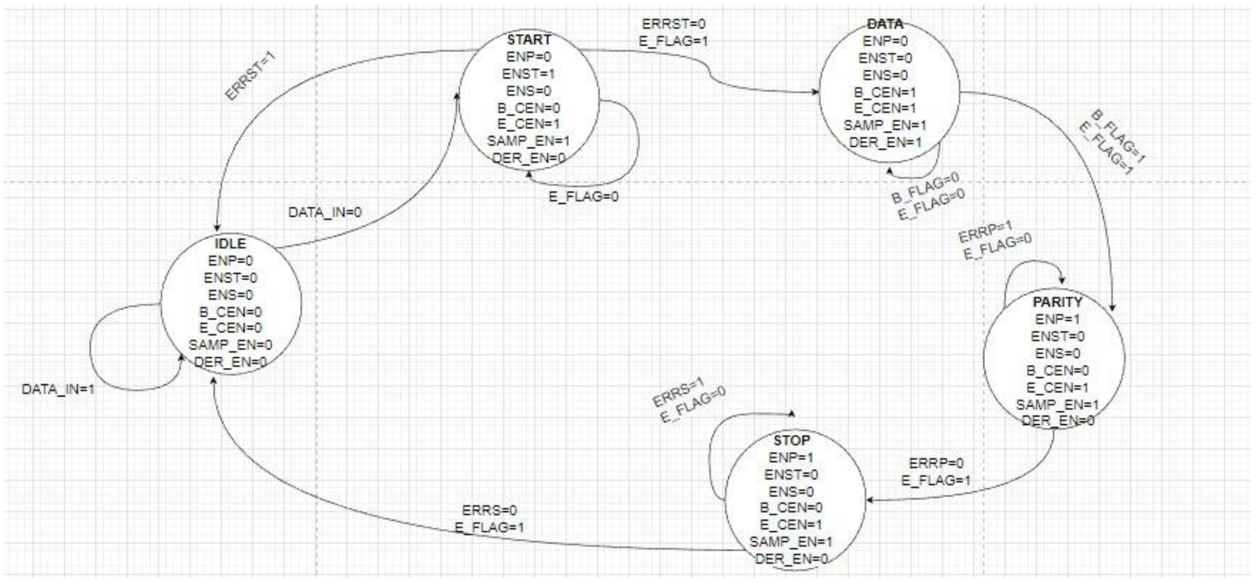

Figure 6 state diagram of the UART_RX frame finite state machine

- Asynchronous FIFO

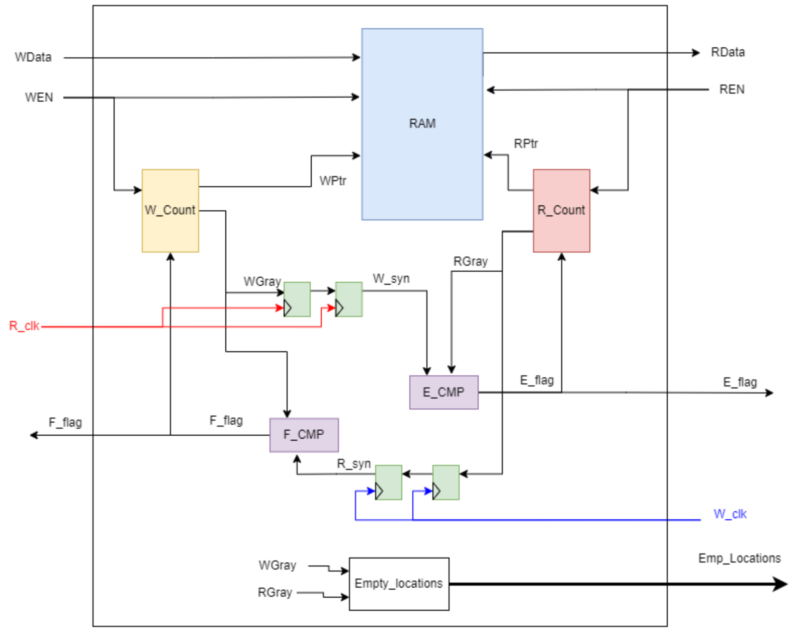

Figure 7 Asynchronous FIFO architecture

- The asynchronous FIFO is used to handle CDC issue resulting due to travelling the data from the clock domain of the UART and the clock domain of the APB and vice versa .
- It synchronizes the WPtr to the read clock domain to detect the empty flag.
- It synchronizes the RPtr to the write clock domain to detect the full flag.
- When the full flag asserted the FIFO doesn’t accept any more data.

