Architecture Project

Name	ID
Mohamed Atef Hassan	1901326
Ahmed Emad Hassan Shafik	1900441
Karim Mohamed Hemidah Aql	1900511
Kareem Ayman Abdelaleem	1901763
Yousef Amer Awadallah Osman	1901524
Hassan Ahmed Fathy Bahnasy	1901371

Name	Contributation
Mohamed Atef Hassan	APB
Ahmed Emad Hassan Shafik	GPIO
Karim Mohamed Hemidah Aql	UART
Kareem Ayman Abdelaleem	APB
Yousef Amer Awadallah Osman	GPIO
Hassan Ahmed Fathy Bahnasy	UART

UART Module Details:

UART was implemented by: Karim Mohamed Aql 1900511

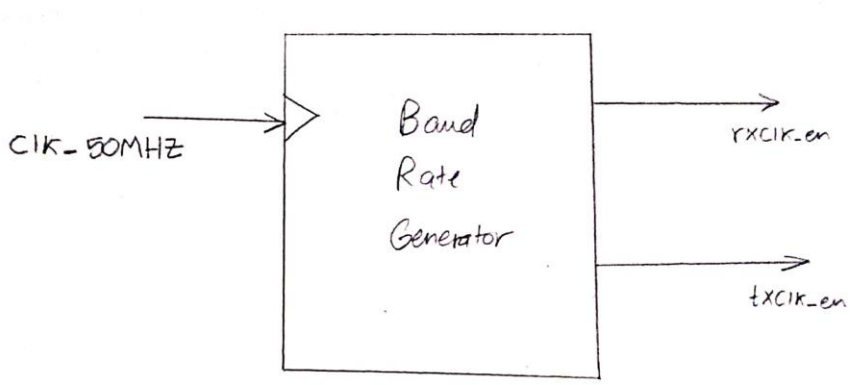
The UART has a top module that consists of 3 internal modules:

1. Baud Rate Generator
2. UART transmitter
3. UART Receiver

Detailed explanation, block diagrams, and finite state machines of each module:

(further explanation for the modules and the signal is provided in the code Verilog file)

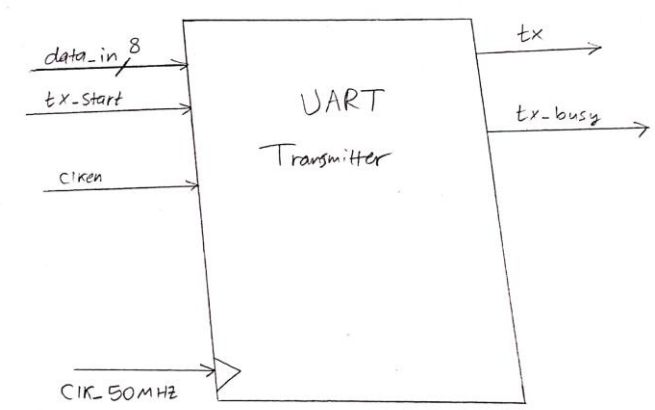
- **Baud Rate Generator:**
  This module takes a 50MHZ clock as input and generates two outputs: *txclk_en* & *rxclk_en* which are connected to *clken* ports in UART Transmitter & Receiver in the top module, *txclk_e*n is our baud rate which in this case is 115200 & *rxclk_en* is the oversampled baud rate = 16*115200



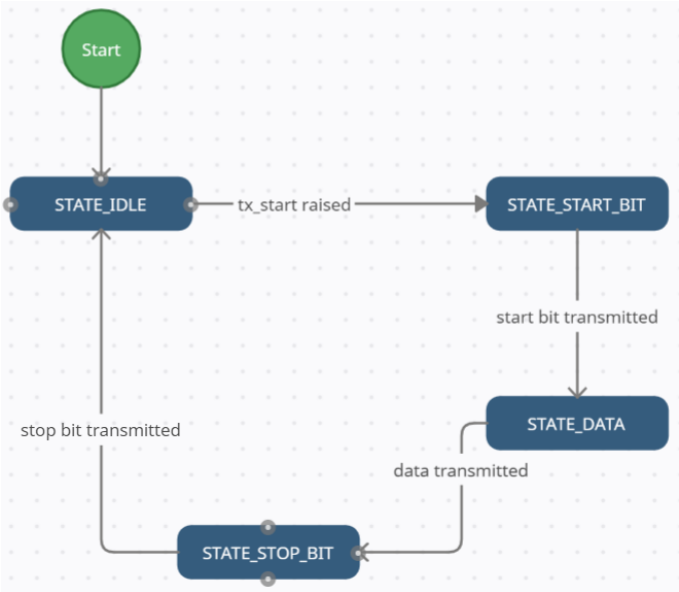
- **UART Transmitter:**

  This module has 4 states:

  1. STATE_IDLE
     Idle state waiting for start bit
  2. STATE_START_BIT
     Transmits the start bit
  3. STATE_DATA
     Loads parallel data into a shift register and transmits them serially
  4. STATE_STOP_BIT
     Transmits stop bit



Finite State Machine:



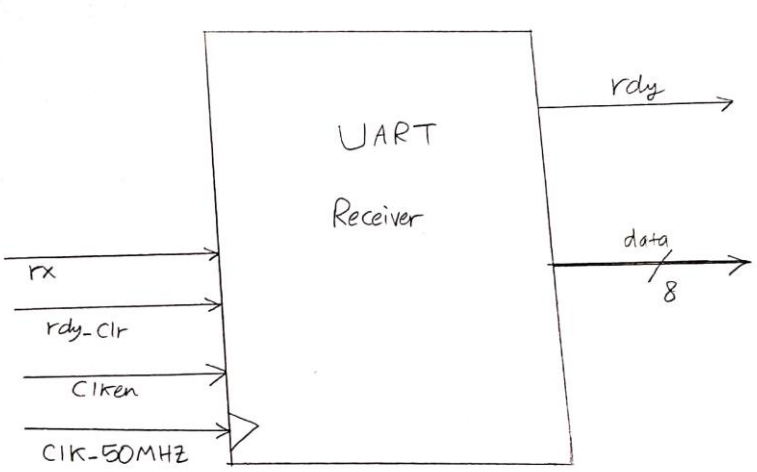
- **UART Receiver:**

  This module has 3 states:

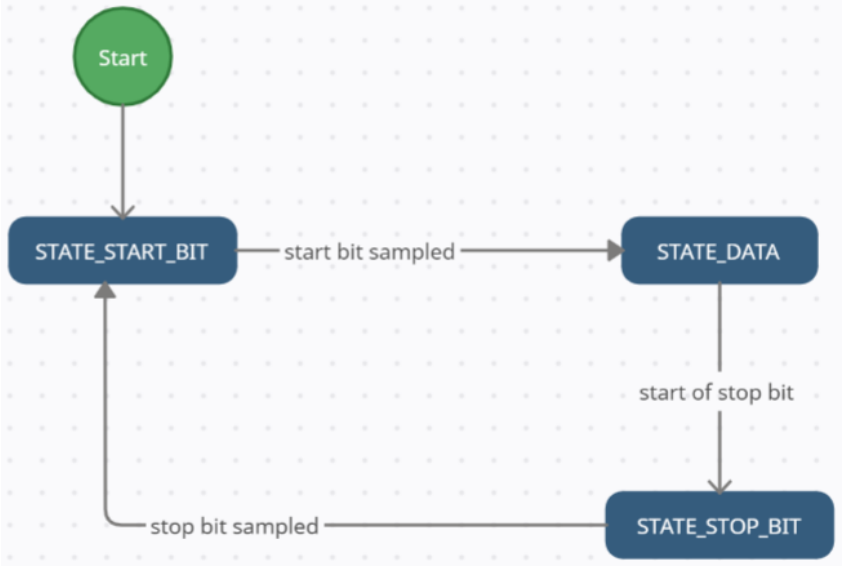
  1. STATE_START_BIT
     we will start the counter from the first time we sample a low(0),

once we have sampled a full bit, we will start collecting data bits
(we will go to STATE_DATA)

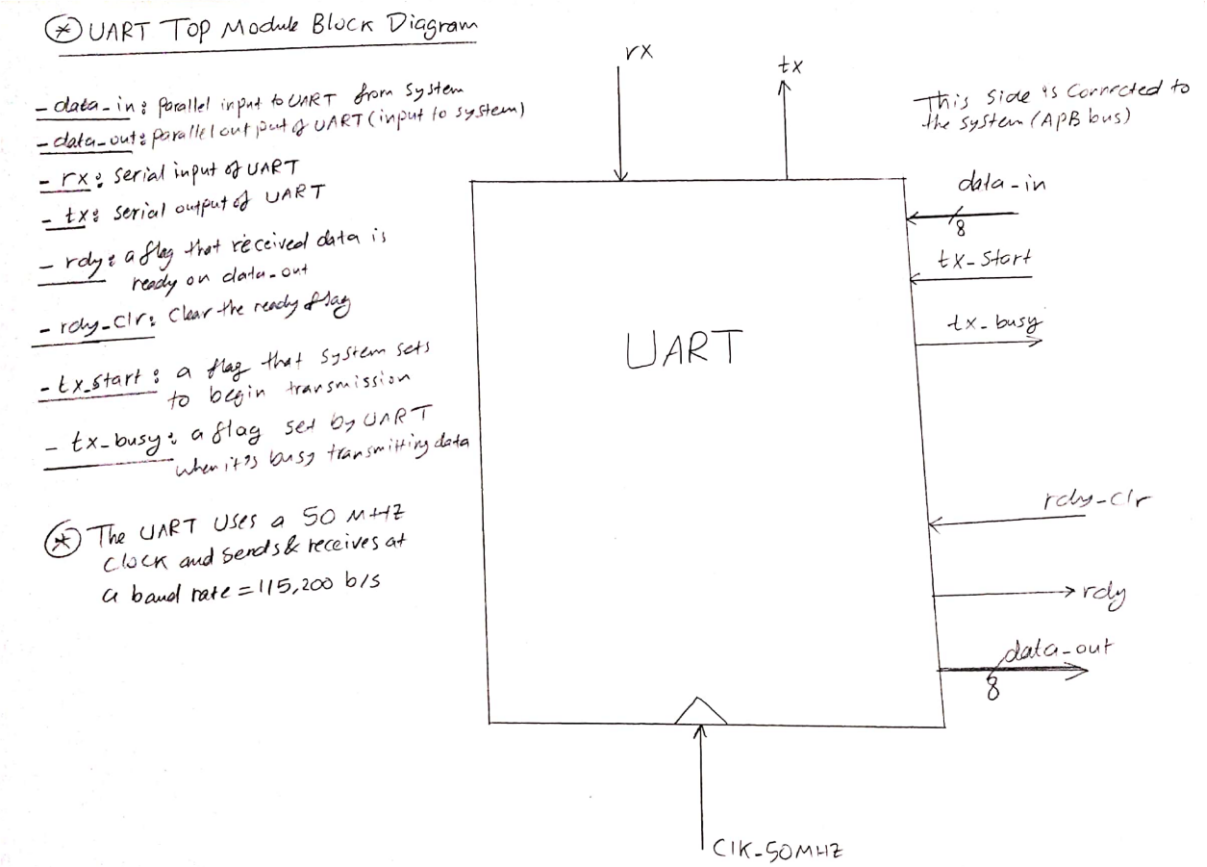
2. STATE_DATA
   Samples data and collects it in a draft register that's constantly updated
3. STATE_STOP_BIT
   Samples the stop bit



