ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1582

Serial Peripheral Interface Conversion for Fast Data Transfer

Swathi Burra[1]; Dr. E. Krishnahari[2]
Department of ECE Engineering
Holy Mary Institute of technology Gatkesar, Hyderabad – 501301

Abstract:- The reason of this paper is to plan and mimic a Convention between the two broadly acknowledged serial communication conventions SPI and I2C. PI may be a serial transport and is exceptionally common within the inserted world. SPI bolsters full duplex communication with higher throughput than I2C. Many inserted frameworks have as it were SPI Interfaces, making them troublesome to associate with I2C peripheral gadgets. You'll adjust the associations, but the coming about framework isn't proficient. One of the most excellent ways to bargain with this issue is to form an SPI-to-I2C Interface and actualize it in FPGA. This paper clarifies approximately the communication between SPI and I2C, where it moreover amplifies up to USB communication. The transmission of information from MOSI and MISO utilizing SIPO, PIPO and at long last we will utilize FIFO to stack the information and check outcomes about in FPGA through recreation prepare. I2C, a serial transport designed by Philips, is utilized to communicate with low-speed peripherals. It employments two bidirectional open-drain lines: Serial Information (SDA) and Serial Clock (SCL). The master at first sends a begin bit, taken after by the 7-bit address of the slave it wishes to communicate with, which is at long last taken after by a single bit speaking to whether it wishes to type in (0) to or studied (1) from the slave. In case the slave exists on the transport, it'll react with an ACK bit (dynamic low for recognized) for that address. The master at that point proceeds in either transmit or get mode, and the slave proceeds in its complementary mode. Every data byte put on the SDA line has to be 8-bits long. At last, the recreation results are found in the test bench behavioral displaying in Simulator Window in Xilinx ISE Block of the respective test bench.

Keywords:- *Serial Peripheral Interface(SPI), Inter-Integrated Circuit (I2C), Serial in Parallel Out (SIPO), Master Out Slave in (MOSI), Master in Slave Out (MISO), Serial Clock (SCLK) for SPI, Slave Select (SS), Serial Clock (SCL) , Serial Information (SDA), to begin with in to begin with Out(FIFO).*

I. INTRODUCTION

For serial communication two conventions specifically Serial Peripheral Interface (SPI) and Inter-Integrated Circuit (I2C) are utilized. A Serial Peripheral Interface (SPI) encourages short-distance communication between Peripheral coordinate's circuits and microcontrollers. In this article, we are going get it the components of SPI, applications of Serial Peripheral Interface (SPI), and more. SPI stands for Serial Peripheral Interface. It may be a convention that's synchronous serial communication. It is utilized to communicate between the Peripheral gadgets i.e. input and yield gadgets and microcontrollers. It is permitted to exchange high-speed information. It is prevalent with computerized communication applications and implanted frameworks. SPI can exchange the information and get information from one gadget to another gadget at a time.

Serial Peripheral Interface (SPI) is the method of synchronous serial communication convention. It is primarily utilized for interfacing the microcontrollers to Peripheral gadgets like sensors, shows, and memory chips. It encourages the full-duplex, synchronous serial communication between one or more slave devices and a microcontroller.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1582