- We can calculate the number of empty locations using the Empty_locations logic.

- APB MASTER and SLAVE

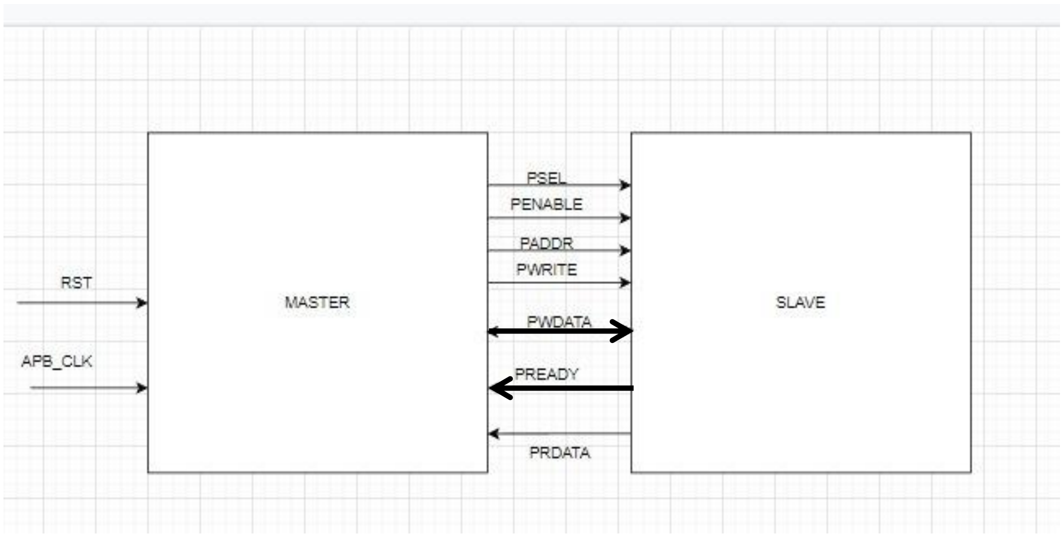

Figure 8 Architecture of the APB interface

- FSM for Master in IDLE state there is no transfer then when transfer is detected go to SETUP state and activate the slave to start reading or writing then go to access state with no condition until the transaction is done let PREADY =1