Finite State Machine:



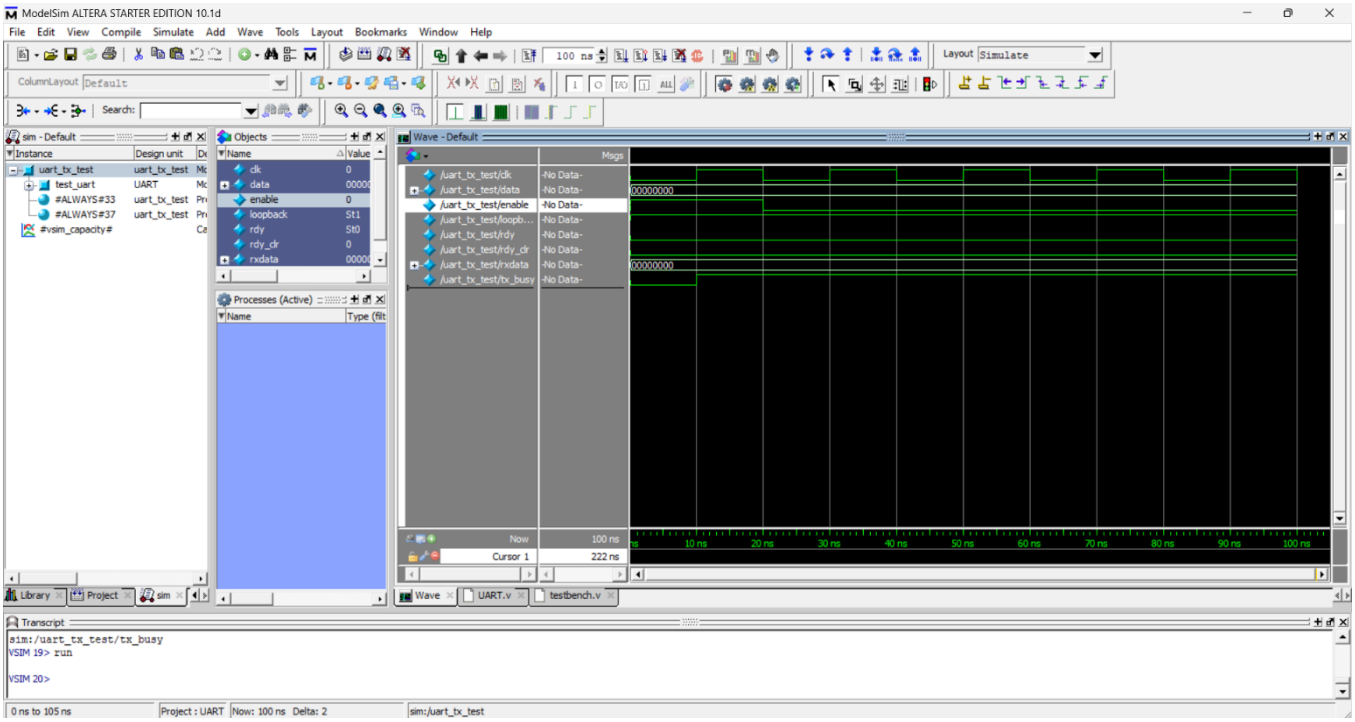
- UART (Top Module) and signals Description:



UART Testbench:

Testing strategy: looping the rx and tx pins to each other, sending data and receiving it at data_in

Simulation output:



We see that the data was successfully received .

