High Speed Universal Shift Register

Vikkram Srinivasan[1], Sashanth O[2], Vignesh M[3], Mrs.Saraswathy B[4]
[1, 2, 3] Dept of ECE
[4]Assistant Professor, Dept of ECE
[1, 2, 3, 4] Meenakshi Sundarajan Engineering College

Abstract- *As technologies like AI are evolving much faster, there is need for improving the performance of the computers for fast and effective computing. Fast computing involves loading and unloading data faster to the register. Universal shift registers are used in every computer for storing data, also performing parallel to serial operations and vice versa. Usually universal shift registers are designed using flip flop which can store a 1-bit data. When it comes to storage devices, speed and area are very important factors. The speed and area can be improved by using latches in place of flip flops since latches are level triggered and uses less logic gates to design compare to flip flop. So the idea is to use latches in place of flip flops to simulate and see the behaviour of latches and analyse its performance which will be compared with its equivalent flip flop counterpart to have a conclusion.*

Keywords- Universal shift register, latches, flip flops, FPGA, AI, level triggered

I. INTRODUCTION

Universal shift register is a type of register which can be used to perform all kind of shifting operations such as

1. Left shift
2. Right shift
3. Parallel loading
4. Memory operation

The main application of universal shift registers will be to perform serial load parallel out operation and vice versa. Generally USR are designed using flip flops and MUX. For example, a 4-bit universal shift register is designed by using four D – flip flops and four multiplexers. The reason why flip flops are used in registers are because they are edge triggered meaning the value will not change until the next clock pulse arises of falls. This allows to store the value for a certain amount of time and there will be no violations on timing. Latches which also function same as flip flop and also the building block for designing flip flops, are faster in storing and loading values. Latches are basic storage element which also can be used to store a 1 – bit value. One of the main problems with latches are they are very sensitive and the values can change instantaneously whenever the input value changes. Changing values continuously will lead to the violation of setup time and hold time and therefore the circuit will become unreliable.

But in recent years, many research had been made to make these latches more reliable and conventional to apply on designing of storage blocks, which lead to the discovery of pulsed latches. Pulsed latches works exactly like normal latches but the only change will be the clock pulse for the enable. Clock pulses with very low duty cycle seem to provide a solution for using latches.

II. LITERATURE REVIEW

So far there has been many attempts made to improve the performance of USR and one good thing is there are many options to improvise on it. But generally cost and liability also comes into factor while using various other technologies. As per theory, if the number of transistors are reduced, then it will consume less power and area. But recent FPGA's are very good in optimization so small change in transistors will not produce much greater result. Which motivate us to design a 16-bit USR to see better results. The FPGA that we try to simulate plays a massive role in terms of calculating result metrics such as area and delay so choosing right FPGA will also a bit challenge. Based on research, we see that this exact idea is implemented on shift registers and could able to produce positive results.

III. PROBLEM STATEMENT

A universal shift register is a type of register which has both right and left shift and parallel loading operations as well. Universal shift registers uses flip flop as storage element. In order to improve the speed of the universal shift registers, latches are used in place of flip flops to reduce the delay. Latches are generally much faster in terms of loading and storing operations and has less gates compare to flip flops as well. In order to use latches in registers, non - overlapping pulsed latches are used to avoid the timing problem. The timing problem will happen while using flip flop this is because the flip flop is edge triggered and therefore it will be active only either at positive edge or negative edge of the clock. So the idea is analyse how much the delay will reduce

by using pulsed latches and how much the area will reduce as well. As per the theory, flip flop is done using two D – latches is form of master – slave configuration. As a result the area taken by the flip flop will be double the space compare to latches. Also the delay will be high since there will be double the number of logic gates used for flip flop. First analysis is by comparing 4 – bit USR using flip flop and by using pulsed latches and then by using the 4 – bit module, a 16 – bit USR is designed as well for both latches and flip flop to compare between them. For analysing area and delay, Xilinx 13.3 ISE is used and the coding of modules is done using Verilog. The technology family used for the simulation is Spartan 6 and the device used is XC6SLX25.

WORKING PRINCIPLE

LATCHES AND FLIP FLOP

Latches are basic storage element which can store a 1 – bit value. The function of latch is simple. When enable is low the circuit will be in memory state where it will hold the value and does not change for changes in D input. When enable is high then based on the value of the input D, the output Q will change. Here as could see the value changes as soon as the enable is held high so there is no control when the enable is high. When the input changes its value very quickly, it may violate setup time and hold time principles which will result into uncertainty.