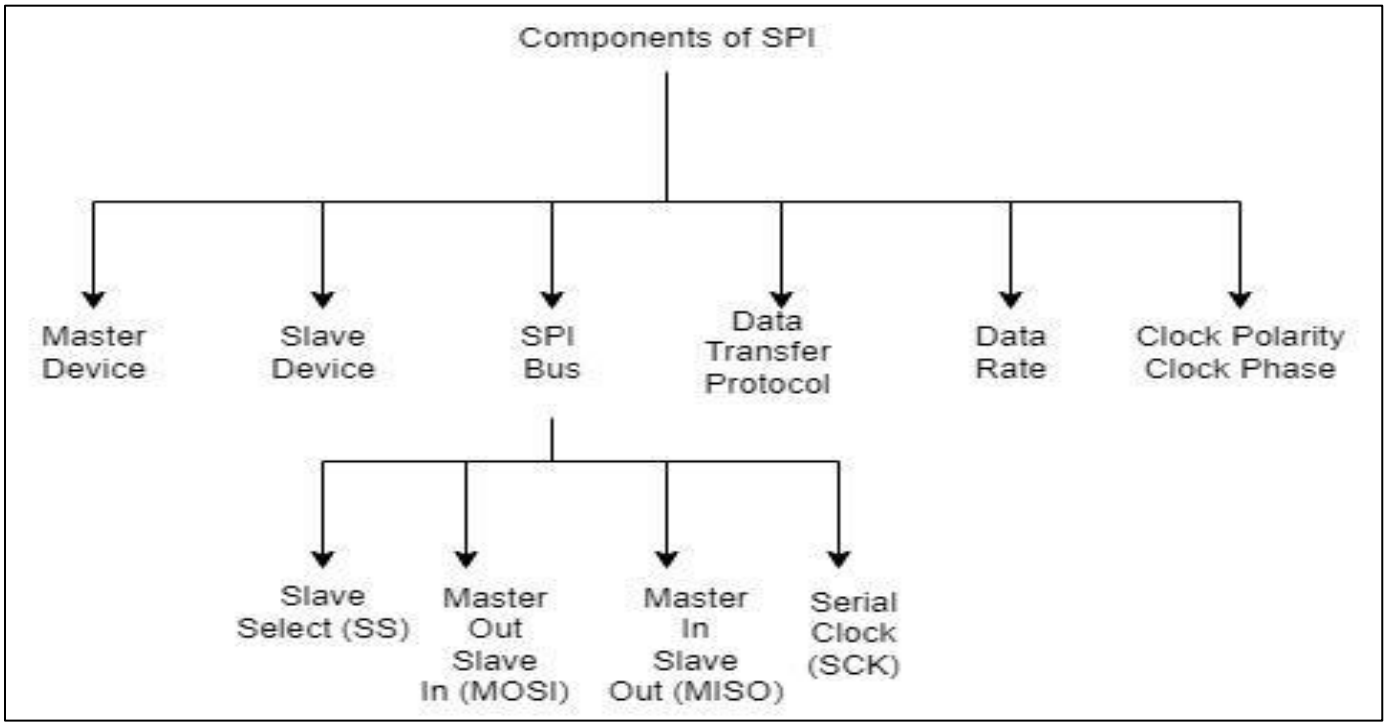

Fig 1: SPI Devices

II. SPI PERIPHERALS

➢ *Master Gadget:*

The master gadget is nothing but it controls the method of change of information on the SPI transport. It controls the information stream and it creates the clock flag. In most of the applications, the master gadget is the microcontroller or specialized SPI controller.

➢ *Slave Gadget:*

Slave gadgets are Peripheral gadgets that are associated to the SPI bus and controlled by master gadgets. Each slave gadget contains a different slave select (SS) line, permitting the master to choose which device it wants to communicate with.

➢ *SPI Transport:*

SPI transport could be a physical association over the information exchanging between the slave gadgets and the master. It contains four flag lines as underneath.

➢ *Slave Select (SS):*

In Slave Select, each slave gadget contains a devoted SS pin. In the event that the master will communicate with the particular slave. Numerous slave gadgets can be shared with the as same as MOSI, MISO, and SCK lines but it must have isolated SS lines.

- Master Out Slave In (MOSI): In Master Out Slave In, MOSI can share the information or data from the master to other slave gadgets.
- Master In Slave Out (MISO): In Master In Slave Out, MISO can share the information or data from the slave gadget with the master.
- Serial Clock (SCK): In Serial Clock, this clock flag is utilized by the master and the slave gadgets for planning the information exchange timings.
- Data Transfer Protocol: SPI is used as a synchronous serial communication for simple transferring of data. The data is transferred and received at the same time in full duplex mode. By generating the clock pulses, the master-slave will initialize the transfer of the data. In every clock cycle, one bit of data will be transmitted both from master to slave and from slave to master directions.

➢ *Data Rate:*

The SPI transport can bolster the distinctive information exchanging rates depending upon the slave capabilities of the slave gadgets and the transmission line's length. The information rate is indicated in bits per megahertz (MHz) or moment (bps).

➢ *Clock Extremity (CPOL) and Clock Stage (CPHA):*

These are utilized to characterized the relationship between the information signals and the clock signals. The information signals are nothing but, MOSI and MISO are called as the information signals. The SCK is called as the clock flag. There are accessible in four distinctive conceivable combinations of CPHA settings and CPOL, they are permitting adaptable to arranging to the SPI Interface for work with the diverse gadgets.

➢ *I2C*

I2C stands for Inter-Integrated circuit. It can moreover be alluded to as IIC or I2C. I2C Convention may be a serial communication convention, and it is broadly utilized for short-distance communication. It gives straightforward and vigorous communication between the Fringe gadget and the microcontroller.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1582

I2C Convention comprises of two wires SDA and SCL which is bidirectional synchronous serial transport communication. Hence, I2C Convention takes two wires for communication which moreover deciphers to moo taken a toll and this moo taken a toll has made I2C Convention the foremost commonly utilized Serial Transport over most applications counting IoT, shopper gadgets, car, aerospace master, and mechanical equipment.

I2C Convention could be a synchronous convention that permits a master to start communication with a slave gadget. I2C convention may be a master-slave communication where the slave gives the clock which gets to be the information exchange rate or clock recurrence. It may be a bi-directional bus meaning the slave can compose to the slave and perused from the slave it could be a serial transport which implies information could be a clock and it is moved bit by bit and there are two transport lines serial clock (SCL) and serial information (SDA).As u see within the graph three slaves are associated to the same I2C Slave and there's two pull-up resistor which is required for the I2C gadget to communicate appropriately usually since the I2C convention works on the SCL and SDA transport lines which are an open deplete or open collector the transmitting gadget fair lets go of the I2C transport to form rationale one and pulls or drives the line the ground to make rationale 0.In I2C Convention there are 5-speed categories counting standard mode, quick mode, quick additionally mode, high-speed mode, and ultra speed mode these speed categories run from 100 kHz to 5MHz, where standard mode is 100KHz, Quick mode is 400KHz, Fast mode furthermore is 1MHz, High-Speed Mode is 3.4MHz and Ultra-fast mode is 5MHz. 100KHz up to 1MHz are very comparable, whereas 3.4 MHz needs a few extraordinary thought, and 5MHz being unidirectional requires indeed more uncommon consideration.

