

I2C

DESIGN OF I2C MASTER CONTROLLER USING VERILOG

INTRODUCTION

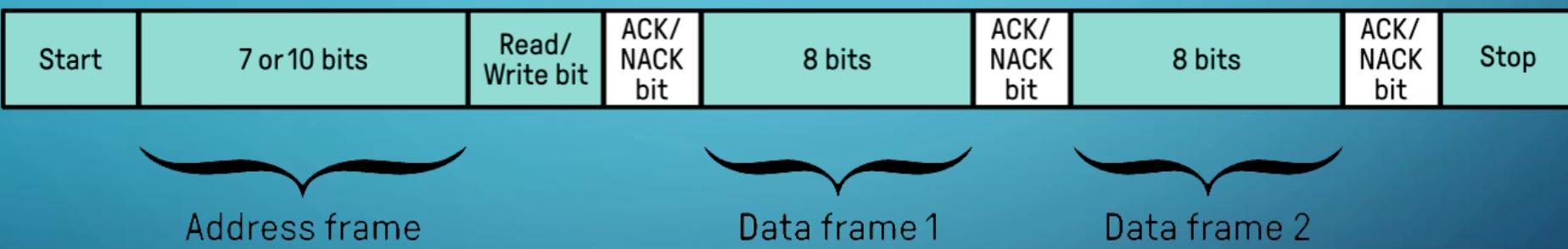
- The I2C (Inter-Integrated Circuit) protocol is a popular and simple two-wire serial communication standard used in electronics. It enables data exchange between various devices, including microcontrollers, sensors, memory modules, and more. With its minimal hardware requirements and support for multiple devices on a single bus, I2C is widely adopted for efficient and versatile communication in embedded systems.

FRAME

- An I2C frame consists of a series of information and control bits exchanged between a master device and one or more slave devices over a two-wire bus (SDA & SCL). The essential components of an I2C frame are as follows:

- 1) Start Condition
- 2) Slave Address
- 3) R/W Bit
- 4) Acknowledge (ACK)
- 5) Data Bytes
- 6) Stop Condition

Message



- **Start Condition:** The communication begins with the master device generating a start condition. This is a special transition on the SDA (Serial Data) line while the SCL (Serial Clock) line remains high. It signals the beginning of a data transfer
- **Slave Address:** Following the start condition, the master sends the address of the slave device it intends to communicate with. This address includes the 7-bit or 10-bit address of the target slave. The most significant bit (MSB) typically indicates whether it is a write (0) or read (1) operation.
- **R/W Bit:** The Read/Write (R/W) bit follows the slave address bit and specifies whether the master intends to write data to the slave ($R/W = 0$) or read data from the slave ($R/W = 1$).

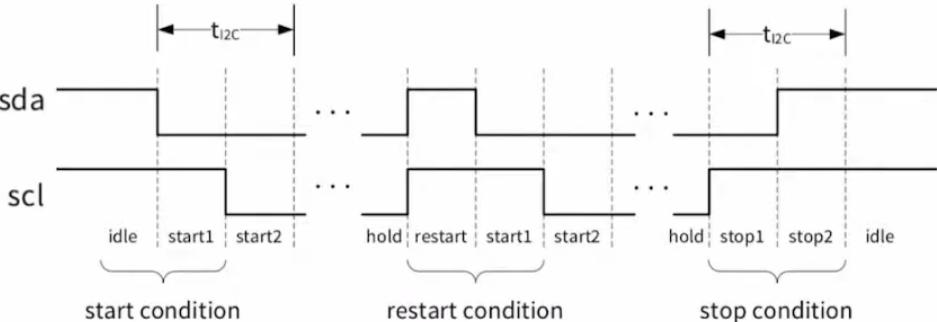
- **Acknowledge (ACK):** After each byte of data (either address or data payload), the receiver (slave or master) sends an acknowledgment bit (ACK/NACK) on the SDA line. An ACK is typically pulled low to acknowledge the successful receipt of data. A NACK (no acknowledgment) indicates that the receiver is not ready or unable to proceed.
- **Data Bytes:** Data bytes are transferred following the address and R/W bit. Both the master and slave can transmit data bytes in sequential bytes if needed. Each byte is typically 8 bits in length.
- **Stop Condition:** The communication concludes with a stop condition, which is another special transition on the SDA line while the SCL line remains high. The stop condition indicates the end of the data transfer and releases the bus for other devices to use.