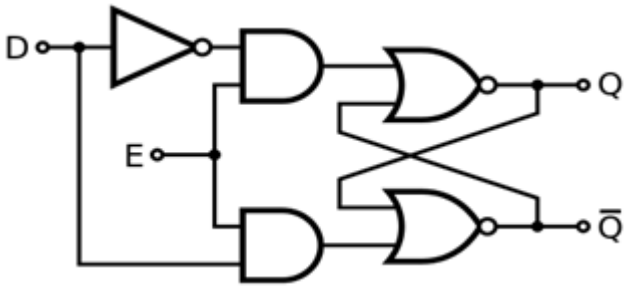

Fig -1: Circuit diagram for D-latch

Flip flops on the other hand is also a storage element which stores a 1 – bit value. The truth table of flip flop is shown below.

D	CLOCK	Q	$\overline{Q}$
0	0	Memory	Memory
0	1	0	1
1	0	Memory	Memory
1	1	1	0

Table -1: Truth table of flip flop

When clock is low the master latch will be off and the slave latch will be active which implies that the circuit will not allow any changes and store the previous value. When clock is high then master latch will be on and slave latch will be off, then there will be changes in master latch based on the value of input, but the changes will be reflected to the output only when the slave latch is on, that is when the clock pulse becomes low again.

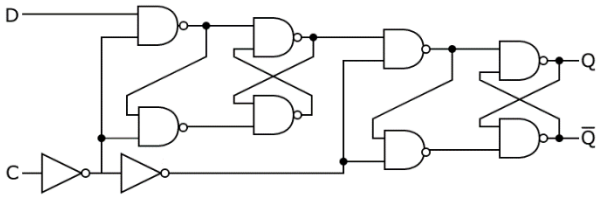

Fig -2: Circuit diagram of master slave flip flop

DELAYED PULSED LATCHES

Latches will change the value as soon as the input changes which came to be disadvantage in terms of designing any shift registers as every latches changes its value instead of changing in one of the laches. In order to prevent this, delayed and non-overlapping clock pulses are given as clock signal for latches so that there won't be any timing problem while shifting the data. The duty cycle of the clock pulse is kept very low this is because the latches are level triggering and hence it is safe to keep the duty cycle as small as possible.

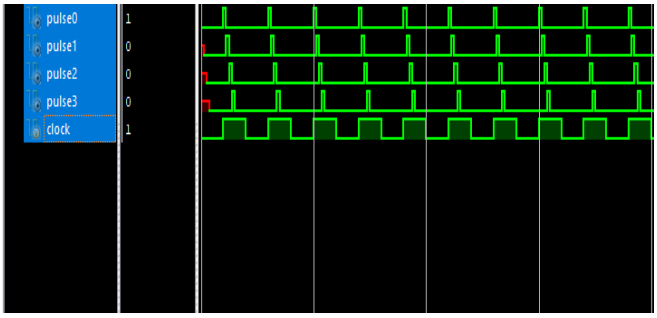

Fig -3: Non overlapping clock pulses

DESIGN AND IMPLEMENTATION OF UNIVERSAL SHIFT REGISTER

I. 4-BIT USR USING LATCHES

A 4-bit USR requires four 4×1 multiplexers (MUX) and four latches. In this model four separate clock pulse will be given. These clock signals are nothing but the non-overlapping pulsed clock signals. These clock signals are made in such a way that only one pulse will be high at a time. Two selection lines S_1 and S_0 will select the type of operation to be performed and the output of the MUX will be connected

to the input of the latch. The collective output of latches are the output of the USR.

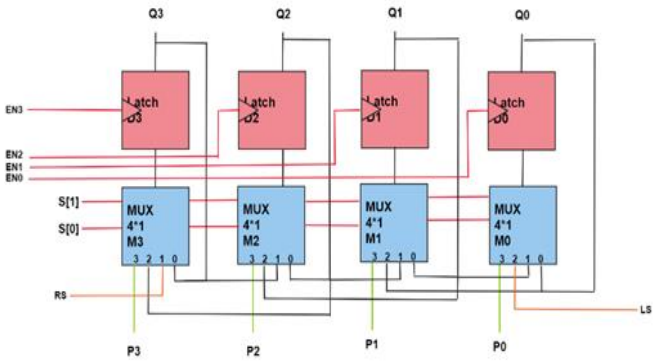

Fig -4: 4-bit USR using latches

P_3, P_2, P_1, P_0 are parallel inputs in which the USR will perform parallel loading operation when selection line becomes 11. LS represents left shift input value given at LSB and RS is right shift input which is given at MSB.

