Finite State Machine Serial Adder

Akshatha A Mallya[1], Anjali Shetty B[1], Floyd James Sequeira[1], Srigowri Hebbar[1], Divyesh Divakar[2]

[1]*Third Year, Electrical & Electronics Engineering Department, Canara Engineering College, Mangaluru, Karnataka*

[2] *Assistant Professor, Electrical & Electronics Engineering Department, Canara Engineering College, Mangaluru, Karnataka*

***Abstract*— Logic design is in itself bifurcated to- Combinational and Sequential circuits. The later has memory and former doesn't, so in an advent effort to incorporate memory into a combinational circuit brought in the concept of Finite state machine serial adder.**

***Keywords*— D-latch, Finite state machine, Mealy Model, Multisim, Serial adder.**

I. INTRODUCTION TO FINITE STATE MACHINE

A finite state machine can be represented by a state transition table or a state diagram. There is often a fixed start state which is the initial state of the Finite State Machine (before any input has been read). Thus a finite state machine (FSM) is a model describing the behavior of a finite number of states, the transitions between those states, and actions [1].

II. SERIAL ADDER

The serial binary adder or bit-serial adder is a digital circuit that performs binary addition bit by bit. The serial full adder has three single-bit inputs, two for addition and one for carry in(C-in). There are two single-bit outputs for the sum and carry out(C-out). The C-in signal is the previously calculated C-out signal. Adding each bit, lowest to highest, one per clock cycle, performs the addition [2]. Fig 1 shows a basic structure of a FSM serial adder.

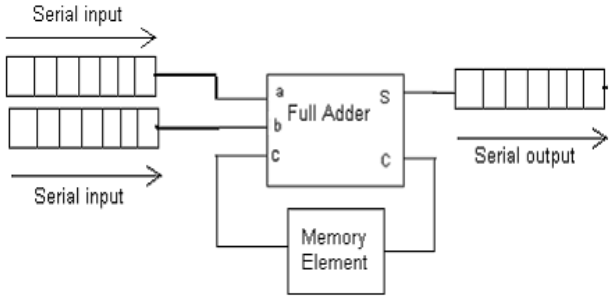

Fig 1: Block diagram of a serial adder[2]

The design is based on Mealy model. Let us consider two states, G & H i.e. when carry is generated we take H state & when carry is zero we take G state. A & B are taken as the inputs to the serial adder. Table 1 shows the state table of the serial adder [2].

Table 1: STATE TABLE of SERIAL ADDER[2]

Present state	Next state				Output *s*			
	ab =00	01	10	11	00	01	10	11
G	G	G	G	H	0	1	1	0
H	G	H	H	H	1	0	0	1

Based on the state table we can construct the state diagram. The state diagram is as shown in Fig 2. As observed in the state figure as long as the there is no carry generated, it stays in state G. but if there is a carry generated, it immediately moves to state H. In this state carry is added to the sum. When sum of A & B does not create a carry it moves back to state G.

Serial adder requires simple circuitry as compared to parallel adder, so causes of simple circuitry thus gives low speed and performs bit-by-bit operation [3].

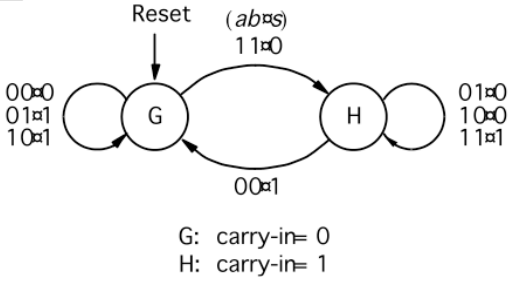

Fig 2: State Diagram for serial Adder.[2]

III. SIMULATION IN MULTISIM

Multisim is a simulation software developed by National Instrument. The Option on choosing this software is purely based on the simplicity of use & understanding of the software. As for the simulation part, the software gives a wide range options. The simulation is done based on real time & therefore it is easier to debug.

Simulation was done in two parts, a basic FSM serial adder which has no storage of its input data or its output & another with shift registers, which acts as dual purpose, i.e. input & output storage.

IV. SIMULATION 1 (FSM SERIAL ADDER)

In this simulation we built a full adder using basic gates and observed the sum via an LED at the output. The two inputs are

given as A & B. The carry generated is passed to a D-latch. The D-latch is a memory element. Once a carry is developed it is stored in the D-latch & kept for the new set of values. The output of the D-latch is connected to the 'C-in' of the full adder. The clock pulses are provided via a push button to the D-latch. The circuit rigged up in Multisim is shown in Fig 3. When two inputs are given to the adder, a sum is generated. Simultaneously if there is a carry generated, it will be stored in the D-latch for the next cycle of values. If the carry generated is directly coupled to the 'C-in' of the full adder carry, which has to be added to the new set of values, will be added to the current set of values. In general, the D-latch provides a delay for the carry so that it is added only to the new set of values.