OPERATION

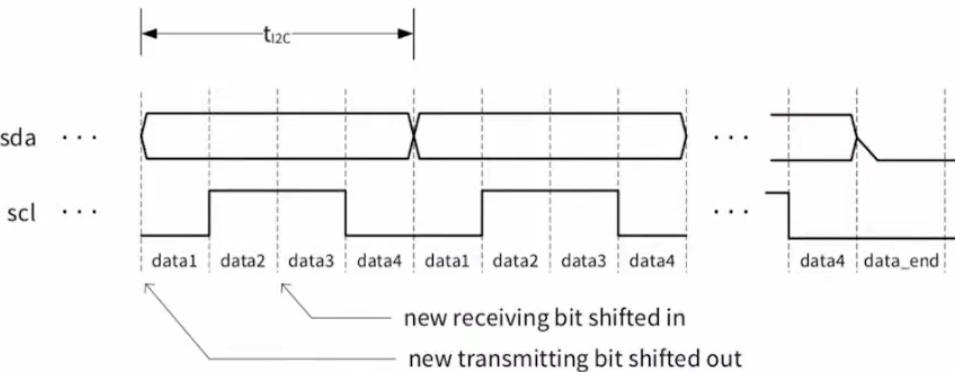
- The I2C system uses a serial data line (SDA) and a serial clock line (SCL) for data transfers. All devices connected to these two signals must have open drain or open collector outputs. The logic AND function is exercised on both lines with external pull-up resistors.

- Data is transferred between a Master and a Slave synchronously to SCL on the SDA line on a byte-by-byte basis. Each data byte is 8 bits long. There is one SCL clock pulse for each data bit with the MSB being transmitted first. An acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (see START and STOP signals).

