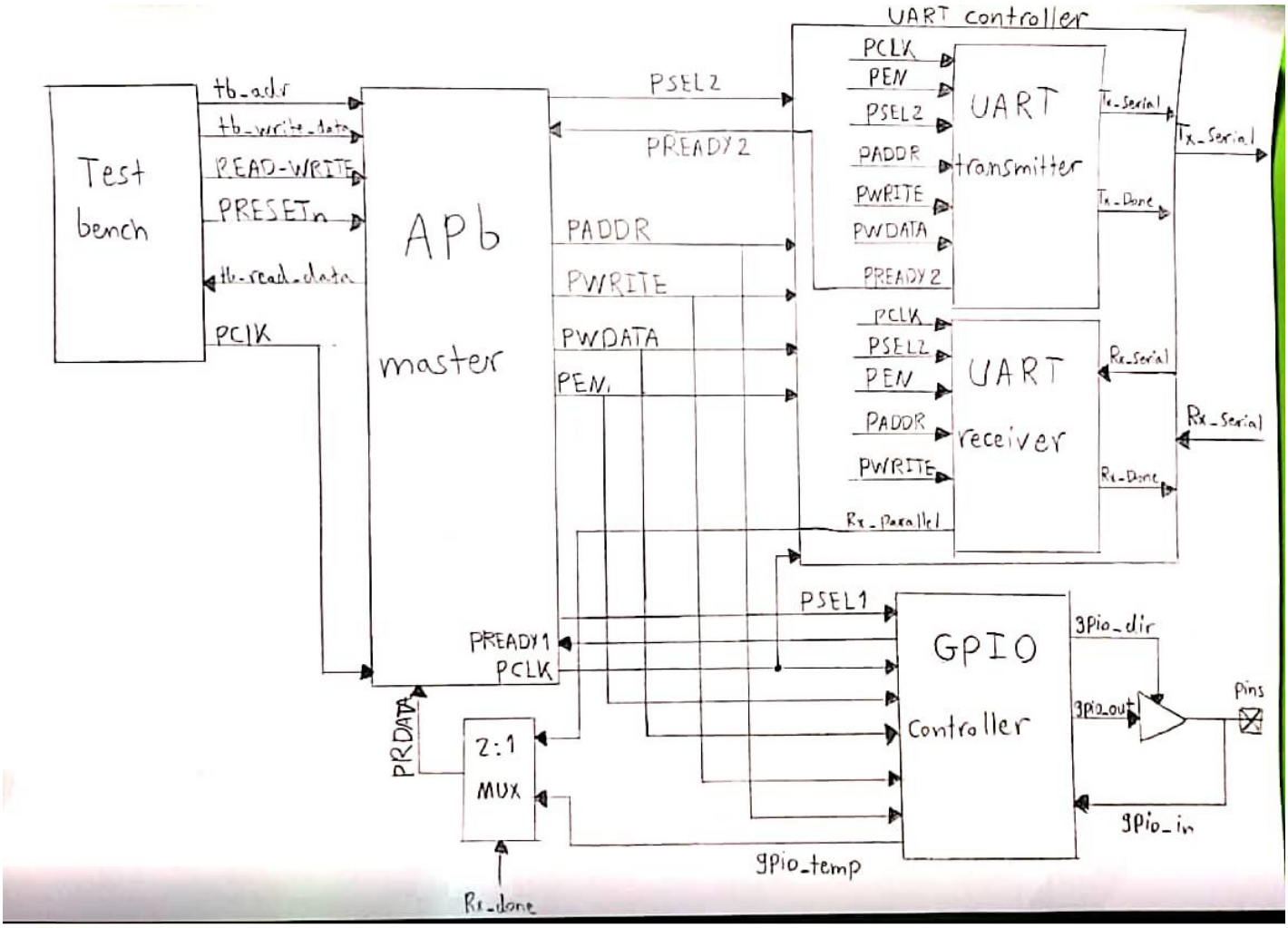
AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
CSE311s : Computer Architecture

Computer Arch Project

Mohammed Saeed Ibrahim	1900084
Mark Ashraf William	1900156
Mohamed Ahmed Samir	1900002
Hossam El-Din Adel	1900969
Marco Sherif Magdy	1900285
Andrew Samir Gayed	1900242