But the foremost common speed categories for I2C Convention are Standard mode, Quick Mode, and Quick mode additionally these modes are simple to implement.I2C Convention exchanges are started with a begin condition, the begin condition is characterized as the slave driving SDA line moo whereas SCL remains tall, and note that it must be started with the I2C transport in an sit out of gear state implies SDA and SCL lines are both tall.

The slave address quickly takes after the start condition, and it'll be 7 bits or 10 bits long it ought to be famous that each gadget on the I2C transport needs a one of a kind slave address. Following is the read-and-write bit, which promptly takes after the slave's address, this bit illuminates the slave on the off chance that the slave needs to examined or type in. 1 demonstrates the slave needs to studied from the slave and demonstrates the slave needs to compose.

The next section is the Recognize bit and able to think of ack bits like a handshake between the slave and slave and they happen on each 9th clock cycle, in the event that ACK is at that point information can be transmitted, and in the event that ACK is 1 information cannot be sent. Another is the information byte which contains the data being exchanged between the slave and slave, It contains 8 bits of data. The final segment of the I2C convention is the Halt condition all I2C exchanges ought to be ended with a halt condition, the halt condition is characterized as the slave discharging the line whereas the SCL flag level is tall after a halt condition the I2C transport will stay in sit still state and won't be free for the another I2C transaction. There are two vital concepts of I2C Convention:

➢ *I2C Assertion and I2C Clock Stretching I2C Assertion:*

I2C Convention may be a multi-master transport it can be conceivable that different experts are driving the transport at the same time, In case one of them recognizes SDA is moo when it ought to be tall, it expect that another slave is dynamic and instantly halt its exchange.

By now, you should have enough information to decide between I2C vs SPI transmission protocols for your embedded device engineering project. If you can afford the additional cost, it's often better to go with the SPI protocol unless you have a compelling reason to choose I2C. These could include factors like:

- The project requires transmission distances of longer than 10 meters where noise and interference are significant factors
- The project requires multiple master devices, the configuration not supported by SPI protocol
- On-chip real estate is at a premium and you would prefer to have just two wires rather than four, as with the I2C Interface protocol.

Once you've chosen between the I2C vs SPI protocols for your embedded systems project, you'll need to invest in the right diagnostic tools that can reduce your time-to-market and validate the correct performance of your product.

➢ *USB Data Format*

- In USB protocol, master devices are known as USB hosts which start all the communication that happens above the USB bus. Here, a computer otherwise other controller is usually considered as the master device, so if they request any information they only respond to other devices. The slave device or peripheral is connected simply to the host device which is programmed to provide the host device with the information it requires to operate. In general, slave or peripheral devices mainly include keyboards, mouse of computers, USB flash drives, cameras, etc.
- It is very essential for host devices to communicate effectively with each other. Once the peripheral device is connected to the computer through USB, then the computer will notice what type of device it is & load a driver automatically that permits the device to function.
- The small amount of data transmitted between the two devices is called as ‘packets’ where a unit of digital information is transferred with every packet. The data transfer that can be occurred within the USB protocol is discussed below.
- Message Format

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1582

- The data of the USB protocol is transmitted within packets LSB first. There are mainly four types of USB packets Token, Data, Handshake & Start of the Frame. Every packet is designed from various field types which are shown in the following message format diagram.

➢ *Message Format Diagram of USB*

- *SYNC*

✓ In USB protocol, every USB packet will begin with a SYNC field which is normally utilized to synchronize the transmitter & the receiver to transmit the data precisely. In a slow or high-speed USB system, the field like SYNC includes 3 KJ pairs which are followed through 2 K's to frame 8-bits of data.
✓ In a Hi-Speed USB system, the synchronization needs 15 KJ pairs followed through 2 K's to frame 32-bits of data. This field is long with 8 bits at high &low speed otherwise 32-bits long for maximum speed & it is utilized to synchronize the CLK of the transmitter & receiver. The final 2-bits will indicate wherever the PID field begins.

- *Packet Identifier Field or PID*

✓ The packer identifier field within the USB protocol is mainly used to recognize the packet type that is being transmitted and thus the packet data format. The length of this field is 8 bits long where the upper 4- bits recognize the kind of packet & lower 4- bits are the bit-wise complement of the upper 4- bits.

- *Address Field*

✓ The address field of the USB protocol indicates which packet device is mainly designated for. The 7-bits length simply allows support of 127 devices. The address zero is invalid because any device which is not yet allocated an address should be reacted to transmitted packets to the zero address.

- *Endpoint Field*

The endpoint field within the USB protocol is 4-bits long & allows for extra flexibility within addressing. Usually, these are divided for the data moving IN/OUT. Endpoint '0' is a special case called as the CONTROL endpoint & each device includes an endpoint 0.

- *Data Field*

The length of the data field is not fixed, so it ranges from 0 to 8192 bits long & always an integral the number of bytes.

- *CRC Field*

The Cyclic Redundancy Checks (CRC) are executed on the data in the packet payload where all the token packets include 5-bit CRC & the data packets include a 16-bit CRC. The CRC-5 is five bits long & used by the token packet as well as the start of the frame packet.