I²C Action Phases



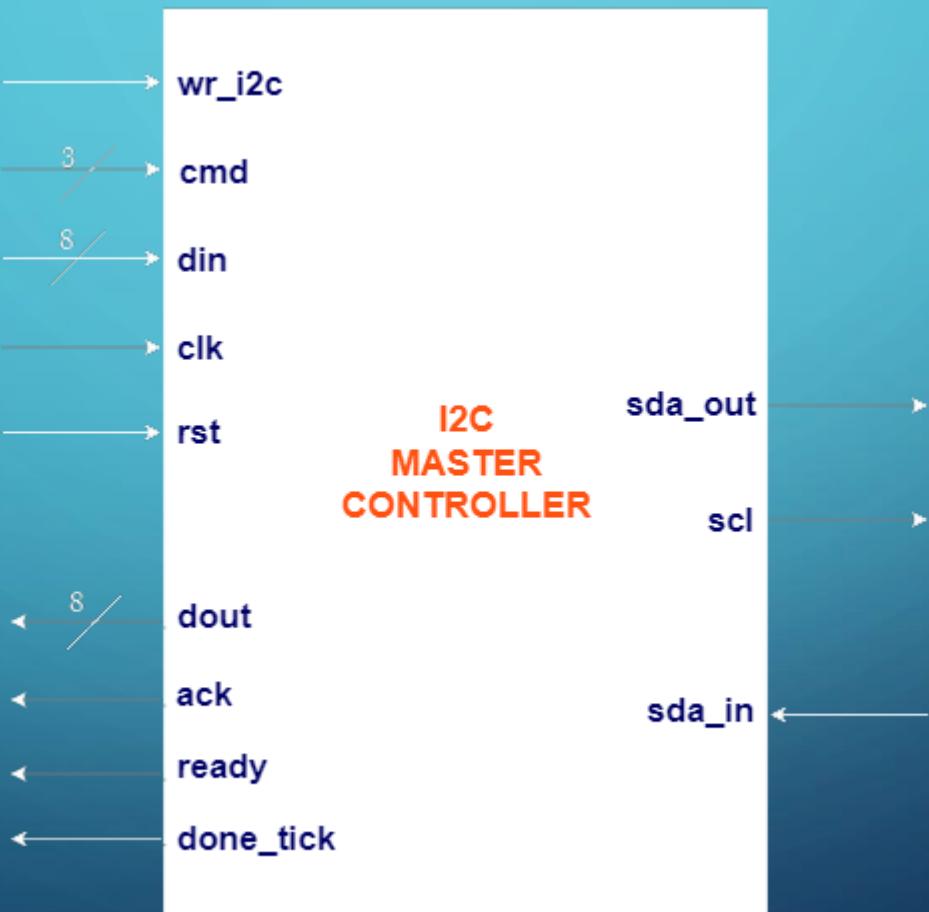
(a) Division of control conditions





□ VERILOG IMPLEMENTATION

BLOCK INTERFACE



- I²C Master will send the start condition, slave address, R/W bit then depending on write or read, it sends or receives the data. Then slave will send the ACK bit after receiving the slave address and after receiving the data through SDA line. So, SDA line has to be controlled by both I²C Master and I²C Slave, which becomes a problem while implementing I²C in Verilog. Instead of using same SDA line by both, that SDA line can be split into two i.e., SDA_out & SDA_in for I²C master block and I²C Slave block respectively.