Table of Contents

1) Block diagram



2) Code and video link

https://drive.google.com/drive/folders/1SMgbF2_dFvOmoqst9QJ4j0oONBl4I055?usp=share_link

3) Signals description

Signal	Description
PCLK	The clock of the system
PRESETn	If driven low, the apb master returns to the IDLE state
PSEL1	It is driven high by the apb master to indicate that the operation is directed to the GPIO controller
PSEL2	It is driven high by the apb master to indicate that the operation is directed to the UART controller
PEN	It is driven high by the apb master to indicate the start of the ACCESS state and to start transmission
PADDR	It holds the address of the desired peripheral device that the apb master wants to read or write from
PWDATA	It holds the data to be written in the destination
PRDATA	It holds the read data that was returned by the peripheral device
PWRITE	It is driven high if the master wants to write and low if it wants to read
PREADY1	It is driven high by the GPIO controller to extend an APB transfer
PREADY2	It is driven high by the UART controller to extend an APB transfer
gpio_out	This register holds the values of the GPIO output pins, and the master updates it if it is required to write on the output pins

gpio_in	This register holds the values of the GPIO input pins, and the master reads from it if it is required to read the values on the input pins
gpio_dir	This is the control register of the GPIO pins, where high means the pin is output and low means the pin is input
gpio_temp	It holds the read data that was returned by the GPIO controller
Rx_parallel	This register holds the parallel data after it has been received serially by the UART receiver
Rx_done	It is driven high by the UART controller to indicate the receiving operation is finished
Rx_serial	It contains the input serial data to the receiver
Tx_serial	It holds the data after it has been serialized by the transmitter
Tx_done	It is driven high by the UART controller to indicate the transmitting operation is finished
tb_paddr	It contains the address assigned by the testbench to be sent to the apb master
tb_write_data	It is updated by the data to be written by the testbench
tb_read_data	It holds the read data that was returned by the apb master
READ_WRITE	High for read and low for write and driven by the testbench

4) Modules description

Apb_master:

It is a module made to act as the apb master bus that connects all the slaves together, it takes the address and the operation from the testbench and transfer them to the apb bus signals.

It has 3 states IDLE, SETUP and ACCESS, in the IDLE state the bus remains idle and both PSELx are low, when one of them is driven high the bus enters the SETUP state and puts the data and address on the bus, then it enters the ACCESS when PEN is driven high to indicate start of transmission.

GPIO:

It is the GPIO controller module which has 3 main registers (gpio_dir,gpio_out,gpio_in) where the first two are output and the third is input.

gpio_dir is the control register of the pins where high means output and low means input.

gpio_out is the data register of the output pins where we write on it the data we want on the output pins.

gpio_in is the data register of the input pins where any input data comes from the input pins it is stored inside it.

Each register has its own address that the master sends to the gpio controller followed by a read or write in any of them

uart_transmitter:

It is a module acts as the UART transmitter that takes parallel data and serializes it, it returns the serial data in Tx_serial and takes the input data from PWDATA from the bus, the frequency of the transmitter is less than that of the system, it works at the baud rate which is defined by a parameter (C) where $C = \frac{freq\ of\ system}{Baud\ rate}$.

It has 4 states IDLE, START_BIT, DATA_BITS and STOP_BIT.

In IDLE state the Tx_serial is driven high by default, and it enters the START_BIT state when PEN is high, and its address is the address on the bus.

In START_BIT state it transmits a low signal which acts as the start bit then it enters DATA_BITS after a clock cycle.

In DATA_BITS state it sends the data bits bit by bit on the Tx_serial.

In STOP_BIT state it sends a high signal to indicate the stop bit and the end of the operation and drives Tx_done signal high.

uart_receiver:

It is a module acts as the UART receiver that takes serial data as input and returns it parallel, it returns the parallel data in Rx_parallell and takes the input data from the testbench, the frequency of the receiver is less than that of the system, it works at the baud rate which is defined by a parameter (C) where $C = \frac{freq\ of\ system}{Baud\ rate}$.

It has 4 states IDLE, START_BIT, DATA_BITS and STOP_BIT.

In IDLE state the receiver waits until a read operation by the bus is required and PEN is high, then it enters the next state.

