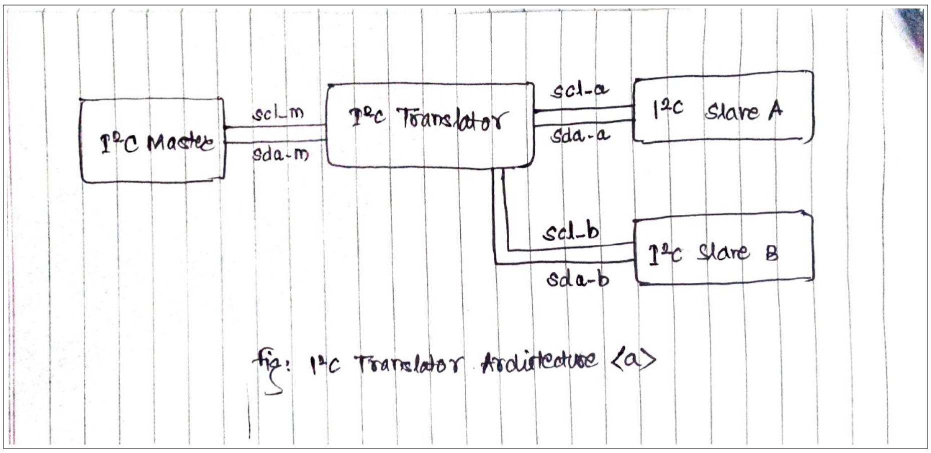
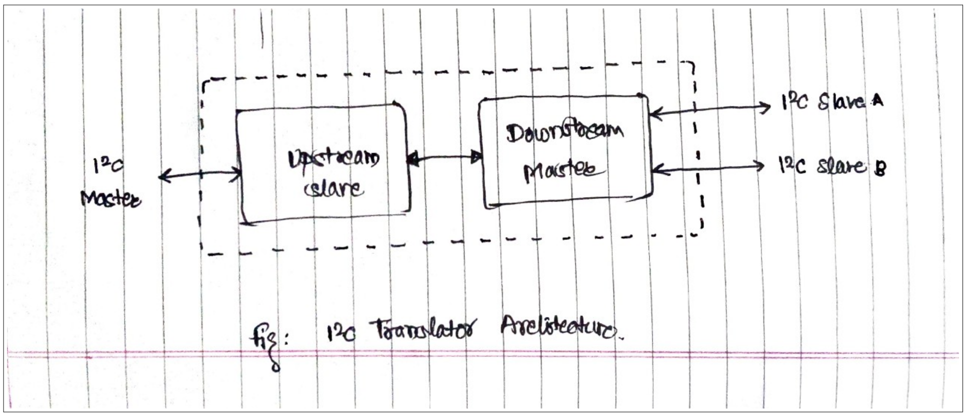
I2C Translator

- ARCHITECTURE OVERVIEW

Aim was to design an FPGA-based I2C address translator to allow a single device's i2c address to be dynamically remapped, enabling it to coexist with other devices that may have the same device address.



The transalator needs the capability to identify the virtual address send by i2c master in its transaction, mappin the virtual address of the slave to the desired i2c bus of the slave keeping rest slave lines in idle and perform the transactions seamlessly.



For receiving the I2C Transaction Data from the I2C Master theres the need of the I2C Slave (here, Upstream Slave) whereas, for sending the received byte to the desired slave, theres need of the I2C Master (here, Downstream Master)