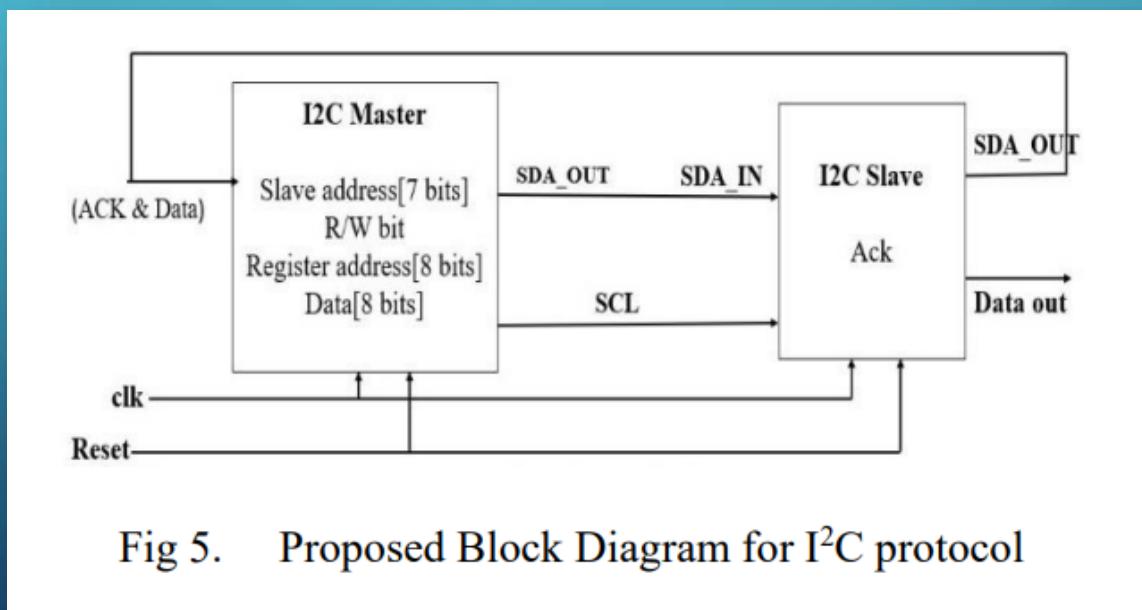


Fig 5. Proposed Block Diagram for I²C protocol

```
I2C_controller.v
I2C_controller > I2C_controller
1 module I2C_controller (
2     input clk ,rst,
3     input wr_i2c,
4     input [2:0] cmd,
5     input [7:0] data_in,
6     input [15:0] dvsr ,
7
8     output [7:0] data_out,
9     output ack,
10    output reg ready,
11    output reg done_tick,
12
13    //inout sda,
14    input sda_input_s,
15    output sda_output_m,
16    output scl
17 );
18 /*
19 parameter [2:0] cmd_start = 3'b000 ,
20             cmd_wr = 3'b001 ,
21             cmd_rd = 3'b010 ,
22             cmd_stop = 3'b011 ,
23             cmd_restart = 3'b100 ;
24 /*
25 parameter state_reg_width = 4;
26 parameter [state_reg_width-1:0] idle_state = 0,
27             start1_state = 1,
28             start2_state = 2,
29             hold_state = 3,
30             stop1_state = 4,
31             stop2_state = 5,
32             data1_state = 6,
33             data2_state = 7,
34             data3_state = 8,
35             data4_state = 9,
36             data_end_state = 10,
37             restart_state = 11;
```

□ GITHUB REPO FOR THE I2C MASTER CONTROLLER VERILOG CODE

- <https://github.com/MostafaEsa mAbdelhameed/I2C>

□ SIMULATION RESULTS

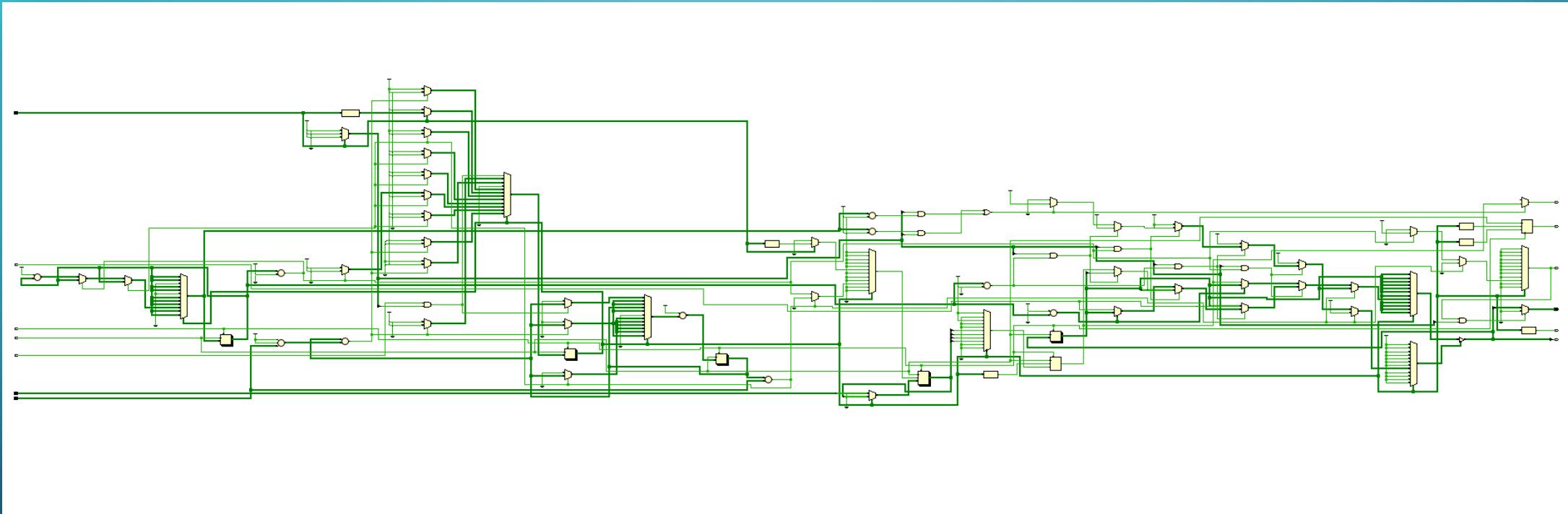


```

run all
For the Write Task
The Recived data = 101010100
Data = 10101010
Acknowledge = 0

For the Read Task
The Recived data = 111100000
Data = 11110000
Acknowledge = 0

```



□ SYNTHESIS RESULTS

- The synthesis process was performed using the Design Compiler tool.
- Technology_Library → tsmc130nm

Constrain	Value
Clock_Period	10ns
Clock Uncertainty Setup & hold	0.2ns
Clock Transition Rise & Fall	0.05ns
Clock Latency	0
Input Delay	20% of clock period
Output Delay	20% of clock period
Input driving cell	BUFX2M
Output Load	0.5pf
Don't touch	Clk , Rst