S1	S0	Operation
0	0	Memory
0	1	Right shift
1	0	Left shift
1	1	Parallel load

Table -2: Truth table of USR

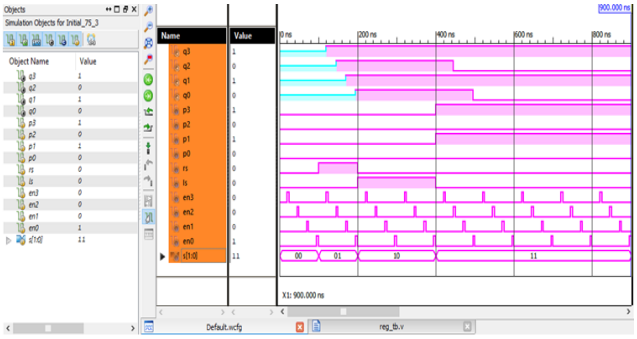

Fig -5: Simulation of 4-bit USR using latches

II. 4-BIT USR USING FLIP FLOPS

The flip flop module is also done similar to that of latch moule expect single clock pulse is enough to give as clock pulse for all the flip flops. The flip flops used here are made up of two latches in form of master slave configuration. For designing USR the same latch module will be used but instead there is one more module designed for flip flop by using latches. This flip flop will be the storage element and will store values based on the operations

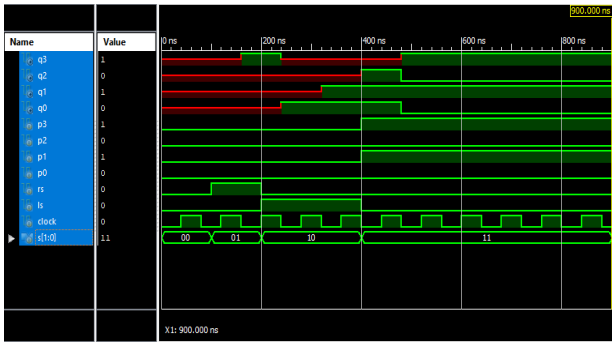

Fig -6: 4-bit USR using flip flops.

III. 16-BIT USR USING LATCHES

A 16-bit USR is created by following the same procedure as for designing 4-bit USR. Here sixteen MUX and sixteen latches are used. The 4-bit USR module is used here to create four 4-bit USR and connect each other by using combinational circuits to function as a 16-bit USR. The circuit diagram for 16-bit USR is shown below.

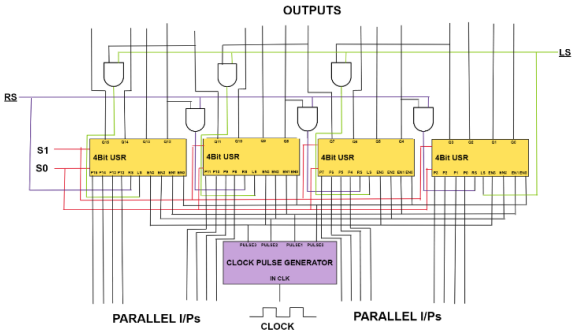

Fig -7: 16-bit USR using latches

The clock pulse will be same for all the 4-bit USR which is nothing but four non-overlapping clock pulses given for 4-bit USR before.