- *EOP Field*

Every packet is terminated by an EOP (End of the Packet) field which includes an SE0 or single-ended zero for 2-bit times followed through the J for 1-bit time.

III. DESIGN

But the MOSI Slave select, this is often the SPI slave. This is often a 32-bit SPI slave, which can work with distinctive CPOL and CPHA settings. The top-level bland sets the CPOL and CPHA settings. The default settings are CPOL = and CPHA = 0. The SPI slave gets a command from the outside SPI slave and passes it to SPI_I2C controller. It moreover sends the information studied from the outside I2C slave and status of I2C slave back to the external SPI slave through the MISO block. 2. SPI_I2C Controller piece. Typically the most piece that performs the SPI-to-I2C work. The SPI slave gets the command from an outside SPI slave and passes it to this select. The SPI_I2C controller translates the command and takes suitable activity.

The four MSB of the flag from the SPI slave (MOSI) characterizes the message. For a type in or examined operation to the I2C slave, the SPI_I2C controller designs the I2C slave to send the I2C slave address, and after that sends or gets information to or from the outside I2C slave. Amid the "Send the studied information back to outside SPI slave" and "Read the status of I2C master" commands, the SPI_I2C controller sends the information or status back to the SPI slave through the SPI slave select. 3. I2C Slave Select:

The I2C slave employments a custom I2C center. This IP center has four generics that are utilized to arrange diverse modes of operation.

➢ *SYNC_BE:*

This is often utilized to arrange the yield signals from the center. On the off chance that '0', the yield signals are synchronous to the SCLK clock and substantial for a whole SCLK clock. In case '1', the output signals are synchronous to the input clock.

➢ *MODE:*

Usually utilized to indicate the speed of the I2C exchanges. In the event that '0', the STD speed (100 Kbps) is chosen and on the off chance that '1', the Quick speed (400 Kbps) is selected. This bland and REFCLK_SPEED are utilized to arrange the clock rate for the mode.

➢ *REFCLK_SPEED:*

Usually used to indicate the recurrence input to the reference clock input.

➢ *BC_WIDTH:*

Typically utilized to indicate the width of the byte counter within the slave center.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1582