• Area

Number of ports:	76
Number of nets:	537
Number of cells:	429
Number of combinational cells:	384
Number of sequential cells:	44
Number of macros/black boxes:	0
Number of buf/inv:	53
Number of references:	66
Combinational area:	3956.065462
Buf/Inv area:	254.167204
Noncombinational area:	1168.463104
Macro/Black Box area:	0.000000
Net Interconnect area:	undefined (No wire load specified)
Total cell area:	5124.528566
Total area:	undefined

• Power

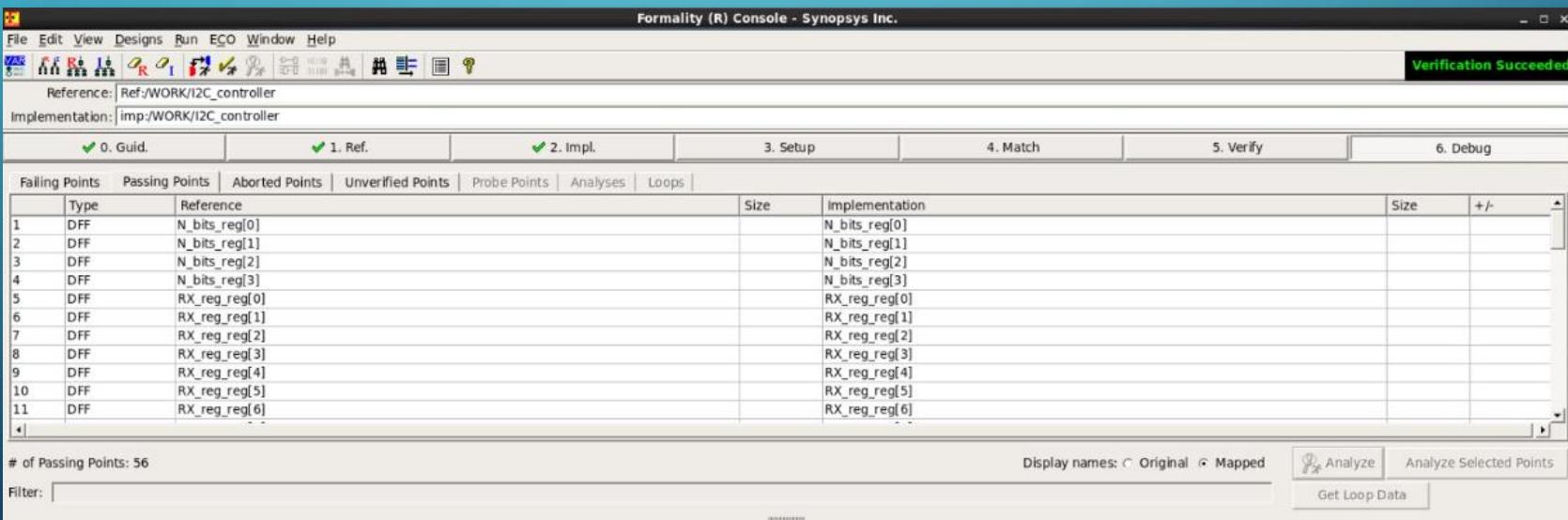
Hierarchy	Switch Power	Int Power	Leak Power	Total Power	%

I2C_controller	9.27e-03	8.18e-02	3.67e+06	9.47e-02	100.0
add_74 (I2C_controller_DW01_inc_0)	1.46e-04	3.92e-04	2.08e+05	7.46e-04	0.8
1					

• STA

```
1 |
2 ****
3 Report : constraint
4      -all_violators
5 Design : I2C_controller
6 Version: K-2015.06
7 Date   : Thu Oct  5 23:01:11 2023
8 ****
9
10 This design has no violated constraints.
11
12 1
```

• Formality (Post-Synth)