GPIO Module Details:

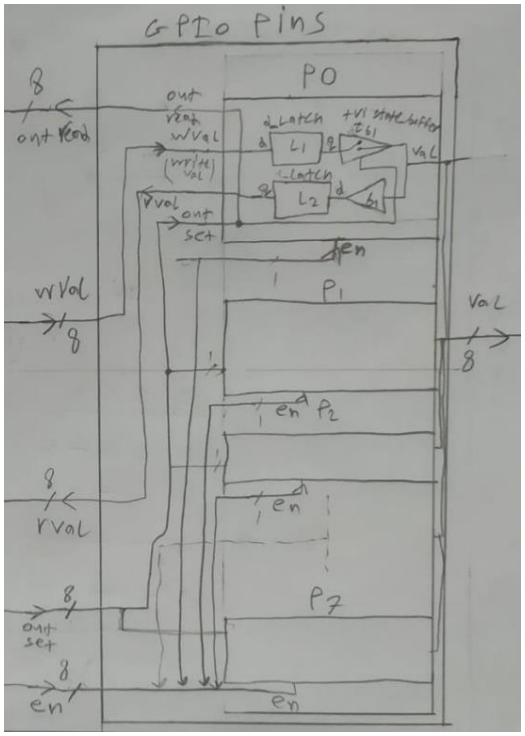
GPIO was implemented by: Ahmed Emad Hassan Shafik 1900441

The UART has a top module that consists of internal modules:

- GPIO Pins
- GPIO Interface Module
- Pin

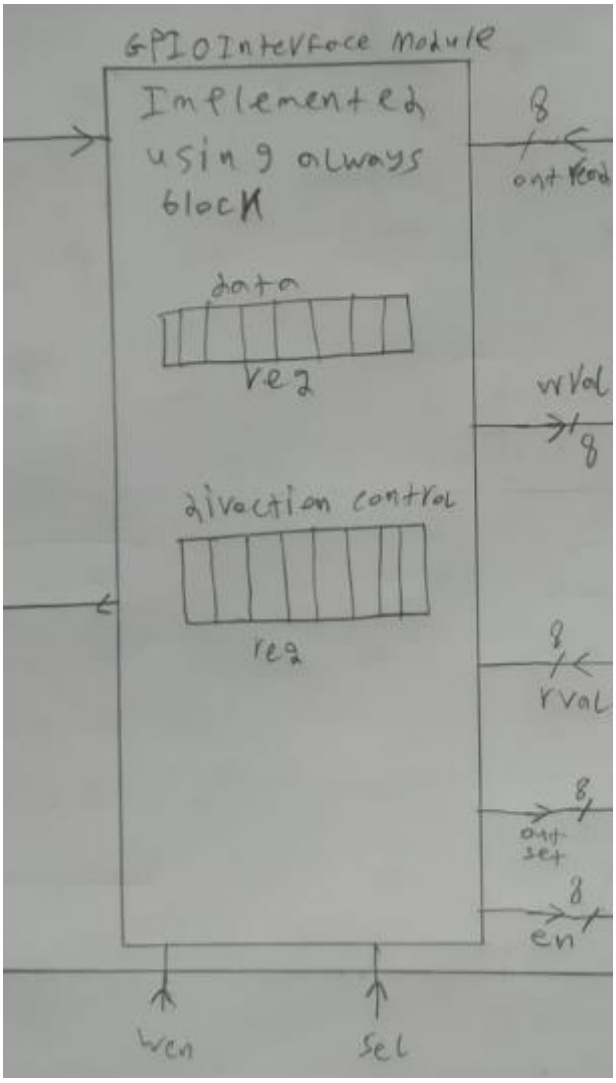
GPIO Pins

This module has 8 pins inside it and controlled by another module



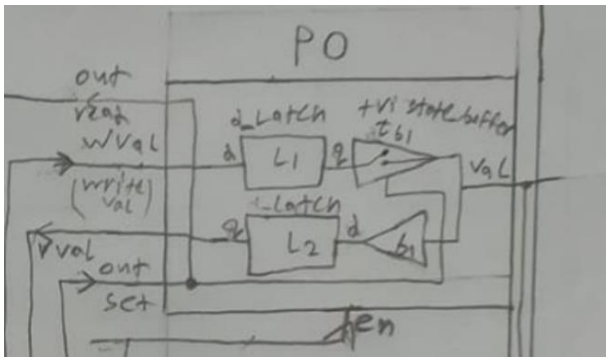
GPIO Interface Module

It sends orders io GPIO and stores registers



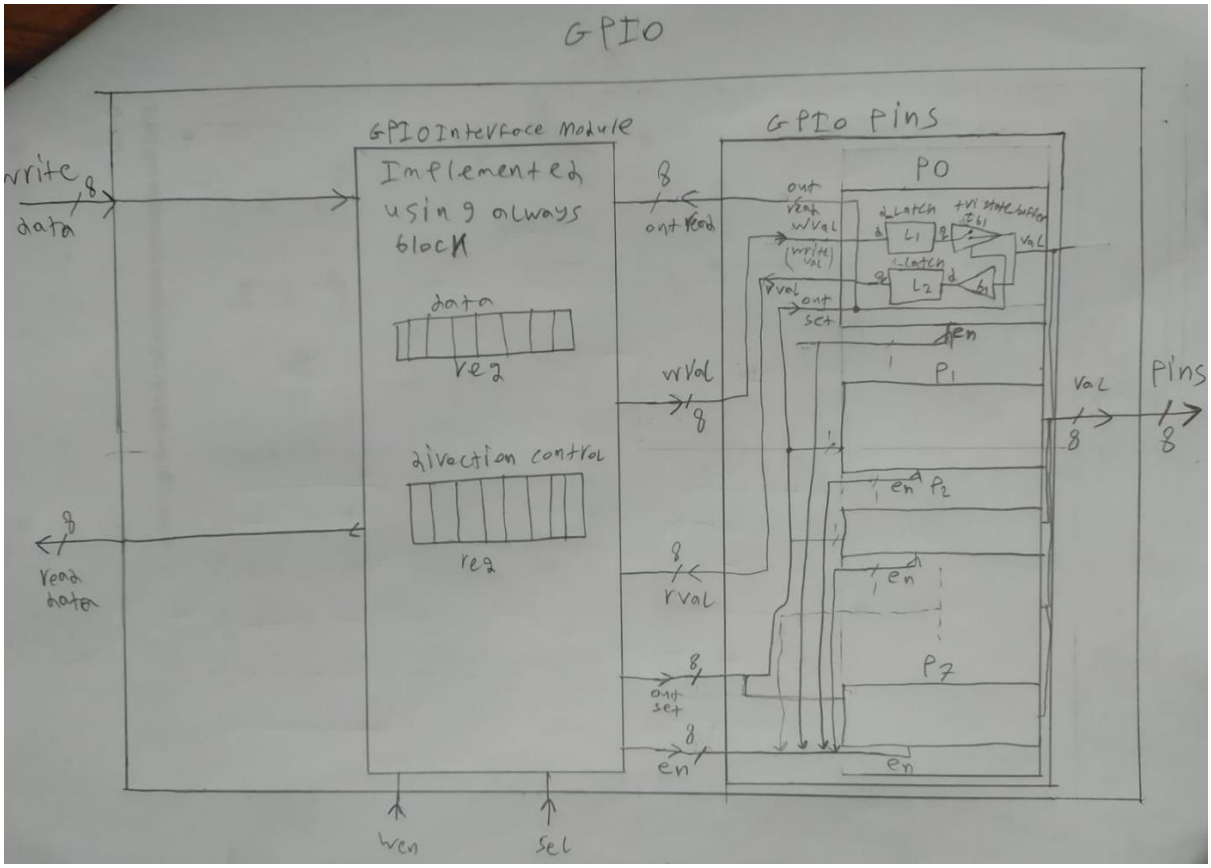
PIN

it is a single pin it contains of tristate buffers, buffers and latches and 8 of them are included in Pins module