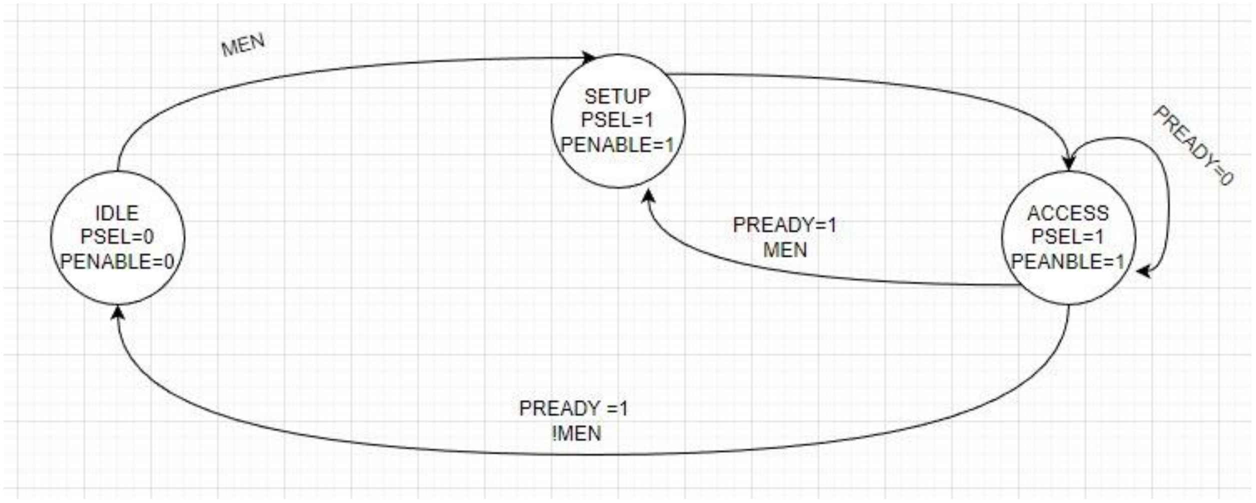

Figure 9 state diagram of the APB master finite state machine

- FSM for Slave in Setup state when PSEL=1 and PENABLE=0

Go to write or read state depends on PWRITE signal

Also depends on PADDR I can write or read in the REGFILE to make some configurations

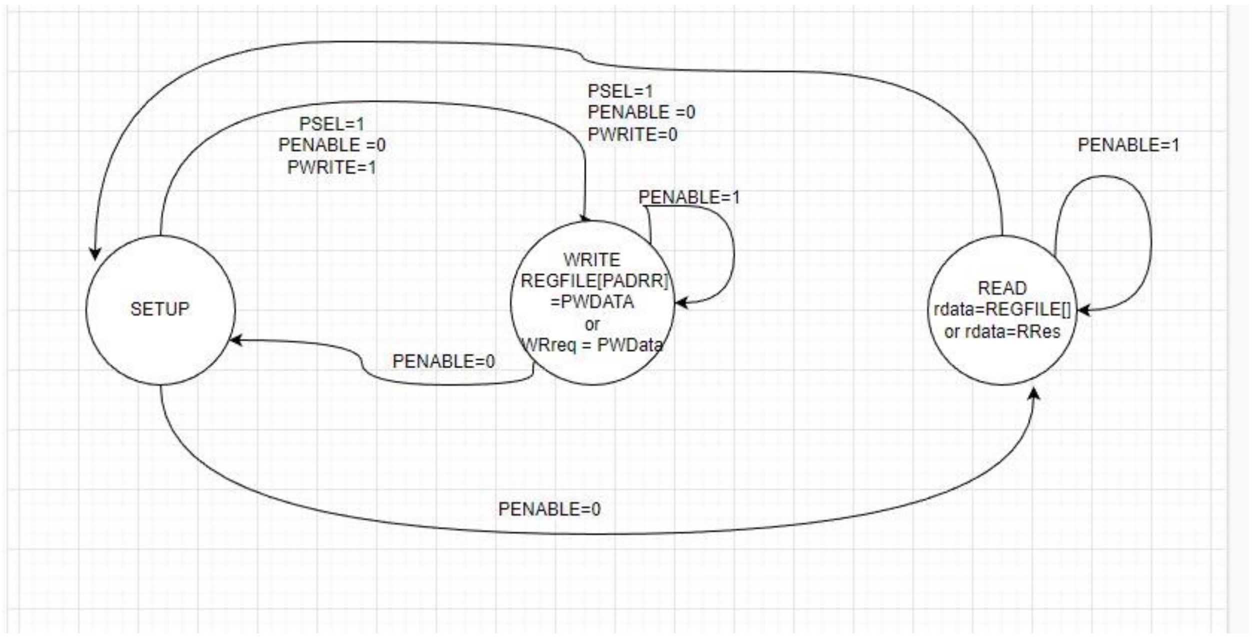

Figure 10 state diagram of the APB slave finite state machine

4.3 Timing Diagram

4.4 Verification Requirements

We will test using the loop signal to verify that the whole system working well.

- **The 1st test case is the case of we have a write request from the local MCU.**
- **The 2nd test case is the case of we have a read request from the local MCU.**
- **The 3rd test case is the case of we have 2 consecutive requests**.
- **The 4th test case is the case of we have a response after a request**.
- **The 5th test case is the case of we have a request after a response**.
- **The 6th test case is the case of we have a request and a response at the same time.**