A Novel Architecture of I²C Slave using One-Hot Encoding Technique

Devashree Mahato
Department of ECE
NIST, Berhampur, Odisha,
India-761008

Sulipta Das
Department of ECE
NIST, Berhampur, Odisha,
India-761008

Durga Prasad Dash
Department of ECE
NIST, Berhampur, Odisha,
India-761008

ABSTRACT
This paper presents a novel architecture of Inter Integrated Circuit (I^2C) slave module for embedded processor at protocol level to provide flexibility. The internals of modular description follows higher level of abstraction in Verilog Hardware Description Language (HDL) to provide a technology independent design for Field Programmable Gate Array (FPGA) implementation using Xilinx ISE 12.1. A brief contrast analysis has been carried out from logic synthesis results obtained by targeting the process technologies of 90nm, 65nm, 45nm and 40 nm individually on Xilinx FPGAs. The behavioral model has been simulated to verify the complete functionality of I^2C serial transmission protocol through the instantiation of slave module in a top-level stimulus block. One-hot Finite State Machine (FSM) encoding scheme is being adopted for slave transceiver to exhibit the I^2C cycle operation. The target device Virtex6: xc6vlx75t-3ff484 offers maximum frequency.

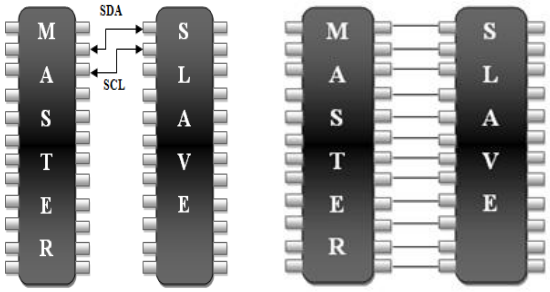
Keywords
I^2C, SDA, SCL, FPGA, FSM, One-Hot Encoding.

1. INTRODUCTION
The emerging technology in electronics has led to develop complex Systems on Chip (SoC) for diverse applications in telecommunications. The key drivers towards electronic components in telecommunication are compact components, system integration and innovative products which offer greater flexibility, reliability and wide range of connectivity. With the development of technology the board-level components can be integrated on a single chip.

In SoC design, power consumption and area are always critical constraints and at the same time high communication flexibility is required due to proliferation of communication protocols. For inter chip communication, many pins are required for inter-chip addressing, selection, control and data transfer. All the integrated components must be connected to each other and every SoC must be linked with each other in an efficient manner that allows a fast and error-free communication.

The use of SoC is to achieve high performances in terms of speed and error free communication. The most used solution for interconnecting SoC is a serial bus which presents great advantage in terms of costs and reducing the complexity of the system. Using few wires to link different peripherals implies the implementation of a simple architectural interconnectivity and consequently the cost gets reduced for the designer. To reduce the manufacturing cost of the electronic products I^2C protocol was created by Philips semiconductors in early 1980s. It is a two-wire, bi-directional serial bus and supports low to medium speed that provides a simple and efficient method of data exchange between devices [1]. It is used for short distance communication between many devices. In I^2C bus the two wires Serial Data Line (SDA) and Serial Clock Line (SCL) are used which carry addressing, selection and control. The data wire is used to carry data, while the clock wire is used to synchronize between the transmitter and the receiver. The Figure1 given below briefly illustrates the reduction of complexity in interfacing within two devices by replacing the parallel interface with I^2C protocol implementation.

This paper is organized as follows. This section explains the requirement for I^2C protocol in communication. Section 2 is all about the characteristics of I^2C bus specifications. Section 3 includes the state machine configuration of the slave device in which the one-hot encoding technique is being adopted and the interpretation of the state vectors is described. Section 4 presents the behavioral simulation results obtained and finally the conclusion in section 5.