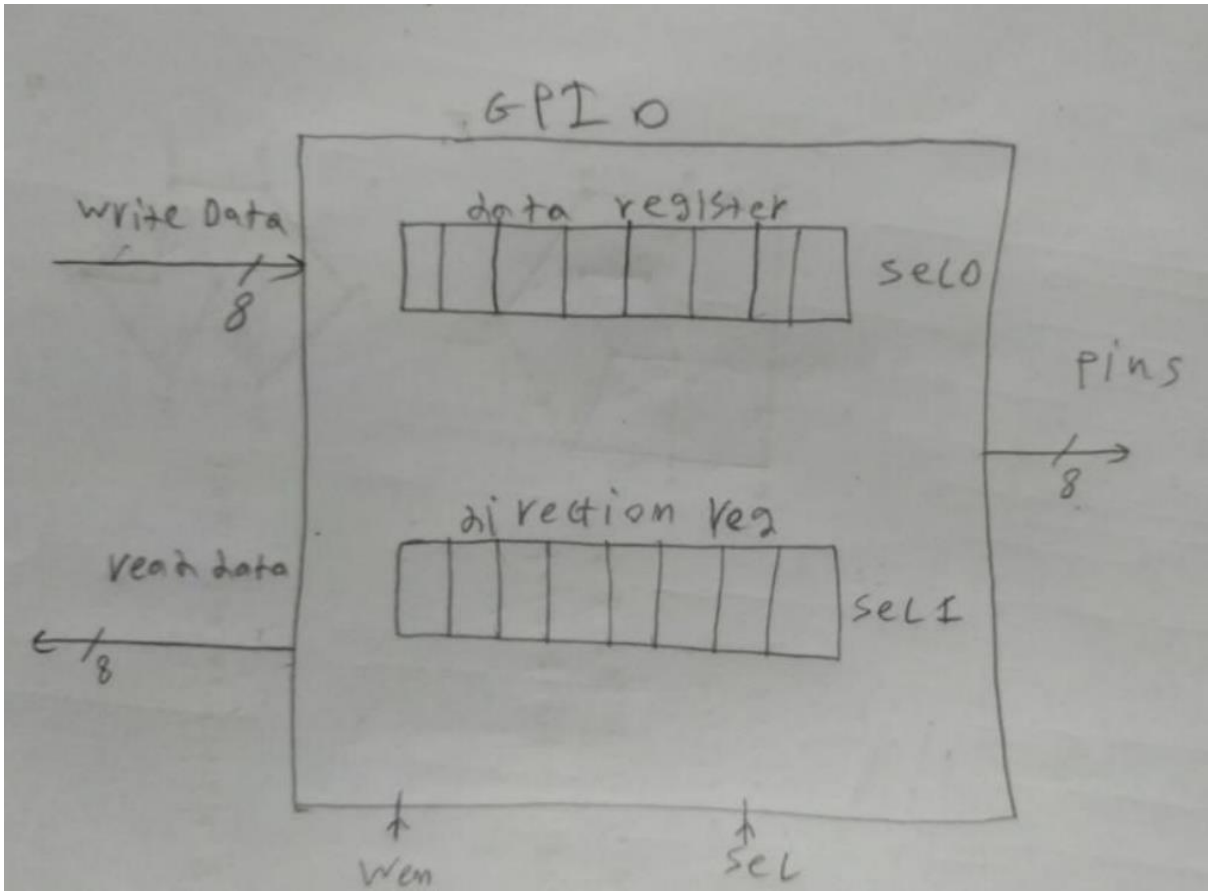


GPIO

It contains all other modules and can be controlled using APB



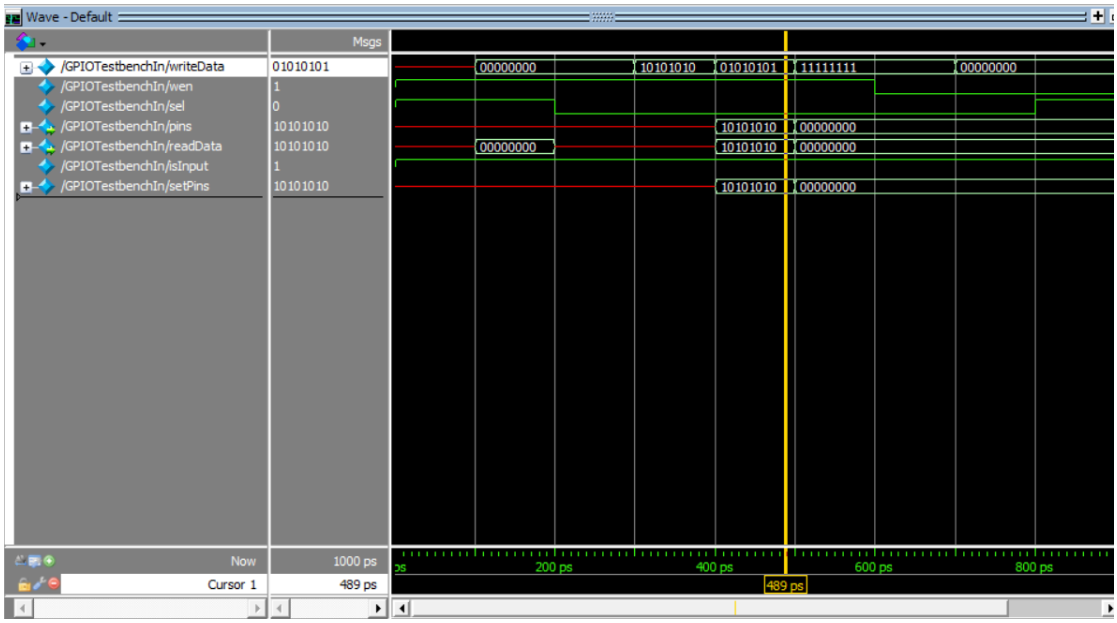
Simplified GPIO



Testing strategy

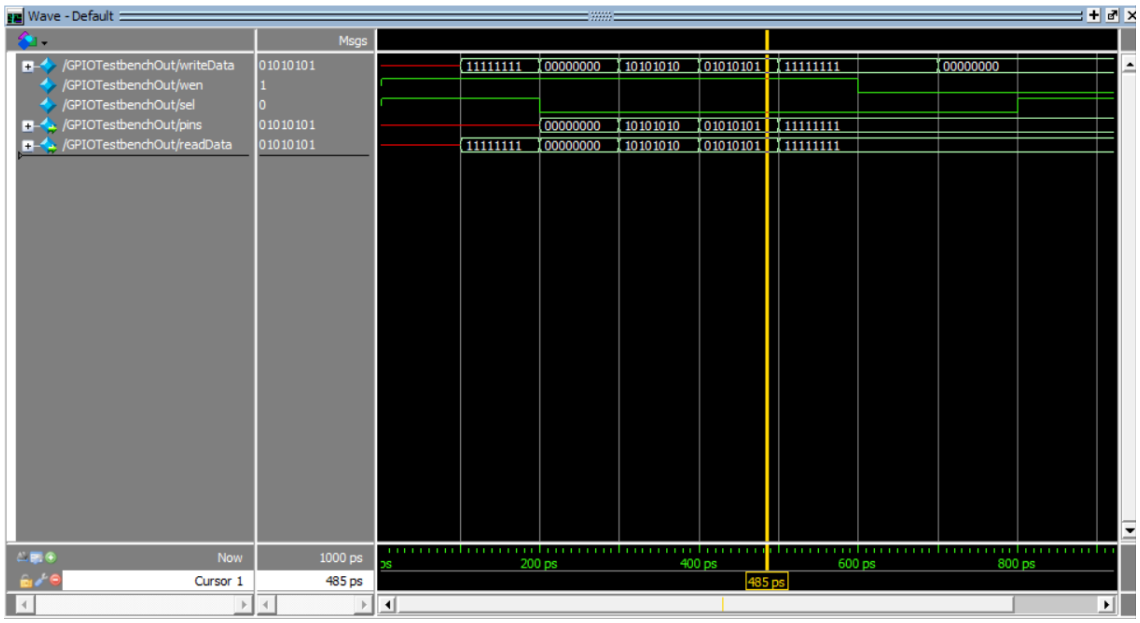
Testing the behavior of all pins using two test benches one to test pins as input pins and other to test output pins.