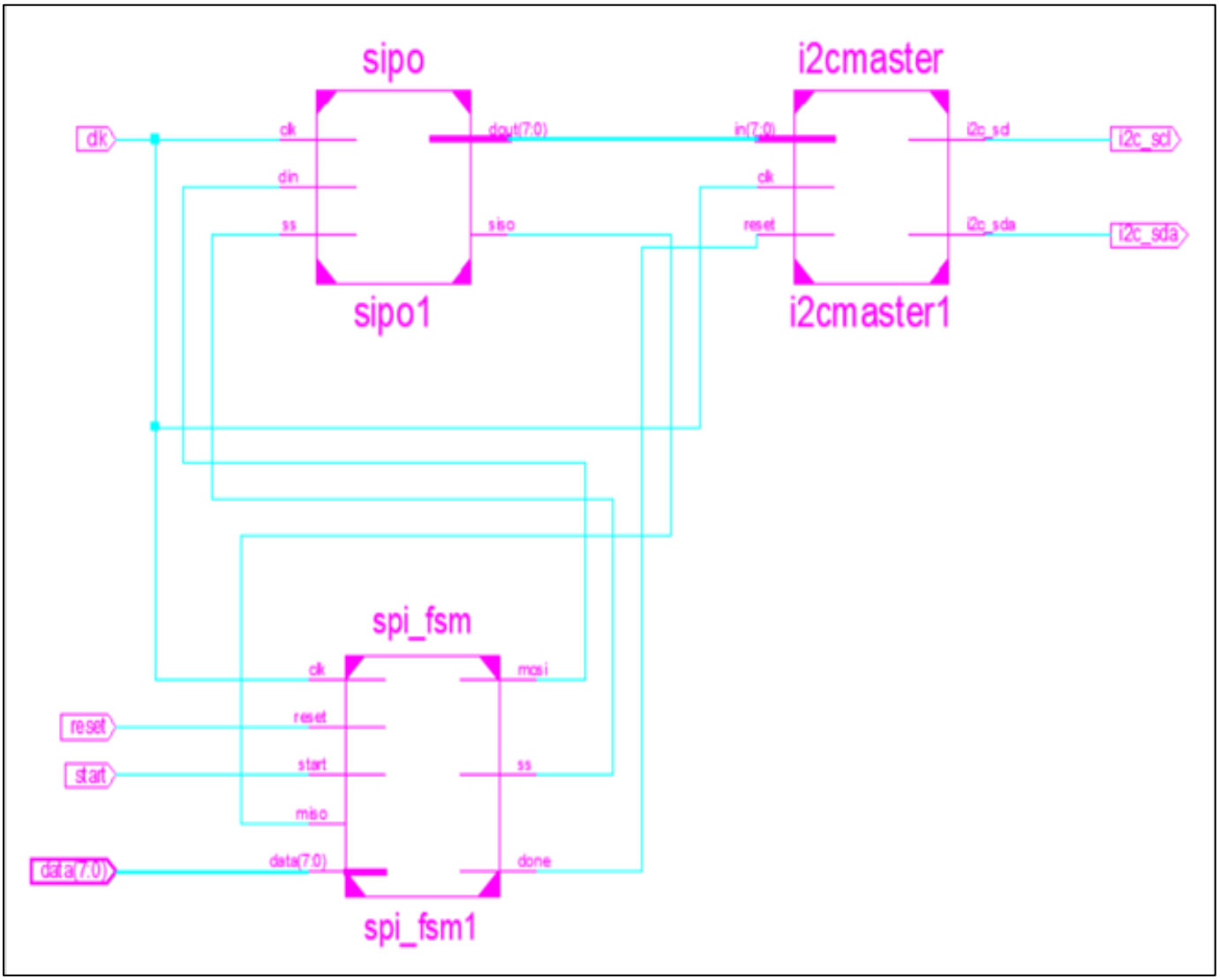

Fig 2: Schematic Block Diagram

IV. RESULT

This design was simulated in Xilinx ISE 14.5 and simulated using ISIM simulator. In order to observe simulation results a top level design called SPItoI2C was implemented. It had three instances:

- Spi_Master: SPI master (sender) device. It was done using Finite State Machine (FSM).
- SIPO: Serial In Parallel Out register.
- I2cMaster: I2C master device .It was implemented using Finite State Machine (FSM).

But the Confirmation of the center is done by recreation in Xilinx ISE. Equipment approval is done on virtix 7 improvement board. In recreation confirmation, the test seat makes a framework with an SPI ace, SPIto-I2C Interface plan, and an I2C slave. The SPI ace signals are created utilizing CPOL = and CPHA = settings. Be that as it may, you'll effortlessly alter the test seat for other CPOL and CPHA settings. The backend Table 2 • Ports Portrayal of SPI-to-I2C Interface Plan Harbour Heading Portrayal sclk Input Input Clock from the SPI Slave. Recurrence depends upon the SPI Slave. MOSI Input SPI information input from SPI ace MISO Yield SPI information yield to SPI ace from SPI_I2C Interface ss Input Dynamic moo slave select yield flag PCLK Input Input clock (20 Mhz) PRESET_N Input Dynamic moo reset flag done Yield Dynamic tall flag demonstrates the I2C exchange is total SDA Input I2C Serial Information SCL Yield I2C Serial Clock Table 3 • SPI-to-I2C SPI-to-I2C Interface Plan Illustration I2C slave employments a custom I2C slave piece.

V. CONCLUSION

The plan in this paper actualized in fpga board which takes command and information from sender gadget (as a rule a controller) working on SPI convention and sends it to collector gadget (more often than not a fringe gadget) working on I2C convention. Hence such a plan would empower the controller to communicate with huge number of peripherals utilizing tall speed of SPI and spare on devoted pins for each fringe gadget utilizing I2C.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1582