Hence the design consists of a top module that consists of two modules viz.,

a) Upstream Slave
b) Downstream Master

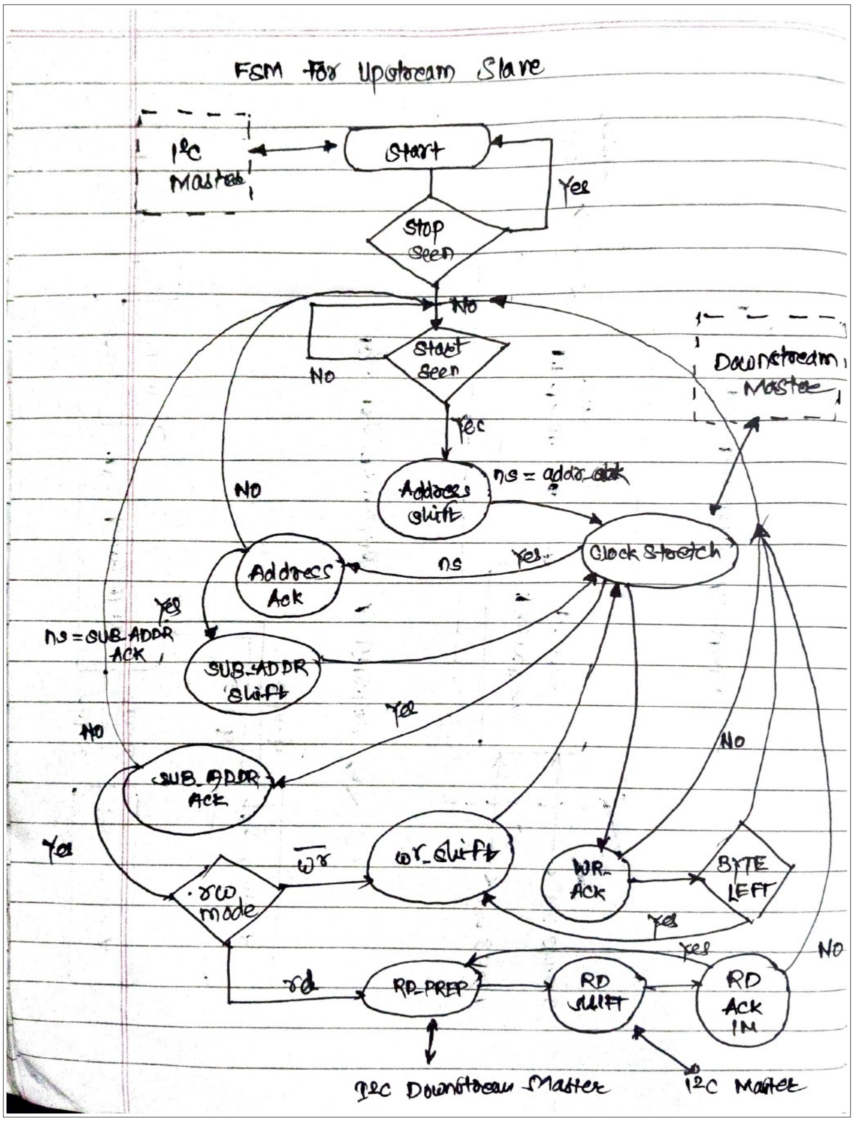
- FSM/LOGIC EXPLANATION

As mentioned in the architecture, there are two modules, both are having separated Finite State Machine both are working integratedly for completion of the i2c transaction initiated by the master.

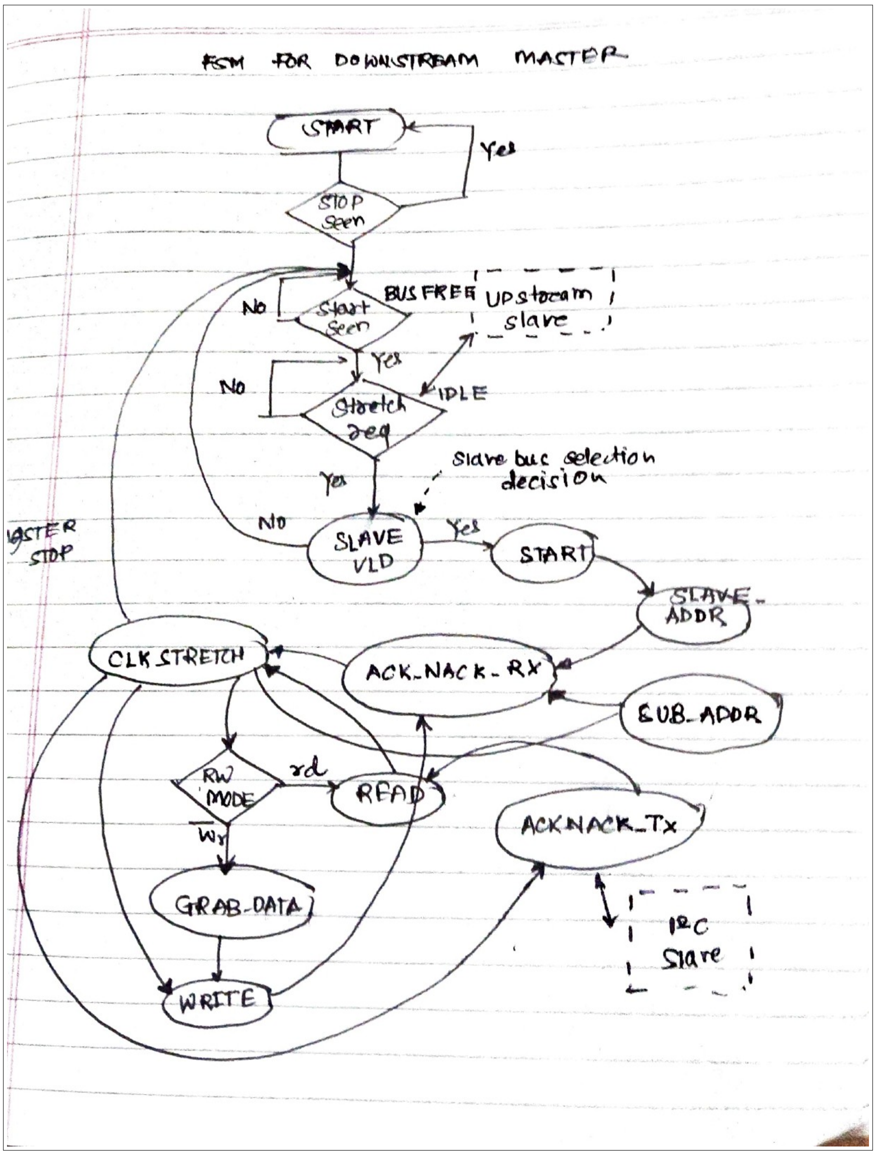
In I2C Standard, the i2c master is the main orchestrator behind the transaction. This gives master the ability to start and stop the transaction as and when required by it. So hearing for the transaction initiation and termination is a must in the design of the i2c translator.

By reading the lines of the i2c master for having proper detection of the START & STOP of the transaction, the logic to reset the states of the state machines of both the modules is implemented.

a. FSM Design for Upstream Slave



b. FSM Design for Downstream Master



➢ IMPLEMENTATION OF ADDRESS TRANSLATION

As I2C Master is going to send the virtual address for the slave, each virtual address is mapped to unique I2C Slave bus for transaction.

The decision to choose the desired slave lines for operation is performed in the DOWNSTREAM MASTER module after receiving the first byte received by the UPSTREAM SLAVE.

Depending upon the virtual address, there is case statement based logic separated each for each I2C Slave Bus.

➢ DESIGN CHALLENGES FACED

1. Designing Clock Stretching Compatible I2C Master and Slave.
2. States selection for the Upstream Slave and Downstream Slave.
3. State Machine Integration of Upstream Slave and Master for proper handshaking between MASTER & SLAVE relying completely on the acknowledgement from respected device.