Input Testbench



Here we see values of pins set using forcing used by testbench using set pins values and read the value of the pins and store it in data register and read it using readData and it is independent of the data written using write data because of tristate buffers that cuts the connection between them when writing

Output Testbench



Here we see pins takes the values written to data register with sel = 0 when direction control register takes 11111111

With sel = 1

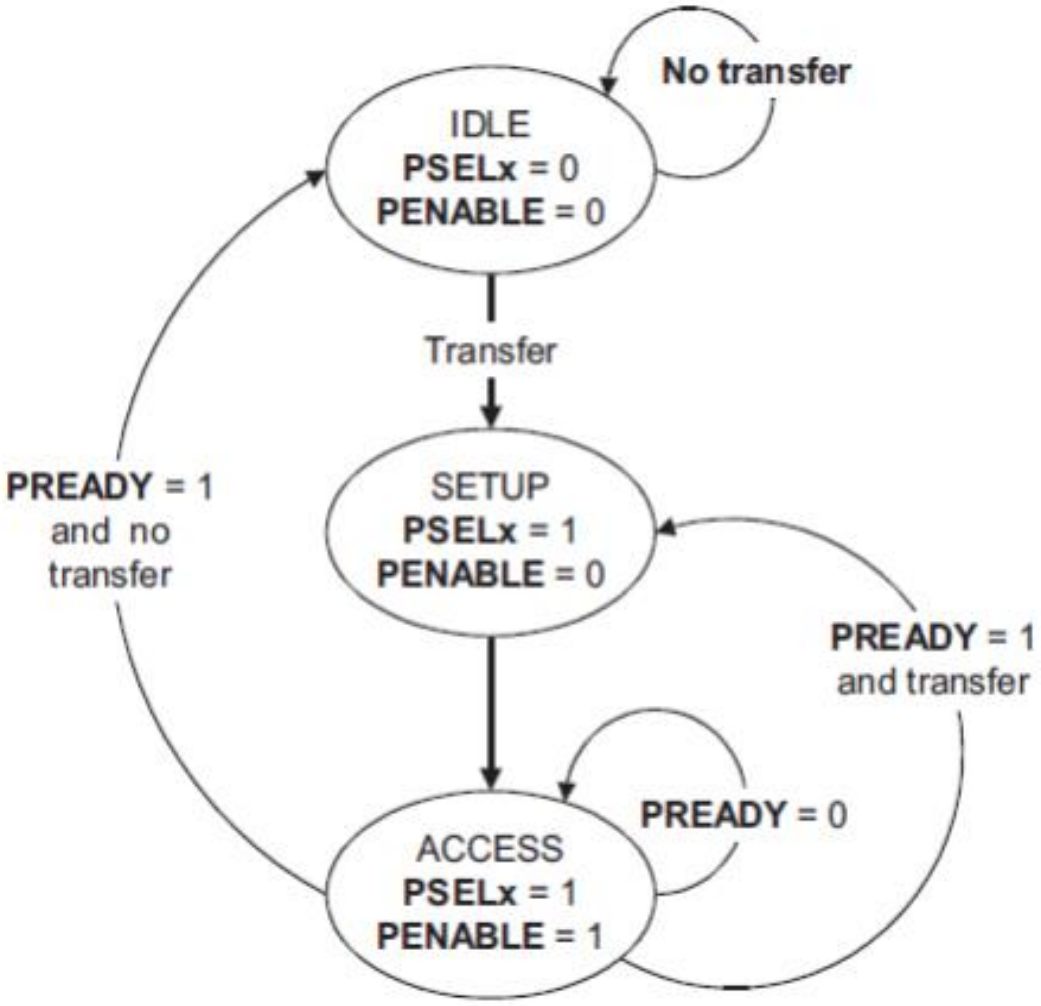
Apb

This module has 3 states:

- IDLE :
  Wait for transfer bit to be 1 to go to next state

- SETUP:
  See pwrite bit if 1 write or 0 read

- ACCESS/ENABLE
  See psel to select and set enable bit to 1



SIGNAL	SOURCE	Description	WIDTH(Bit)
Transfer	System Bus	APB enable signal. If high APB is activated else APB is disabled	1
PCLK	Clock Source	All APB functionality occurs at rising edge.	1
PRESETn	System Bus	An active low signal.	1
PADDR	APB bridge	The APB address bus can be up to 32 bits.	8
PSEL1	APB bridge	There is a PSEL for each slave. It's an active high signal.	1
PENABLE	APB bridge	It indicates the 2^{nd} cycle of a data transfer. It's an active high signal.	1
PWRITE	APB bridge	Indicates the data transfer direction. PWRITE=1 indicates APB write access(Master to slave) PWRITE=0 indicates APB read access(Slave to master)	1
PREADY	Slave Interface	This is an input from Slave. It is used to enter access state.	1
PSLVERR	Slave Interface	This indicates a transfer failure by the slave.	1
PRDATA	Slave Interface	Read Data. The selected slave drives this bus during read operation	8
PWDATA	Slave Interface	Write data. This bus is driven by the peripheral bus bridge unit during write cycles when PWRITE is high.	8

1) Parallel bus operation. All the data will be captured at rising edge clock.