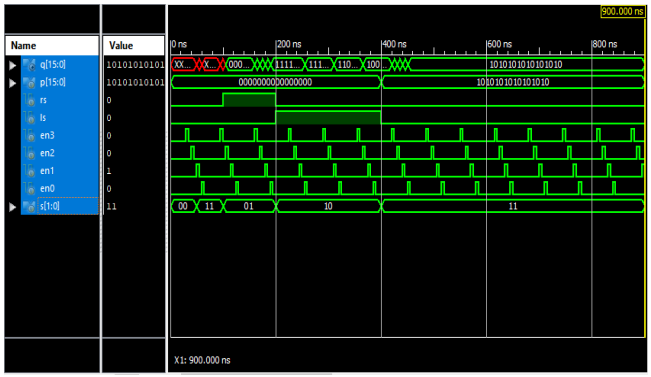

Fig -8: Simulation of 16-bit USR using latches

IV. RESULTS AND DISCUSSIONS

For analysing how fast the USR is working, we need to calculate the delay of the circuit. For every logic gate, there will be propagation delay which delays the response of the circuit. The two main parameters used for the analysis are:

- Overall delay
- Area

All these analysis are done using Xilinx 13.3 version.

Theoretically speaking, the delay should be reduce by half for USR using flip flop. This is because number of logic gates used will be half the number of logic gates used for creating latches. But the FPGA will optimize the given module (especially for flip flops as it is mostly used). So the expected output may not be exactly same as that of theoretical value. Spartan 6 FPGA is known for its optimization and hence it is hard to tell how much changes occurred for smaller modules like 4 – Bit USR. That's the reason why 16 – Bit USR is also implemented to see the results. The device used for simulation is XC6SLX25.

S.NO	Type	No of occupied slices (using latches)	No of occupied slices(using flip flop)
1.	4-bit USR	5	5
2.	16-bit USR	15	18

Table -3: Area analysis

This tells that the latches will reduce the area occupied in slices.

The delay analysis is shown below

S.NO	USR type	Delay using Latches (in ns)	Delay using flip flops (in ns)
1.	4-bit USR	12.542	16.382
2.	16-bit USR	27.564	52.335

Table -4: Delay analysis

From this we can clearly see that the delay is reduced which implies the speed will be improved. More significant changes in delay is observed for 16 – Bit USR.

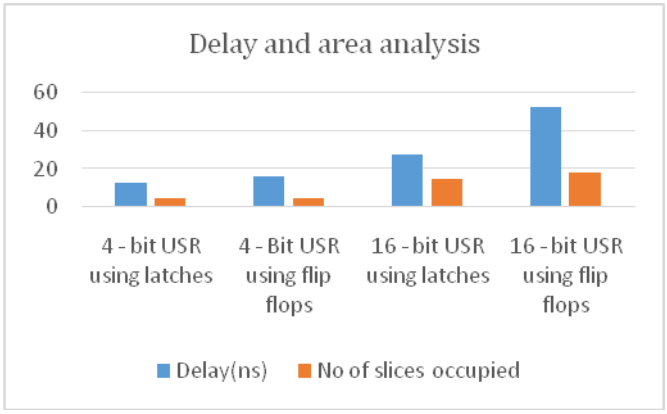

Chart -1: Delay and area analysis

V. CONCLUSION

In this era where there is necessity for high speed computing, the chip designing becomes as sophisticated as ever. By using latches in place of flip flop for designing USR, the number of occupied slices reduces which can be significant for designing purpose. But the most important thing is that the delay is drastically reduced especially for 16 – bit USR. By which we can say that the delay difference will be even high for 32- bit and 64 – bit. Even though latches are not ideal storage element, by using non – overlapping pulses it can be functional. Even though speed is improved, we can improve the performance further by many ways and can extend the research further by testing on other FPGA's and can be implemented on other storage elements as well.

REFERENCES

[1] Markovic D., Nikolic B., Brodersen R.W., Analysis and design of low-energy flip-flops, Proceeding of International Symposium on Low Power Electronics and Design, 2001, 6-7 Aug. 2001, Pages: 52 -55.

[2] H. Partovi et al., "Flow-through latch and edgetriggered flip-flop hybrid elements," IEEE Int. SolidState Circuits Conf. (ISSCC) Dig. Tech. Papers, pp.138–139, Feb. 1996.

[3] RenganayakiG ,Jeyakumar.V, ―Design of an Efficient Low power Shift Register using Double EdgeTriggeredFlip-flop,‖IEEE J. Solid-State Circuits, vol. Vol. 2, Issue 1, January 2014.

[4] SalimPalnitkar, "Verilog HDL: A guide to digital design and synthesis" Sunsoft press 1996

[5] Saranya.M, V.Vijayakumar, T.Ravi, V.Kannan, ―Design of Low Power Universal Shift Register,‖IEEE, Solid-State Circuits,Vol. 2 Issue 2, February 2013