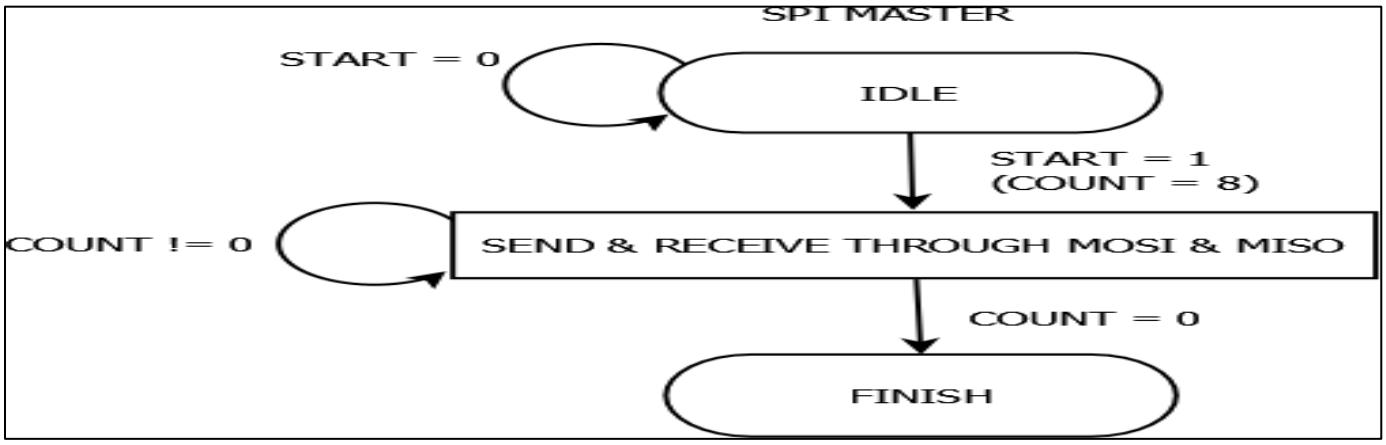

Fig 3: Flowchart for SPI Master Data

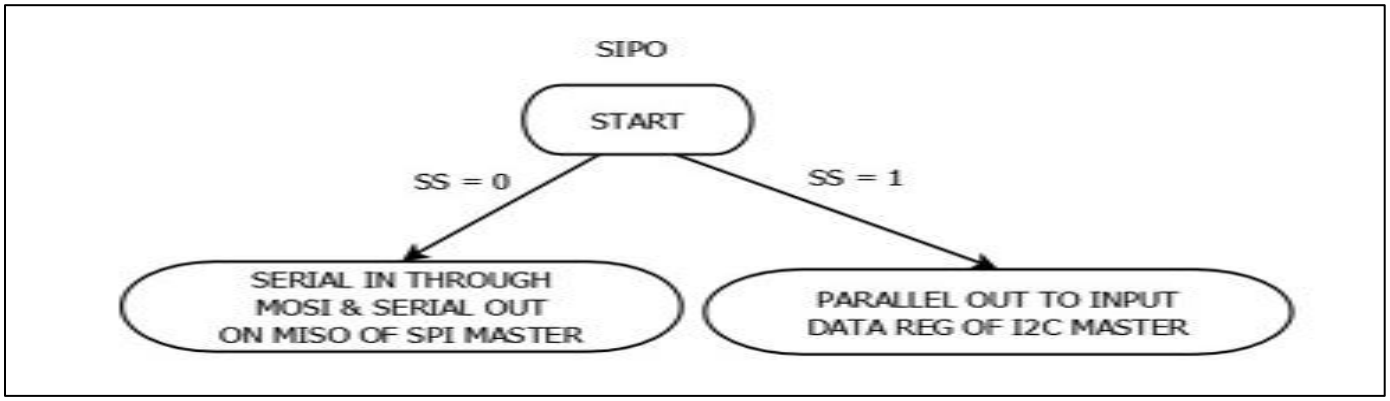

Fig 4: Flowchart for Serial in and Output Parallel

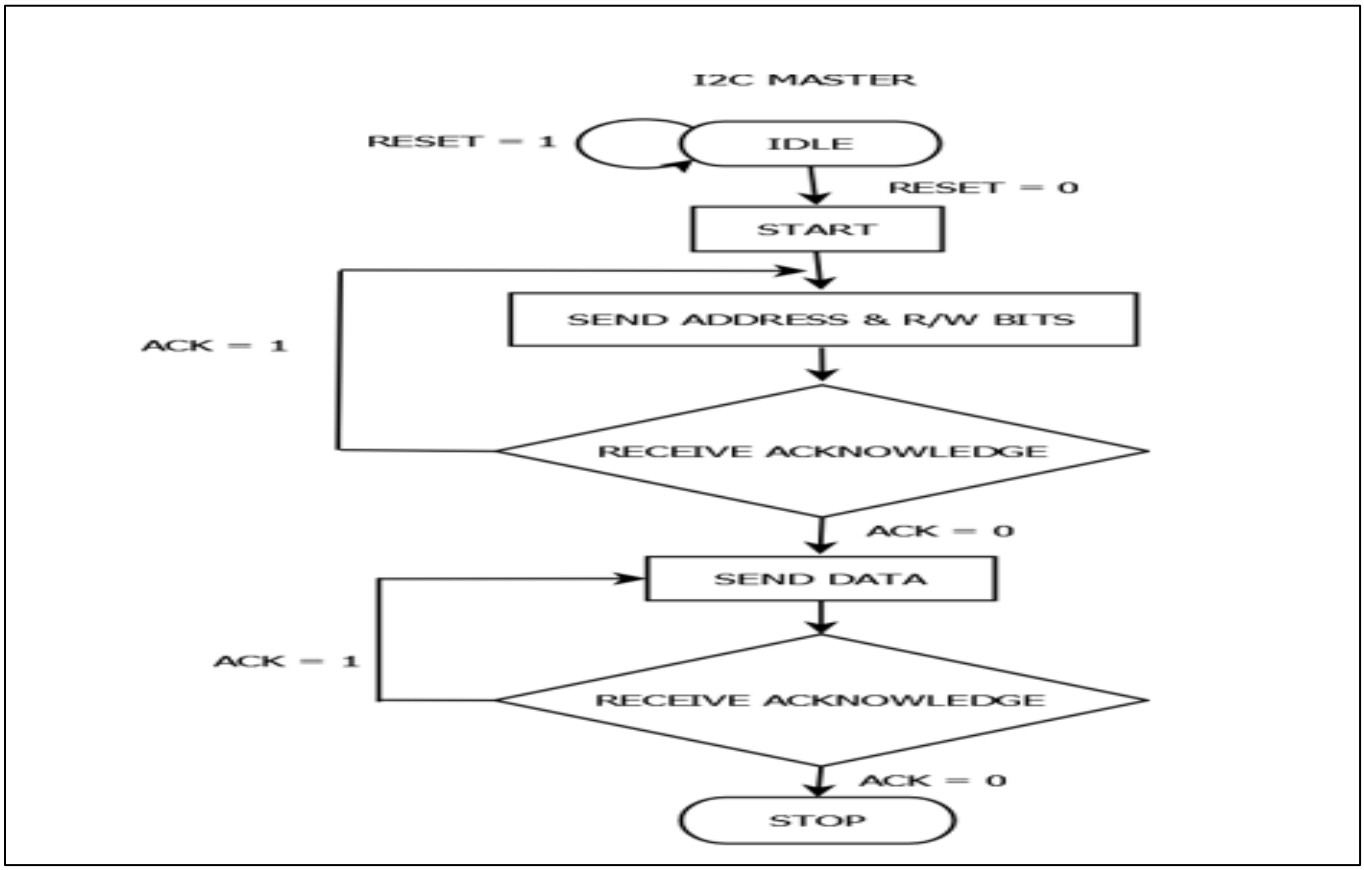

Fig 5: Flowchart for I2C Master Data

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1582

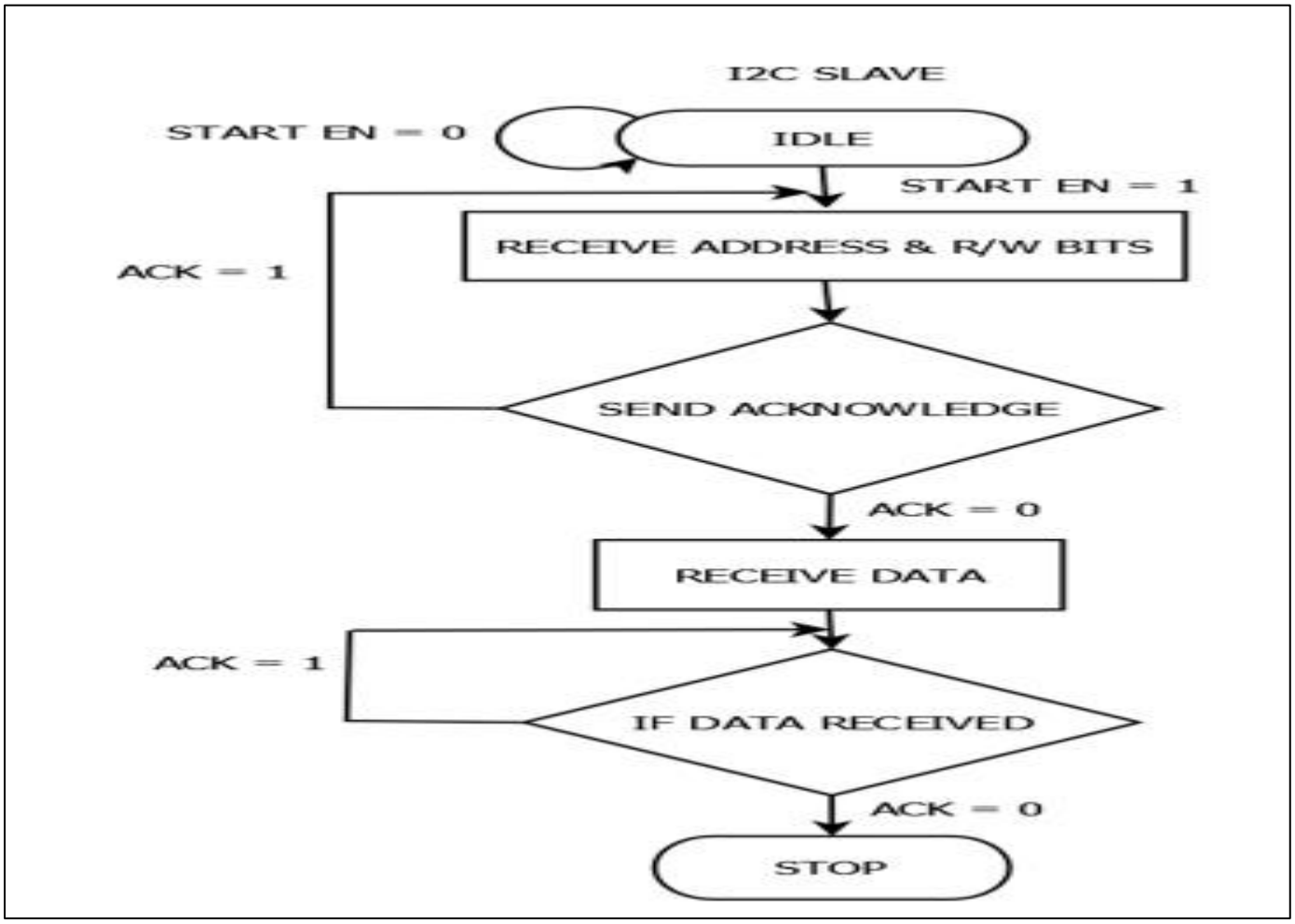

Fig 6: Flowchart for I2C Slave Data

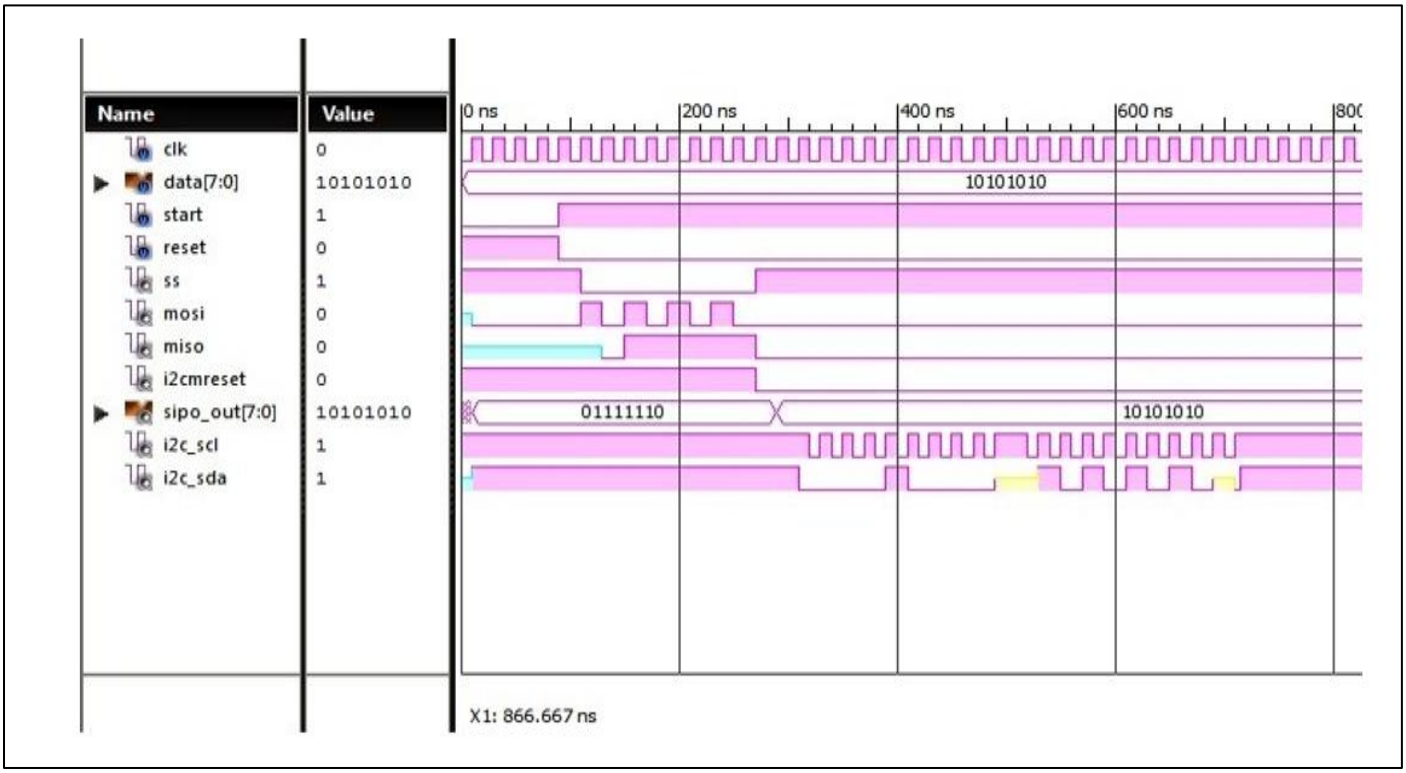

Fig 7: Simulation Result

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1582

VI. FUTURE VISION

The plan within the paper can be overhauled to bolster the communication between so numerous sorts of protocols for fastest communication. We are able amplify this to tall speed information exchange where we are able exchange the information at the speediest occurrence as conceivable. Both I2C and SPI are synchronous (there's a committed