Design of I2C Master Using Verilog

ABSTRACT: This report focuses on the design of a single-master I2C interface , comprising a bi-directional data line(serial data line, SDA) and a serial clock line(SCL). Although configured as a single-master setup, the I2C protocol supports multiple masters. I2C is a two-wire, bi-directional serial bus that facilitates straightforward and efficient data exchange among devices. It enables high-speed devices to communicate seamlessly with slower ones without data loss. With only two lines required for communication ,I2C can manage a network of device chips using just two general-purpose I/O pins, whereas other bus protocols need additional pins and signals for device connectivity. This entire module is implemented in Verilog and simulated in ModelSIM.

INDEX TERMS : Verilog, ModelSIM, I2C bus, Master, slave, SDA, SCL

1.INTRODUCTION: In the realm of serial data communication , protocols such as RS-232, RS-422,RS-485,SPI(Serial Peripheral Interface) and Microwire enable interfacing between high-speed and low-speed peripherals. These protocols, however , demand multiple pin connections on integrated circuits(ICs) to facilitate data transfer. As ICs have become more compact, the need for fewer pin connections for as “point-to-point” data transfer systems. They utilize data path multiplexing and message forwarding to serve multiple devices. To address this limitation, Philips introduced the I2C protocol, designed to enhance communication efficiency between components.

12C PROTOCOL:

The I2C protocol is ideal for applications that require occasional, short-distance communication between multiple devices. It includes collision detection and arbitration features, which prevent data corruption if two or more master devices attempt to control the bus simultaneously. The resource requirements vary depending on how it is implemented. The two I2C bus lines, SDA and SCL, are bi-directional and open-drain, using resistors to pull them up. When sending a logical zero, a device pulls a line to ground; when sending a logical one, the line is released.

Each device connected to the bus has a unique address — whether it’s a microcontroller, LCD driver, memory unit, or keyboard interface — and can function as a transmitter or receiver, depending on its purpose. For example, an LCD driver only receives data, while a memory device can both send and receive data. In data transfer, devices are designated as masters or slaves. The master initiates data transfer and generates the clock signal, while any addressed device acts as a slave for that transfer session.

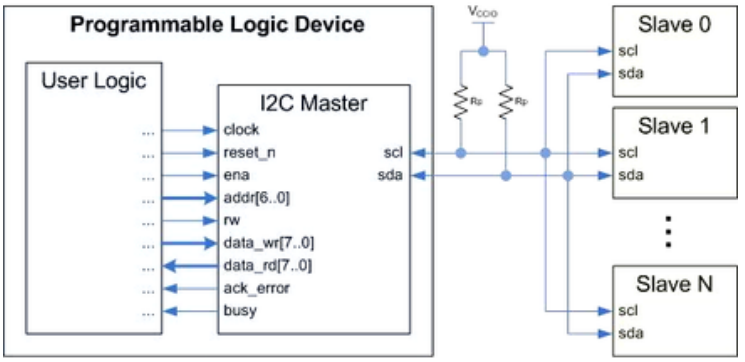


Figure 1: Logical Diagram of I2C Master – Slave

Features

- data transfers: serial, 8-bit oriented, bi-directional
- master can operate as transmitter or receiver
- bit transfer (level triggered)

SCL = 1 , SDA = valid data
one clock pulse per data bit
stable data during high clock
data change during low clocks

