I2C address translator module

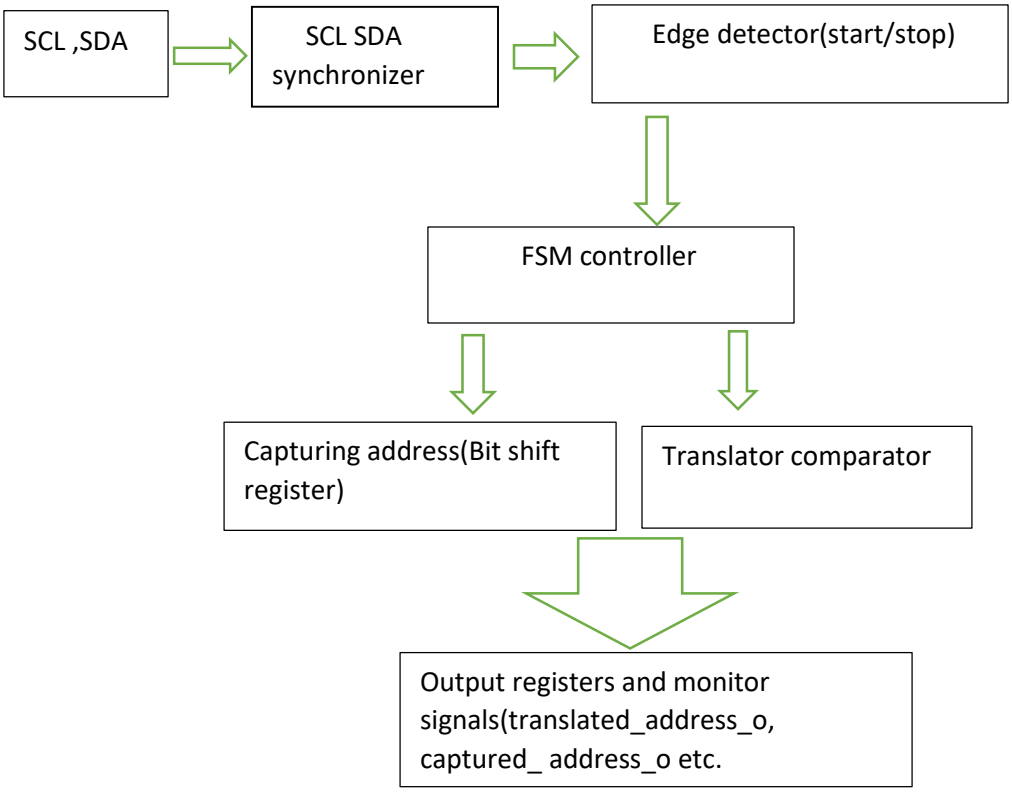
In my design I used 5 states to describe complete operation of Address translator module.This I2C address translator module sits between I2C master and I2C slave .The module-

a)Acts as I2C slave for bus master and master for the target device.

b)Translates the address of one target device at a time and keeps the data intact.

Architectural overview

Primarily the architecture is as follows-



SCL SDA synchronizer-synchronizes asynchronous scl and sda lines to internal clk.

Edge detector –While scl is high and sda goes low (1 to 0) this is the detection of start edge, while scl is high and sda goes high (0 to 1) this is the detection of stop edge.

FSM controller- This stage implements the FSM logic consists of 5 stages-

Idle,Start,address,translate and transfer.

This FSM controls bitcount,enables address capturing and drives datapath sequencing.

Capturing address-captures and collects incoming 7 bit address from SDA line while in the ADDRESS stage.

After receiving the 7 bits,the FSM goes to the translator stage

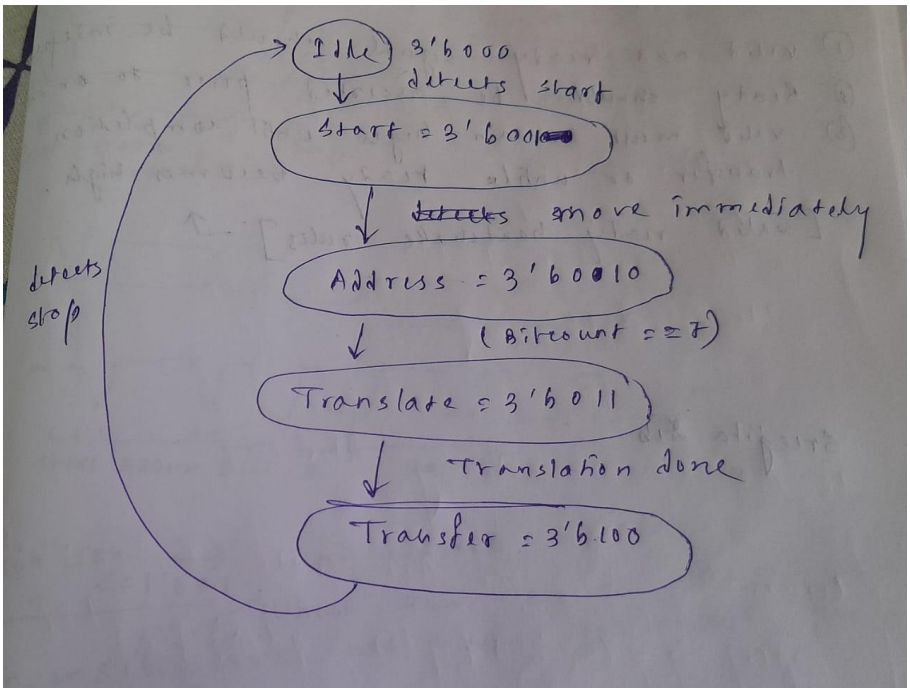
Translation logic and address implementation

Compares the captured address with the given virtual address and decides whether to translate it or not.

Output and debug-

Provides visibility of the internal state for debigging or testbench verification.

FSM LOGIC EXPLANATION-



In this design there are five states to explain the logic.So for describing five states we need 3 bits.

Idle-3’b000- the translator is inactive.When start and enable is true it moves to start state.

Start-3’b001-it indicated the beginning of i2c transaction and resets bitcount and move to next state address.

Address-3’b010-the translator captures 7 bit of address from the sda line during rising edges of scl,when all 7 bits are captured the FSM moves to next state translate

Translate-3’b011-if captured address ==7’h49,it maps to new address 7’h48 and sets trans=1,otherwise trans=0 and it is simply forwarded

Transfer-3’b100-the translator remain active until stop condition is recorded.After detecting stop it returns to the idle state.

Design challenges faced

Bidirectional bus control-i2c lines are open drained.So driving the line incorrectly can damage.Verifying bidirectional bus operation in simulation became a problem .So use of tri state buffers made it possible and correct.

Capturing and translating address within timing window-the translation must occur between address byte and acknowledge phase.So performin translation immediately is onre of the major challenges faced.

Synchronizing scl and sda lines-As i2c is asynchronous so scl and sda lines don’t align with the FPGA clock.So synchrizers are used to ensure clean edge for internal logic.