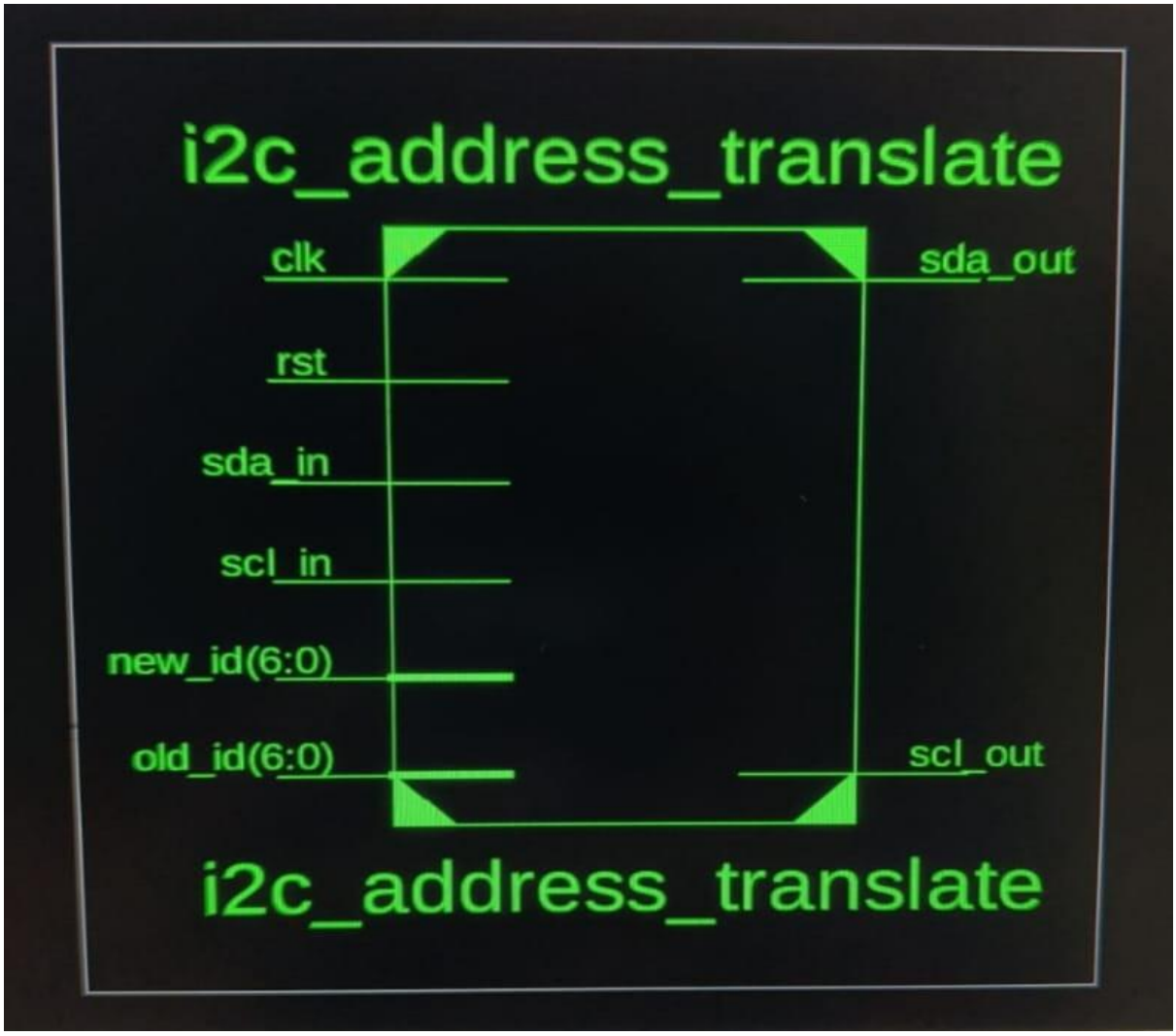
# I2C ADDRESS TRANSLATOR

## 1. Architecture Overview

This design acts like a small block placed between the I²C master and slave.

Its job is to read the incoming 7-bit I²C address and change it to another address before sending it to the slave.




It uses these signals:

Inputs: clk, rst, SDA_in, SCL_in, old_id, new_id

Outputs: SDA_out, SCL_out

Inside the design, the logic reads the address, checks it, and replaces it when it is required.

In short, the block does: read the address → compare it → replace if needed → send it out.

## 2. FSM / Logic Explanation

The design follows the normal working flow of I²C communication.

When the Communication Starts, the circuit waits for a start condition, which happens when SDA goes low while SCL is high.

➔ Reading the Address Bits

After the start, the circuit reads the 7-bit address and the R/W bit.

It reads one bit during each rising edge of SCL_in and stores it inside a register.

➔ Checking the Received Address

When all address bits are collected: If the received address equals old_id, it is replaced with new_id. If it does not match, the original address is kept.

➔ Sending Out the Final Address

The final address is sent out using SDA_out. SCL_out follows whatever is on SCL_in.

➔ When the Communication Ends

A stop condition occurs when SDA changes from low to high while SCL is high.

When this happens, the system resets internal signals and waits for the next I²C frame.

## 3. How Address Translation is Implemented

→The address translation is based on compare-and-replace logic.

→The full incoming 7-bit address is captured.

→The bits are stored in a register.

→After receiving all bits: If the address matches old_id, it is replaced with new_id. If it doesn't match, the address is passed through unchanged.The R/W bit remains the same.

## 4. Design Challenges

→Timing Needs to Be Perfect

I²C is timing-sensitive.Detecting start or stop conditions at the correct time is important for proper operation.

→Sampling Must Be Accurate

If the circuit samples a bit too early or too late, the address becomes incorrect.

Stable timing is required.

→Address Replacement Must Be Smooth

The design must replace address bits cleanly without disturbing the I²C bus signals.

→Handling Noise on the Bus

Small glitches may appear on I²C lines.The design must react only to real changes and ignore noise.