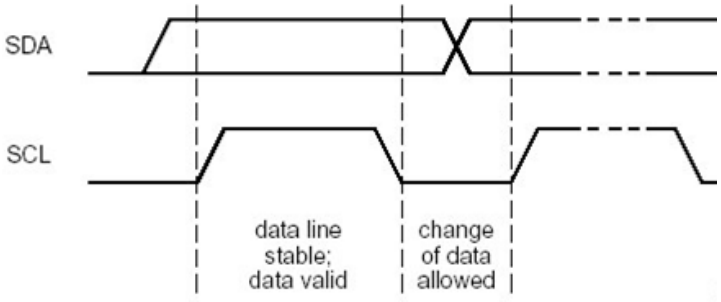
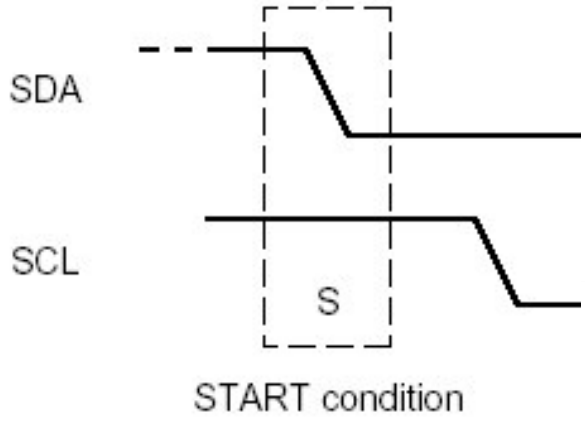


Fig 2: Change of word occurring during low clock

- Start Condition (S)

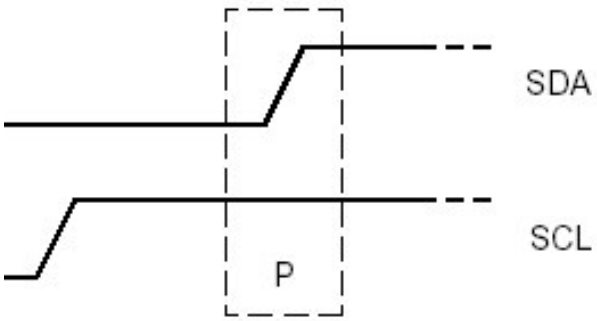
SDA 1 to 0 transition when SCL = 1



START condition

- Stop Condition (P)

SDA 0 to 1 transition when SCL = 1



STOP condition

I2C Master

- controls the SCL
- starts and stops the data transfer mechanism
- controls addressing mechanism of other devices

I2C Slave

- device addressed by master

I2C single Master works as a transmitter or a receiver.

- Master as transmitter sends data to slave-receiver (RW=0 i.e write state)
- Master as receiver requires data from slave transmitter(RW=1 i.e read state) .

2. DESIGN METHODOLOGY

Finite State Machine (FSM) that describes the design of single Master .

Algorithm

State 1: Ready condition: I2C bus doesn't perform any operation.(SCL and SDA remains high) and enable is low. If ena becomes HIGH it enters into next state.

State 2: Start condition: When ena is HIGH, Master initiates data transmission by entering into the next state adr.

State 3: adr state: In this next adr state, master sends the slave address serially (11010000) to the slave. bit_cnt is used as counter to count the bits of address transferred and as it becomes 0,it enters into next state.

State 4: ack state: If the slave address matches with the slave(here single slave is considered hence no need to match it as it is taken as state) it sends an acknowledgement bit in response to the master.

Now R/W bit is checked if it is LOW, it enters write state else read state.

State 5: Write state: The 8 bit data to be transmitted is sent to the slave by the master. After receiving the data, slave acknowledges the master.

State 6: Read state: The 8 bit data is read from the slave by the master. After reading the data, acknowledgement is sent.

State 7: Stop condition: After the transmission of the data, STOP bit is sent.(SCL is high and SDA is from Low to high). Master sends a STOP bit to terminate the connection.

again ena is checked if it is still LOW it remains in STOP state else it enters the READY state. For performing read operation, write operation is performed first and then read operation is done. Slave address used is of 3 bit (010).