clock line), neighborhood (ordinarily not coming to exterior a single gadget or PCB), single-ended (not differential like HDMI or Ethernet), master-slave, one-to-many busses.

ACKNOWLEDGEMENT

I hereby conclude that I am perusing my P.G under JNTU affiliated college under the guidance of all my professors.

I thank each and every one in our educational organization for providing me this opportunity.

REFERENCES

[1]. Abhilash S.Warrier, Akshay S.Belvadi, Dhiraj R.Gawhane, Babu Ravi Teja K, FPGA Implementation Of SPI To I2C Bridge, International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 11, November - 2020.
[2]. M. Morris Mano, Michael D. Ciletti, Digital Design:With an Introduc- tion to verilog HDL, 5e, Pearson, 2019.
[3]. Kevsrobots. UNDERSTANDING I²C (INTER-INTEGRATED CIRCUIT): HOW IT WORKS [Cited 2023 December 22] Available at: Link
[4]. Analog. Proven Implementations of the I²C Bus [Cited 2023 December 22] Available at: Link
[5]. Microchip. Using the PIC Devices' SSP and MSSP Modules for Slave I2CTM Communication [Cited 2023 December 22] Available at: Link
[6]. S. Sarns and J. Woehr, "Exploring I2C", *Embedded Systems Programming*, vol. 4, pp. 46, Sept. 1991.
[7]. Google Scholar
[8]. *Freescale M68HC05 Microcontrollers data sheets*, October 2008, [online] Available: http://www. freescale.com.
[9]. Google Scholar
[10]. *Freescale SPI Block Guide*, October 2008, [online] Available: http://www.freescale.com/webapp/search/Serp. jsp?&QueryText=Freescale%20SPI%20Block%20G uide%20V03.06 &SelectedAsset=Documentation&QueryText=*&&f srch=1.
[11]. Google Scholar
[12]. *IC bus specification*, October 2008, [online] Available: http://www.nxp.com/acrobat_download/literature/93 98/39340011.pdf.
[13]. Google Scholar