(a) (b)

Figure 1: (a) Parallel Interface (b) I^2C Serial Interface

2. I²C BUS CHARACTERISTICS
The I^2C bus is a two-wire bidirectional serial bus comprises of SCL and SDA [1]. These two wires carry information between the devices. The bidirectional signals SCL and SDA are connected to a positive supply of voltage through pull up resistor. In I^2C bus the power consumption is very less and it offers high noise immunity. Device can be either master or slave. The master initiates the transfer, generates the clock signal and terminates the data exchange. The devices are recognised by its unique address which can be either the transmitter or the receiver. The device which is addressed by the master is the slave device.

During the high period of clock, the data on the SDA line must be stable. When high to low transition takes place on SDA line and SCL line is high then the start condition occurs.

When the bus is free it indicates that both these lines are high, the output of the device is connected to the bus. Stop condition occurs when the SDA line transits from low to high and SCL line is high and the bus is released for another transmission. But if master wishes to communicate slave again it generates start condition again as repeated start and data can be transferred again.

The data transmission takes place in byte format [1]-[4]. The data transfer starts from the master device which generates start condition and then sends the slave address. The slave device compares its own address with the seven bits address sent. If it matches, the slave considers as addressed by the master and responds to the master with an acknowledgement. When the slave doesn't acknowledge, the slave can neither transmit nor receive.

Prior to this acknowledgement a direction bit is sent i.e. read or write bit, which indicates whether the slave will read the data or write the data. If it is read operation, the slave device will send the data and it waits for the master for confirmation from the master. For write operation, the slave device receives the data from the master.

Each byte is followed with an acknowledgement bit .When the data transfer is completed, the master generates a stop condition. The fundamental aspects of the specified protocol indicate the attainment of higher throughput by minimizing the need of overhead bits for synchronization and error detection. Here, ease of error free data access is confirmed by means of active low positive acknowledgement field for short distance serial communication.

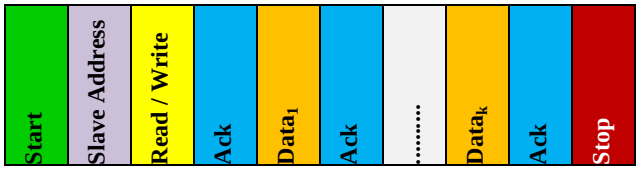


Figure 2: Frame format of I²C Protocol

3. DESIGN APPROACH

3.1 FSM Representation of I²C Slave

We have designed the slave module as transmitter and receiver. The slave module for I²C interfacing is modelled by adopting the mealy Finite State Machine (FSM) based approach to provide higher operating frequency along with efficient FPGA resource utilization. The FSM logic followed here is deterministic in nature, as the next state transition is predictable for each defined state. In mealy FSM technique, triggering a transition depends on both the received asynchronous inputs as well as the current state [5]-[10]. Each state is related with a particular set of outputs. The slave module we have considered consists of 18 states according to the fields specified in the frame format. Always the system is initialized to the idle state which forms the default value, when the system reset is high. The transition from the above default state to addr 7 state is controlled by START and STOP signals. After a complete transfer of data the current state reaches at termination stage i.e. data 0. Here, the positive acknowledgement decides the state transition from data 0 to idle state. The design flow adopts the behavioral modelling.