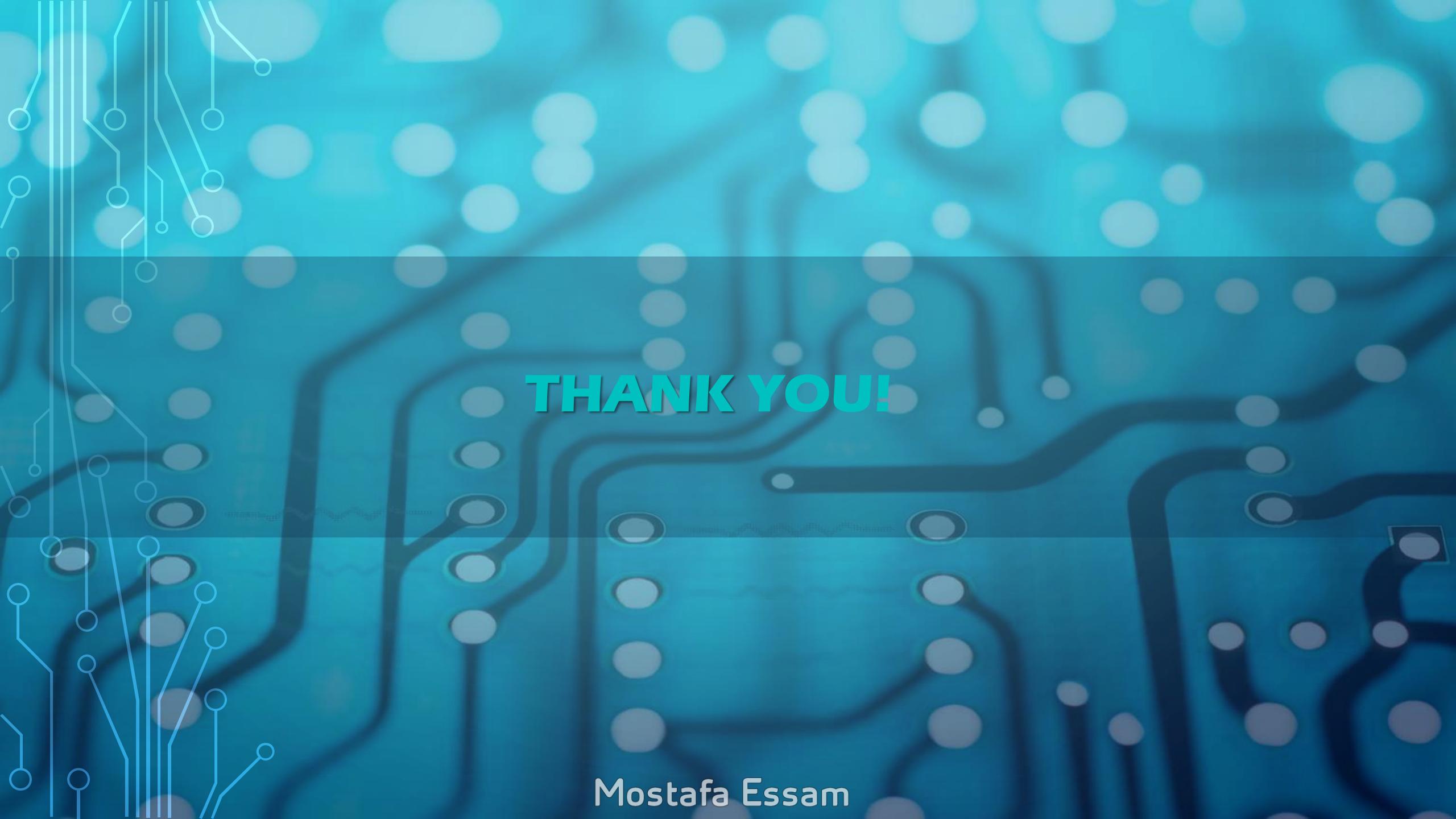
- **Github Project Link:**

github.com/MostafaEssamAbdelhameed/I2C

- **Contact Info**

LinkedIn : linkedin.com/in/mostafa-essam-abd-el-hamid-03b04b22b/

E-Mail : mostafaessamabdelhamid@gmail.com



THANK YOU!

Mostafa Essam