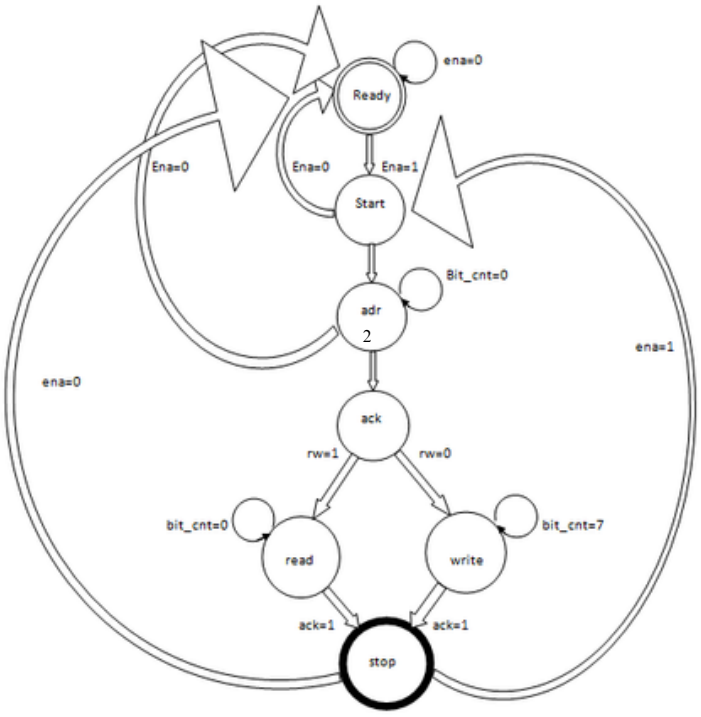


Fig 3: Finite State Machine for Design of Single Master

Functional Description

The functional description of I2C master is described in the Verilog HDL. That is called design module. The test bench program is developed to test the design module. The test bench gives the input to the design module & verifies the output. The test bench is written in such way that the design module can be checked in all possible conditions. The signals given from the test bench to test the working of the prototype design of the I2C master are:

- ena
- ack
- en
- in_data
- sda_r

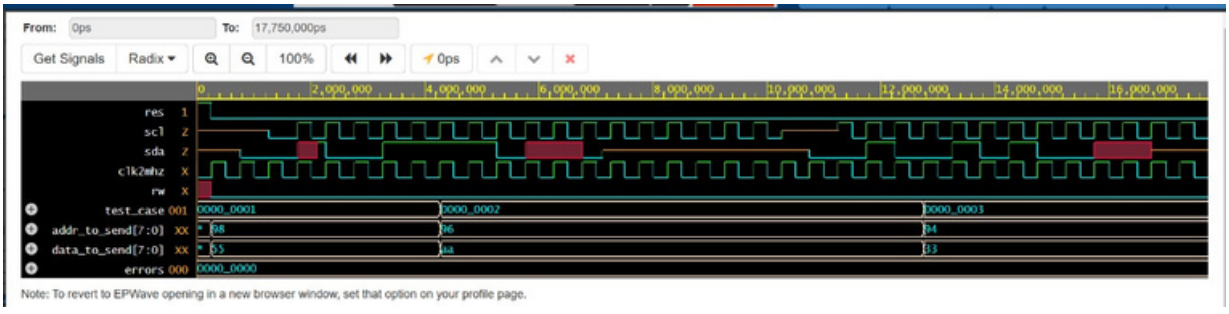
3. SIMULATION RESULTS

The I2C single master has been designed in Verilog and simulated in simulator tool ModelSim which is used to verify the design functioning. This design works for both read and write cycle.

Expected Timing for Each Test Case:

1. Test Case 1 (Basic Write):
   - Start condition around 250ns (after reset)
   - Complete transaction should show:
     - Start → Address(0x98) → ACK → Data(0x55) → ACK → Stop
2. Test Case 2 (Second Write):
   - Starts ~1000ns after Test Case 1
   - Should show similar pattern:
     - Start → Address(0x96) → ACK → Data(0xAA) → ACK → Stop
3. Test Case 3 (NACK Response):
   - Starts ~1000ns after Test Case 2
   - Shorter transaction due to NACK:
     - Start → Address(0x94) → NACK → Stop

```
Simulation completed with 0 errors
$finish called from file "testbench.sv", line 129.
$finish at simulation time             17755000
           V C S   S i m u l a t i o n   R e p o r t
Time: 17755000 ps
CPU Time:      0.370 seconds;       Data structure size:   0.0Mb
```



Simulation result

4. FPGA SYNTHESIS RESULTS

XILINX 14.1 has been used for the synthesis of Single Master on FPGA.