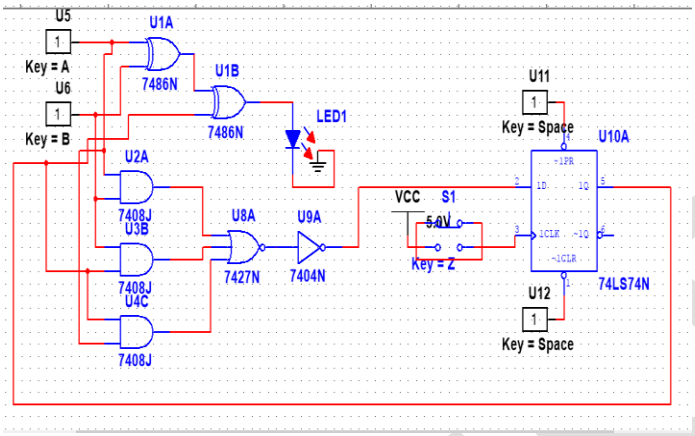

Fig 3: Simulation of FSM serial adder in Multisim.

The draw back with this circuit is that it is completely manually operated, which means we have to manually enter a set of data bit by bit, manually apply clock pulse after every addition & so on. It becomes difficult to use or in other words, it isn't user friendly. The next simulation takes care of few of the disadvantages of the current simulation.

V. Simulation 2 (FSM Serial Adder with Storage)

The circuit simulated is pretty much the same, as done previously but with few modifications. The simulation is shown in fig 4. Firstly shift registers are added to the input side if the serial adder, hence making it easier to store data. The output of the adder, which is the sum, is too stored in a shift register, the very same shift register that stores the input initially. This is done so as to save space & it's economical in the long run. In the simulation, the shift register is basically a cascade of D latches. Here two types of clocks used i.e. manual clock using push buttons & an actual clock signal (based on a given frequency). To store data the push button is used, which helps store the data in the shift register (this process is done manually). Once the inputs are stored, clock signal is applied & the inputs are passed into the adder, bit by bit. At the carry side, we have cascaded two D latches so as to provide sufficient delay while adding the carry to the next cycle.

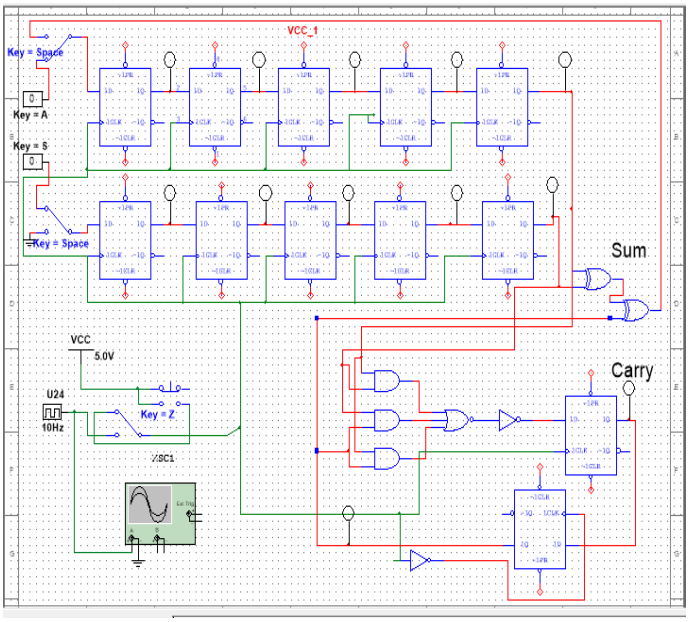

Fig 4: Simulation of FSM serial adder with storage in Multisim.

During positive pulse, the bits are added, and stored. If carry is generated it is stored in the first d latch. During negative pulse the carry is passed into the second d latch where it is connected to the adders 'C-in'. Again at positive pulse, new set of inputs is passed into the adder where the previously generated carry is added. This process continues for n bit of values. In case of the simulation that we have done, it is set for 4-bit addition.

The flaw with this particular simulation is that even though it is a 4-bit addition, the output could be of 7, 8 or even 10 bit due to no of clock pulses, when the actual value should be only of 4 bits. To elaborate, addition of 0110 & 0010 should give an output of 1100, but that happens only if 4 clock pulses are given. If 7 clock pulses are given the output will be 0001100, which still give the same value but with unwanted extra bits. Another flaw is that there isn't a provision for carry to be added externally. This particular instant might come into use when carry needs to be added at the very first instant of addition of first two bits.

VI. Conclusion

To build a circuit that can add any two binary numbers using the sequential method that humans use, a FSM serial adder is the only solution. For any digital circuit performing operations such as multiplication, subtraction and division, adders are the basic components used extensively. It is the demand of the day to improve the performance of digital adders for the execution of binary operations inside a circuit. The main aim of designing the bit serial adder is to perform one bit at a time, using the first bit operation results to influence the processing of subsequent bits. It reduces the amount of hardware required as it passes all the bits in the same logic. This approach needs only (1/n) part of hardware when compared to the n-bit parallel adders.

ACKNOWLEDGMENT

We would like to thank the Management and the Principal of our Institution for providing us the facility to work on this paper. The authors also gratefully acknowledge Dr. Rajalakshmi Samaga B L, Head of Electrical and Electronics Engineering Department, for the moral support throughout the work.

REFERENCES

[1] Steven T. Karris, Introduction to State flow® with Applications, Orchard Publications.

[2] S.Shirani, www.ece.mcmaster.ca/~shirani/2di4/chapter8p2.pdf

[3] Ashivani Dubey and Jagdish Nagar, Acropolis Institute of Technology and Research, Indore, India, "Comparison between Serial Adder and Parallel Adder" IJSERT September, 2013