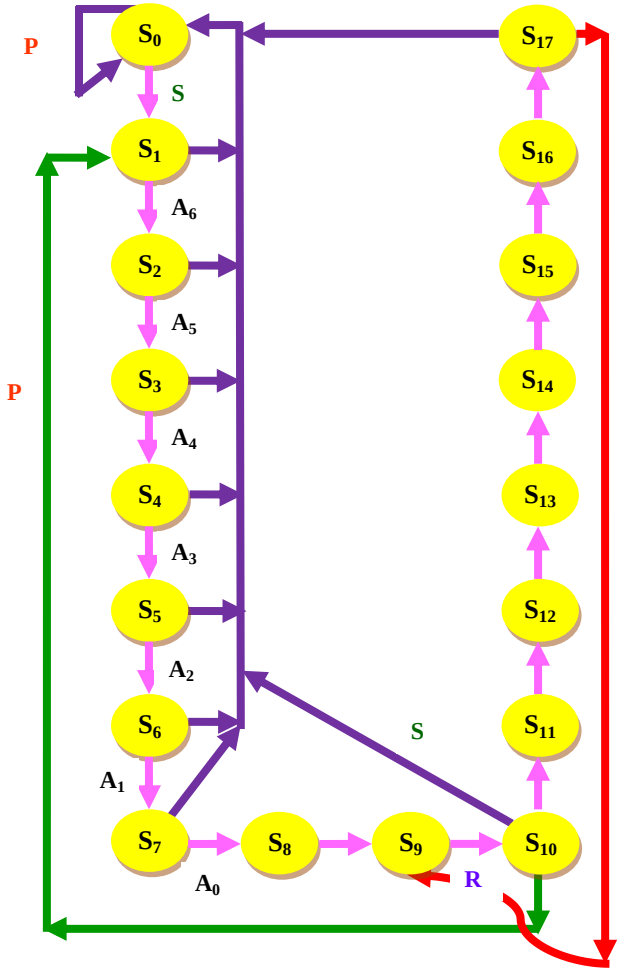


Figure 3: State diagram of the slave transceiver

3.2 Interpretation of State Vectors

The Mealy FSM for I²C slave module consists of eighteen states corresponding to each field of the frame structure as shown in Figure3. The interpretation of each state notation is presented as follows:

S_0: System reset or Idle. Until the detection of START signal, the present state continues to remain at S_0.

S_1 to S_7: Each bit of the slave address field starting from most significant bit (MSB), sequentially. If the shift register holds the next bit, then transition occurs to the next state.

S_8: Corresponds to read or write field of protocol. Here the change over to S_9 takes place directly without the requirement of any input.

S_9: Acknowledgement is received from the slave after matching the transmitted address by master with its own address.

S_{10} to S_{17}: Each bit of the data field from MSB.

If the content of shift register does not contain the next bit at a certain stage during transfer of slave address, then system is restored to idle state by default. Again, INITIATE condition is checked at S_9 to start the data transmission during S_{10}. Also STOP signal is being considered concurrently to terminate the serial communication process. If the data over SDA line is binary zero, indicates the continuity of transmission without any interruption.

3.3 One-Hot Encoding Technique

While designing a state machine to achieve complex modular functionality, the state encoding technique plays an important role. Optimization of timing is greatly affected by the proper selection of state assignment for reduction of logic levels. In one-hot design methodology, the number of flip-flops to be utilized for assertion of state vectors depends on the total number of states in FSM, which avoids the requirement of the additional logic for state decoding. In other words the processing can be made faster with less number of states due to the dependency of speed only on the number of transitions into a specific state [5]. One-hot state machines are typically faster. This technique is flexible as adding and deleting states, or changing excitation equations, and can be implemented easily without affecting the rest of the machine. Low switching activity results in low power consumption. As per our design, eighteen numbers of flip-flops are required for circuit synthesis.

4. FUNCTIONAL VERIFICATION

4.1 Synthesis Results

The encoding scheme used for FSM in this design is one hot encoding [6], [10]. Here, one code bit and one flip-flop is associated to each state. Only one state variable is used during the operation at a given clock cycle. When the transition takes place the two variables between the states gets toggled. To reduce the power dissipation this encoding scheme is used. Both the shift register extraction property and the priority encoder extraction are enabled to achieve high speed. The decoder extraction property is enabled in which inputs are all constant with distinct one hot coded value [10].

From Table1 it is observed that the design offers highest frequency of 310.887MHz for 40nm process technology. When the design is targeted on the device xc6slx4-3-tqg144 the number of BELs and LUTs allocated are 43 and 41 respectively which are same as to those of target device xc6vlx75t-3-ff484. From resource allocation point of view, the device xc5vlx30-3-ff324 requires 4 BELs and 4 LUTs more than Spartan 3E device xc3s100e-4-vq100. The clock to clock period of 3.217ns obtained in case of virtex 6 target device is minimum. The speed synthesis constraint is applied to the design to maximize the frequency of operation. From the comparison of synthesis results the combinational path delay of 0.940ns is found to be minimum on the target device xc6vlx75t-3-ff484.