2) Two slave design.

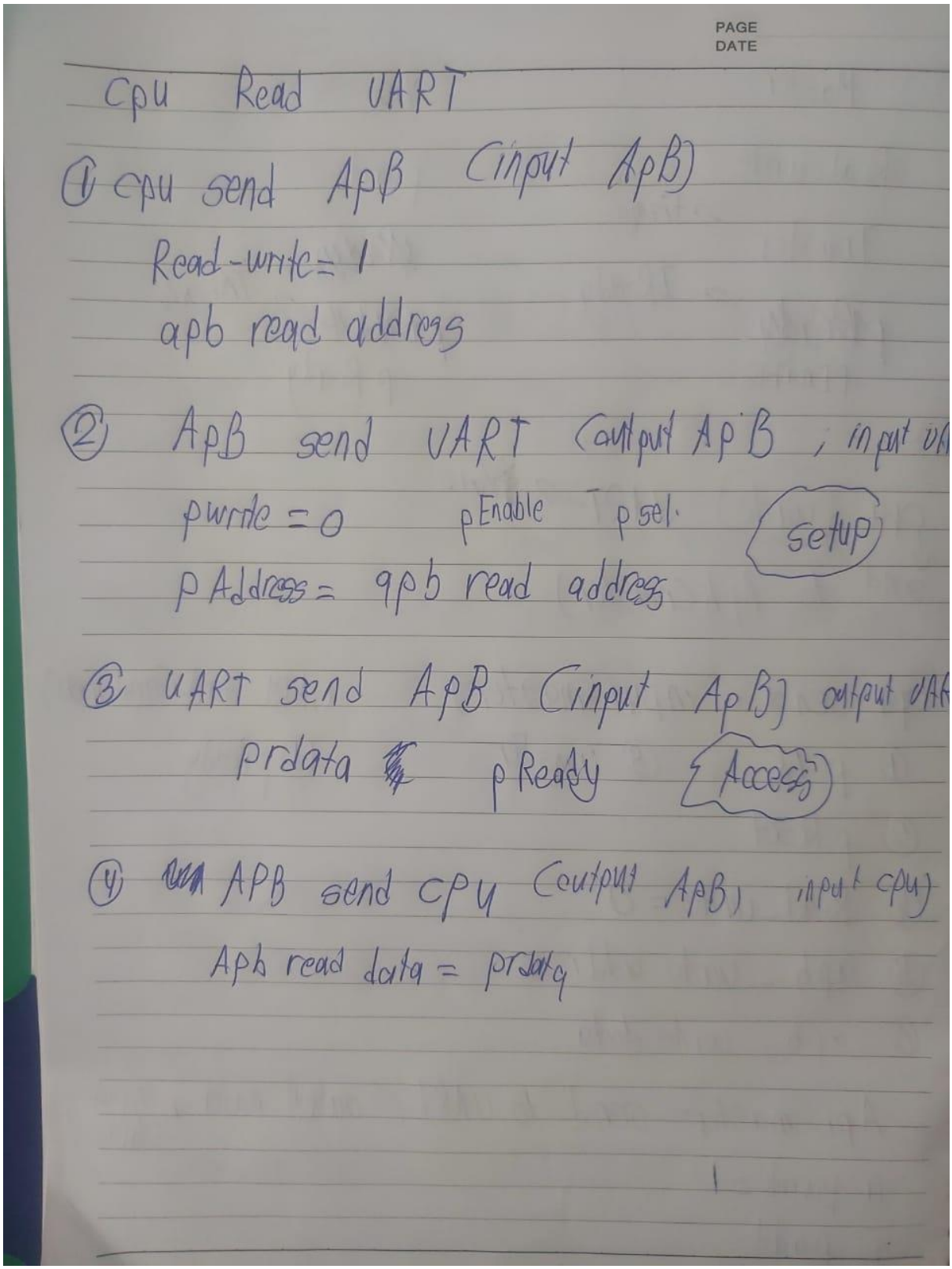
3) Signal priority: 1.PRESET (active low) 2. PSEL (active high) 3. PENABLE (active high) 4. PREADY (active high) 5. PWRITE

4) Data width 8 bit and address width 9 bit.

5) PWRITE=1 indicates write PWDATA to slave. PWRITE=0 indicates read PRDATA from slave.

6) Start of data transmission is indicated when PENABLE changes from low to high. End of transmission is indicated by PREADY changes from high to l

Apb example explain



Link for videos

GPIO video

https://drive.google.com/file/d/1yvt9eSi9vhW6MugaKCOrYc86DUB8MrUH/view?usp=sharing

Uart video

https://drive.google.com/file/d/1jlDTbJvRMeks49H1-g1JsPSj0xwtyqYQ/view?usp=share_link

GITHUB LINK

https://github.com/kareem62/computer-architecture-project.git