In START_BIT, when the receiver finds a low signal on the serial wire, it takes it as the start bit and then go to the DATA_BITS state.

In DATA_BITS state, it receives the bits bit by bit and updates the Rx_parallel with the data at the end.

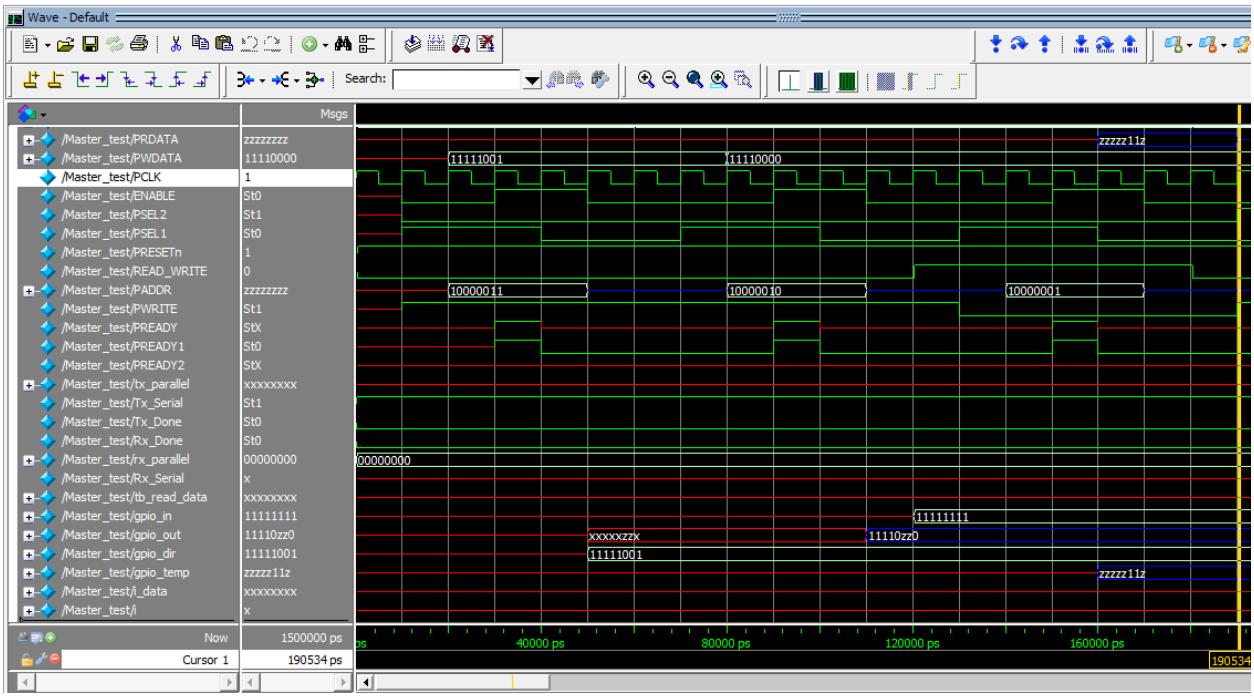
In STOP_BIT state, it detects a stop bit on the serial wire and drive Rx_done high.

Note:

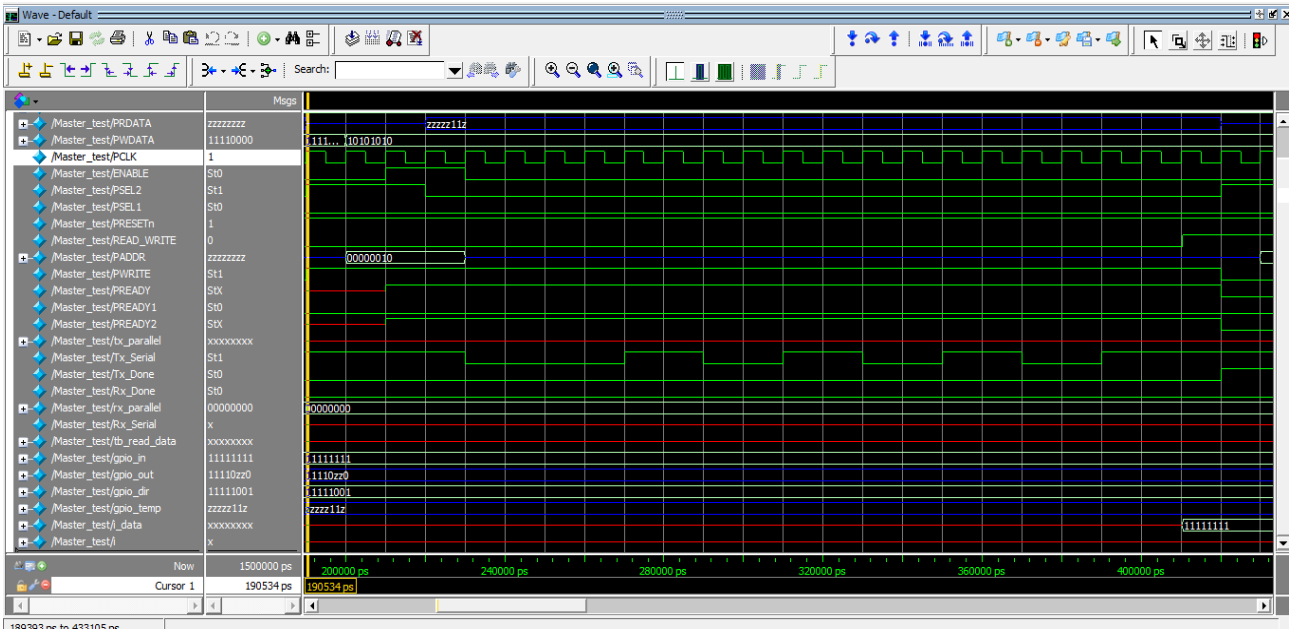
The PRDATA register is connected to the gpio_temp(read output of the gpio) and Rx_parallel(read output of the receiver) by a 2 to 1 mux and the select signal is Rx_done.

5) Snapshots

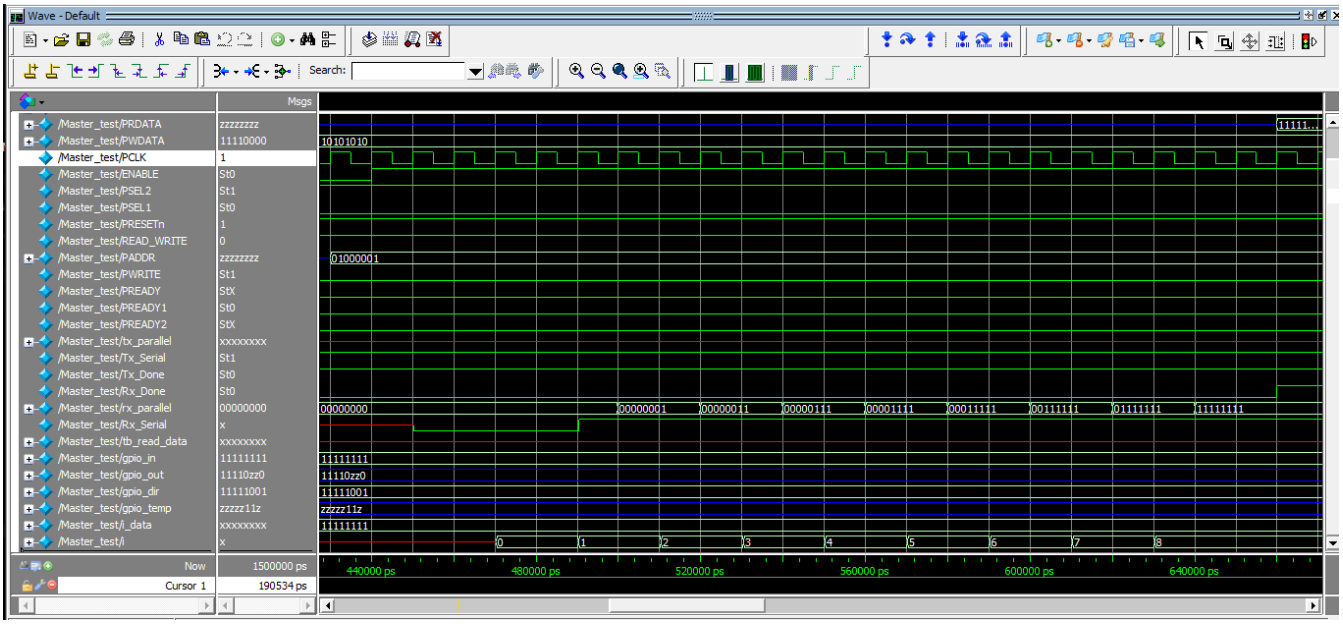
Gpio test:



Transmitter test:



Receiver test:

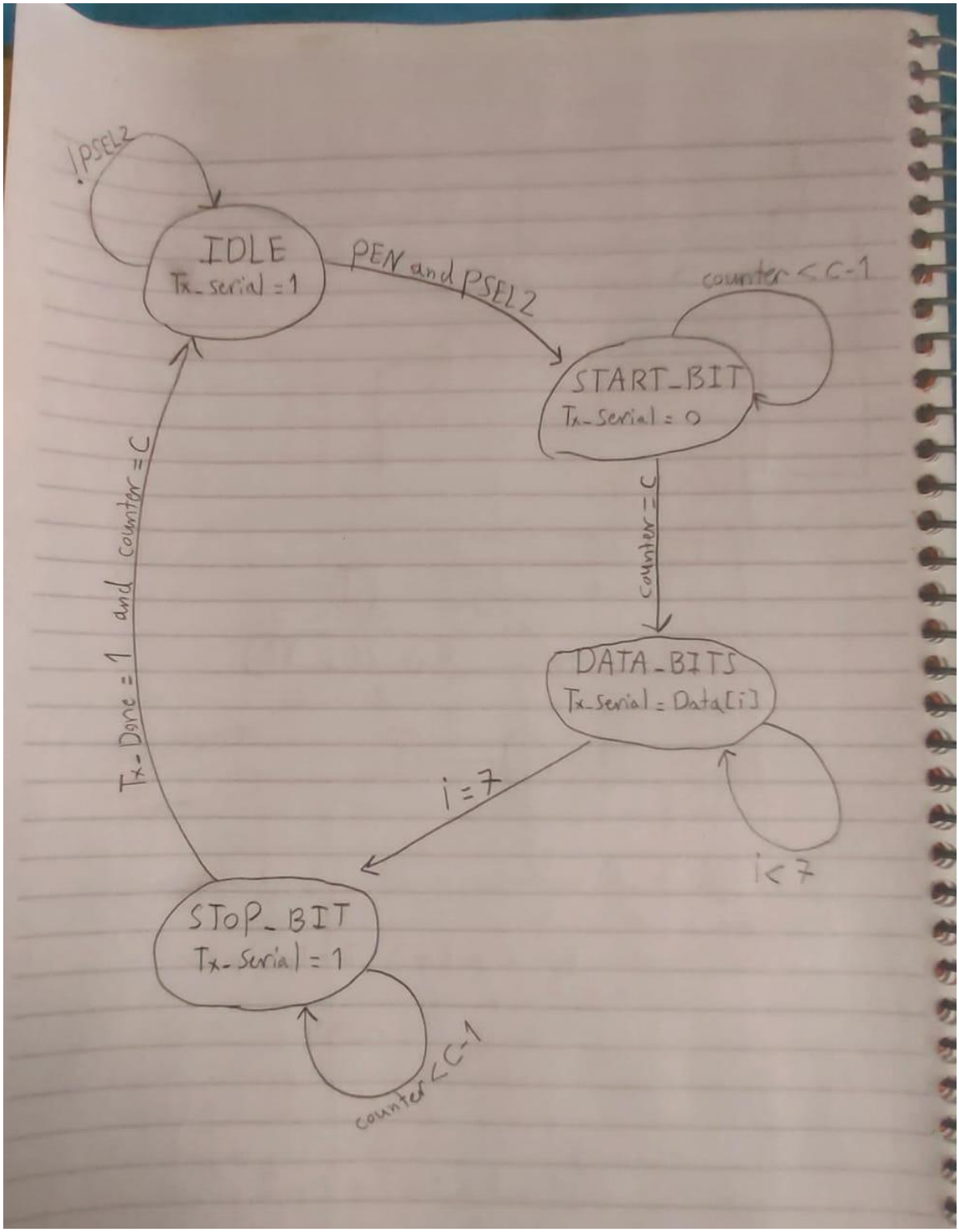


6) Contribution table

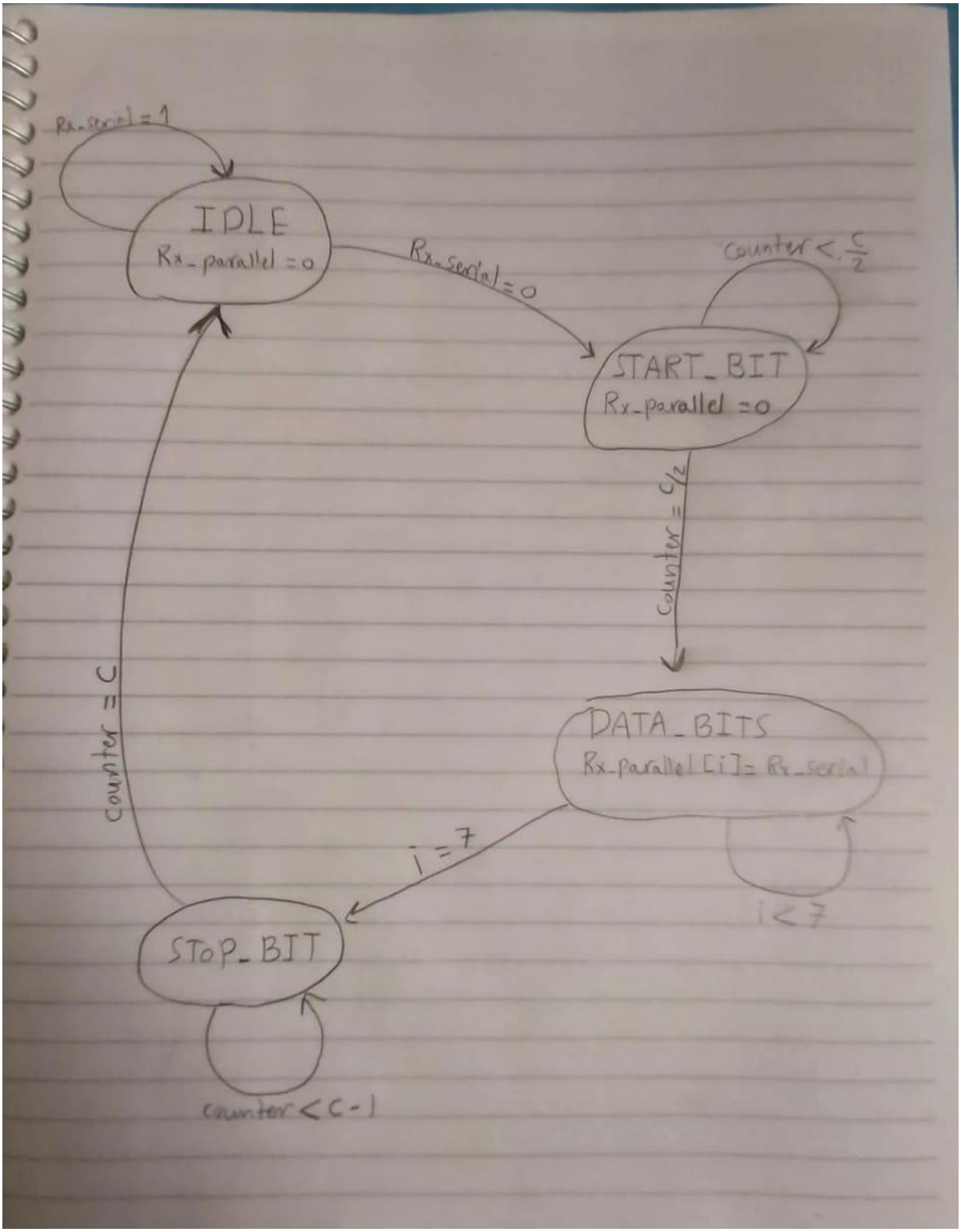
Team member	ID	Contribution
Mohamed ahmed samir	1900002	Master testbench, block diagram and video
Hossam eldin adel talaat	1900969	Apb_master code
Mark ashraf william	1900156	GPIO code
Andrew samir kamel gayed	1900242	Uart transmitter code
Marco sherif magdy	1900285	Uart receiver code
Mohamed saeed ibrahim	1900084	Uart, additional testbenches (in the link) and the documentation