RTL Schematic diagram: Figure 4 shows the Resistor transistor logic diagram.

Figure 5 shows the diagrammatic view of connection of given input output at Register Transfer Level (RTL)

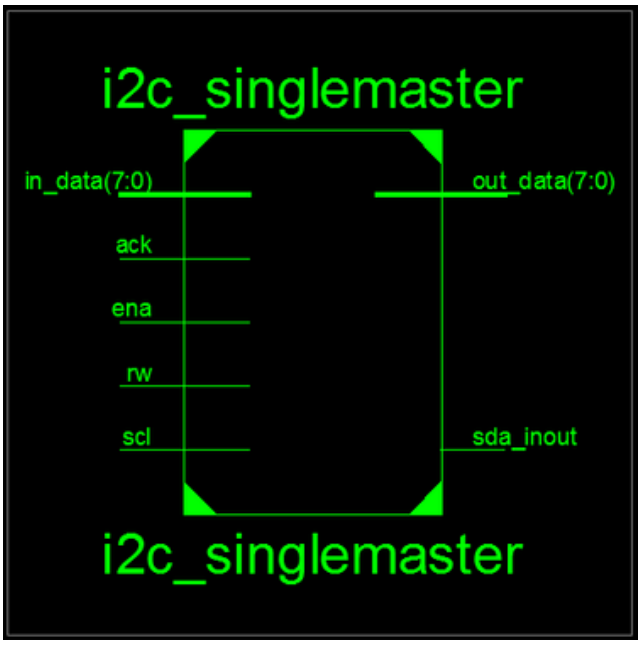


Fig 4: RTL Schematic Diagram of I2C Single Master

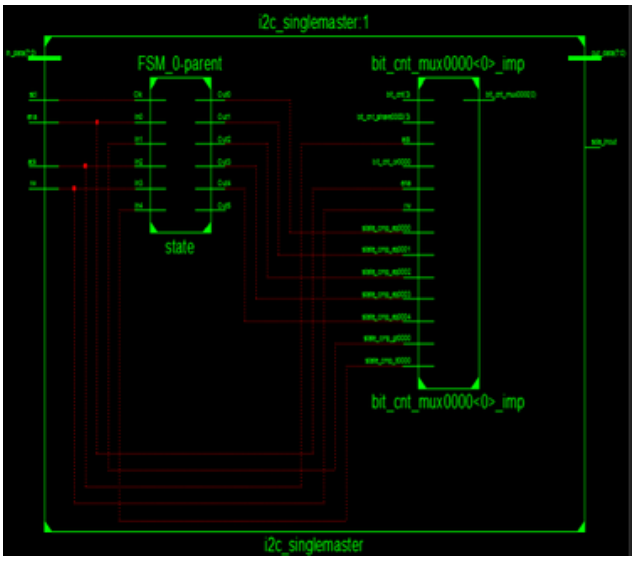
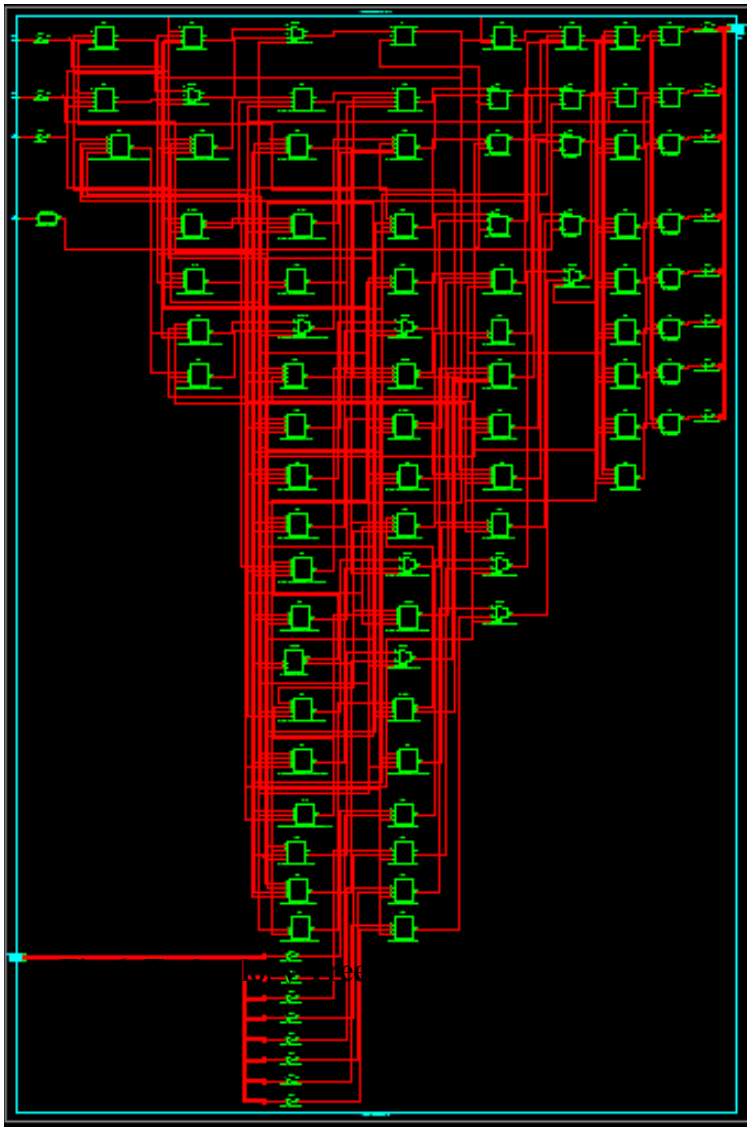


Fig 5: Diagrammatic view of given input output at RTL level

Technology Tree schematic: Figure 6 shows the technology tree of the I2C Single Master



5.Functional use amd capability of I2C master

The I2C master's function is to initiate communication with slave devices on the I2C bus:

Controls the clock: The master device controls the clock line (SCL) for all slave devices.

Initiates communication: The master initiates communication by sending a clock signal, pulling the data line row, and sending address bits on the data line.

Selects a slave: The master selects a slave device by sending a unique address on the bus.

Terminates communication: The master terminates communication by signaling a stop condition and stopping the clock signal.

Issues a restart condition: The master can issue a restart condition instead of a stop condition to hold the bus after completing data transfer.

The I2C bus is a bi-directional interface that uses a master-slave protocol. The I2C interface is flexible and can communicate with slow devices , as well as a transmit large data at high speeds.

6.CONCLUSION & FUTURE SCOPE

I2C Single Master is successfully designed and simulated.As the number of devices connected to a system is going to increase, there is a need for a system which supports multiple protocols. This project can be further extended to design for multiple masters.

References

[1] Samir Palnitkar ,—Verilog HDL A guide to Digital Design and Synthesis‖ SunSoft Press,1996

[2] Stuart Sutherland, —Verilog HDL Quick Reference Guide‖, IEEE Std 1364-2001

[3] M.Morris Mano, —Digital Design‖ EBSCO publishing. Inc., 2002

[4] Philips Semiconductor —I2C Bus Specification‖, April 1995

[5] Philips Semiconductor —I2C Bus Specification‖ version 2. 1, January 2000

Shravani Mamidwar, IIT(BHU) Varanasi

Bandlamudi Manogna, R.V.R& J.C College of Engineering

Sanjana Tomar, Banasthali Vidhyapith

Arvapally Saichandana, NIT RAIPUR