Table 1. Comparison of cell usage and delay on target devices

Design Statistics / Target Device	BELs	LUTs	Flip-Flops / Latches	Combinational Path Delay (ns)	Clock to Clock (ns)
Spartan 3E xc3s100e-4vq100	56	54	43	6.224	12.238
Spartan 6 xc6slx4-3tqg144	43	41	32	5.8557	7.785
Virtex 5 xc5vlx30-3ff324	59	58	43	3.528	4.738
Virtex 6 xc6vlx75t-3ff484	43	41	32	0.940	3.217

4.2 Behavioral Simulations

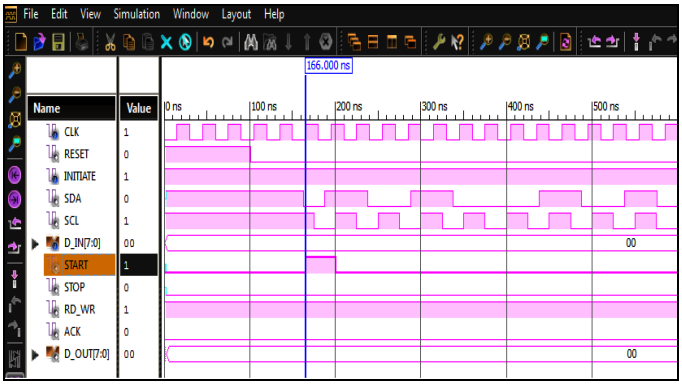


Figure 4: Detection of START condition

In Figure 4, the functional simulation shows that the system clock of the microcontroller has frequency of 33MHz. When SDA signal transits from high to low with SCL high, the START signal is generated. Initially the RESET signal remains high for 100ns and RD_WR is also high till 2475ns.The INITIATE signal remains high up to 2800ns indicating that the slave is ready for processing the data from the master. Here the 7-bit slave address 1010010 is sent on SDA line that requires 7 clock cycles of SCL. At 538ns the data on SDA line is 1 indicating the R/W field of the I^2C protocol frame format.

Figure 5: Detection of STOP condition

The Figure 5 illustrates When SDA signal transits from low to high with SCL high, and then the STOP signal is generated. The active high STOP signal holds SCL at logic 1 level till the detection of next START condition (which is generated at 2048ns). Here the slave again acts as a transmitter which is confirmed from RD_WR signal. From 2125ns on time scale, the slave address is again transmitted in 7 clock cycles of SCL. But RD_WR is made active low to enable write operation hence master becomes the transmitter to send the data followed by the acknowledgement bit. The next field with zero on SDA line indicates the continuity of transmission process.

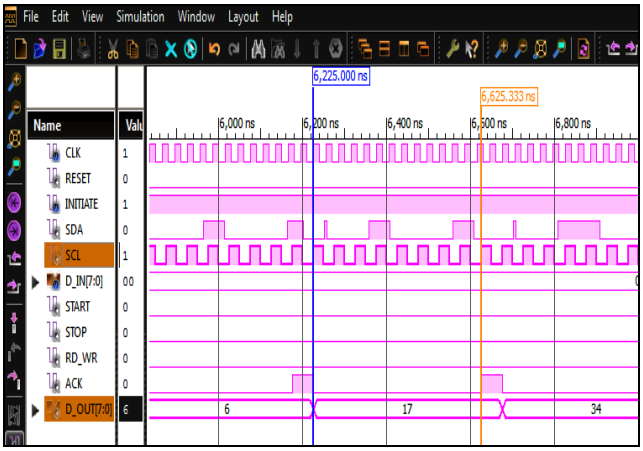


Figure 6: One byte Data transfer

In Figure 6, given above explains that one byte data is sent from master to slave. The next bit on SDA represents positive complete byte transfer. The data 00010001 is sent successfully through acknowledgement. Then the ACK signal goes high after a SDA line.

Figure 7: Acknowledgement for validity of data

From the simulation result as per illustrated in Figure7, the acknowledgement for validation about complete data is obtained. Data transfer is controlled by the two signals namely ACK and STOP.