7) Finite state machines

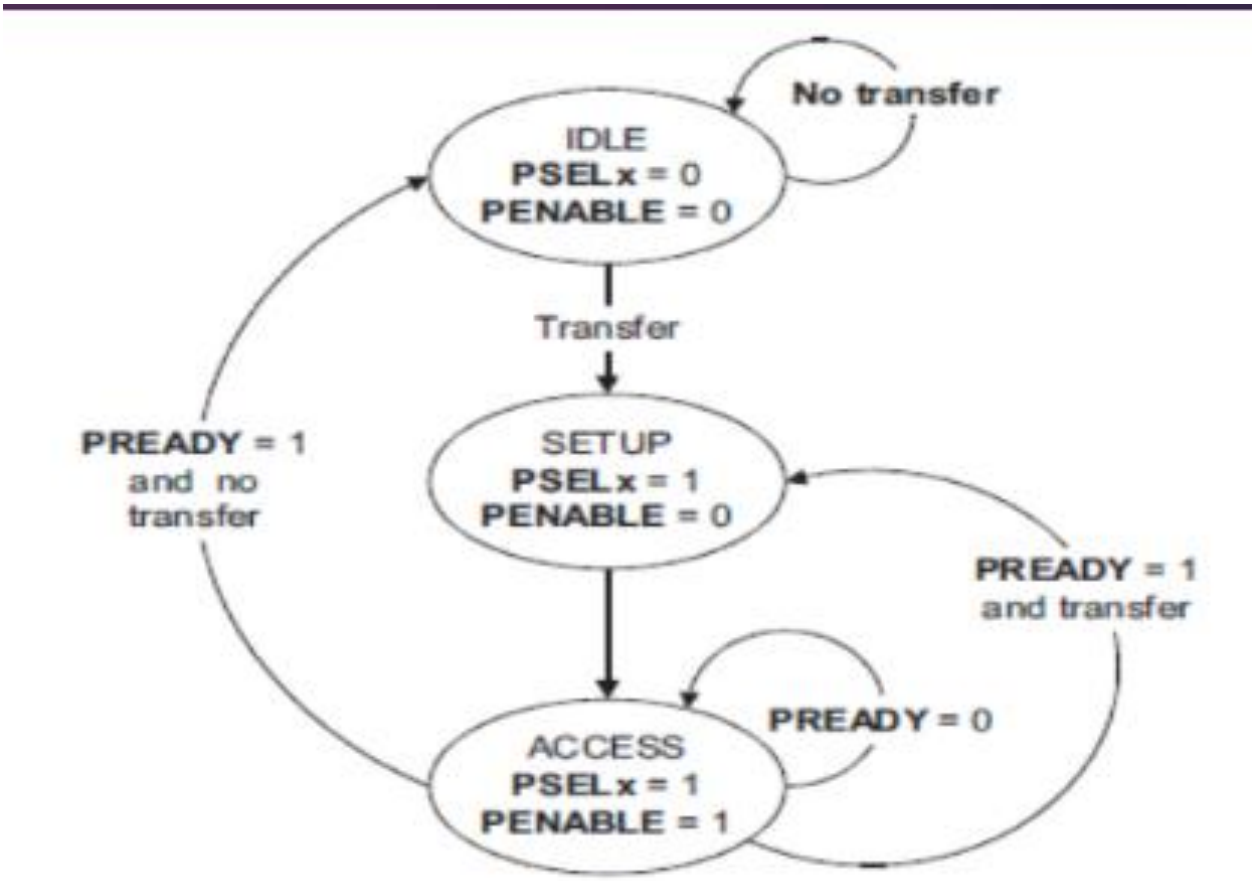
Uart transmitter:



Uart receiver:



Apb_master:


No transfer
IDLE
PSELx = 0
PENABLE = 0
Transfer
SETUP
PSELx = 1
PENABLE = 0
PREADY = 1
and no
transfer
PREADY = 1
and transfer
PREADY = 0
ACCESS
PSELx = 1
PENABLE = 1