4.3 Analysis

The Figure 8 illustrates that Virtex 6 gives the maximum frequency i.e. 310.887 MHz as compared to the other devices of FPGA family. In case of Virtex 5, the parameter is found to be 211.077MHz and for Spartan 6 its value is 128.455MHz, whereas the least maximum frequency is offered by Spartan 3E that is 81.713MHz. This result is obtained from the synthesis reports by targeting the design on each Xilinx FPGA device.

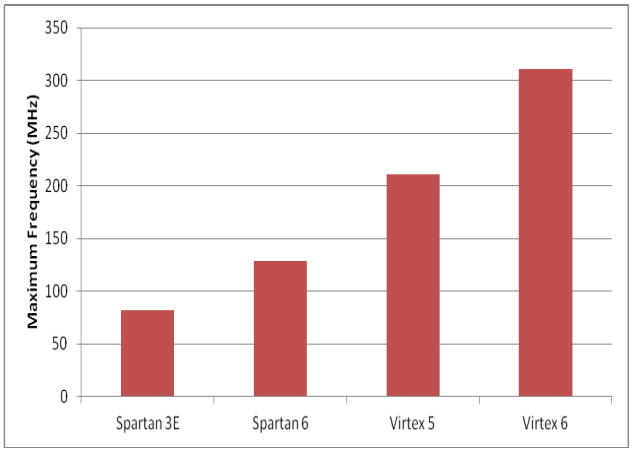


Figure 8: Comparison of Maximum Operating Frequency

5. CONCLUSION

We have developed an I^2C slave module for serial data transfer at protocol level for reusability. The FPGA based I^2C protocol implementation is preferred over microcontroller interfaces due to less complexity, portability along with faster mode of operation. Using one-hot encoding mechanism, the logic complexity gets decreased associated to each state. By issuing only two signals (SCL and SDA) information exchange occurs between the two devices. This module can be utilized for instantiation in multi master and multi slave environment.

6. REFERENCES

[1] Philips Semiconductors, "*The I^2C-Bus Specifications*", version 2.1, January 2000.

[2] P.Venkateswaran, Madhumita Mukherjee, Arindam Sanyal, Snehasish Das and R.Nandi, "Design and Implementation of FPGA Based Interface Model for Scale-Free Network using I2C Bus Protocol on Quartus II 6.0*", International Conference on Devices for communication 2009*.

[3] A. K. Oudjida, M. L. Berrandjia, R. Tiar, A. Liacha, K. Tahraoui, "FPGA Implementation of I2C & SPI Protocols:a Comparative Study", *16th IEEE International Conference on Electronics, Circuits, and Systems*, December 2009, pp. 507-510.

[4] Ramesh Bhakthavatchalu, Deepthy G R, Vidhya S and Nisha V, "Design and Analysis of Low power Open Core Protocol Compliant Interface using VHDL", *International Conference on Emerging Trends in Electrical and Computer Technology*,23-24 March,2011

[5] Steve Golson, "State machine design techniques for Verilog and VHDL", *Synopsys Users Group Conference*,1994

[6] Wolfgang Grieskamp, Yuri Gurevich, Wolfram Schulte and Margus Veanes, "Generating Finite State Machines from Abstract State Machines", *International Symposium on Software Testing and Analysis*, July 2002, Vol.27, No. 4, pp. 112-122.

[7] Bijoy Kumar Upadhyaya and Salil Kumar Sanyal, "Design of A Novel FSM based Reconfigurable Multimode Interleaver for WLAN Application", *International Conferences on Devices and Communications*,24-25 February 2011

[8] Nrusingh Prasad Dash, Ranjan Dasguptay, Jayakar Chepadaz and Arindam Halder, "Event Driven Programming for Embedded Systems - A Finite State Machine Based Approach", *The Sixth International Conference on Systems*, 23-28 January, 2011

[9] M. Morris Mano, *Digital design*, Prentice Hall, 2007

[10] Samir Palnitkar, *Verilog HDL-A Guide to Digital Design and Synthesis*, Sun Soft Press, 1996