JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

IMPLEMENTATION OF LOW AREA DELAY PULSED LATCHES FOR SHIFT REGISTER USING KOGGE STONE ADDER

DANGETI SANNUTHI
Research Scholar, Department of ECE,
International School of Technology and Sciences (Women),
Affiliated to JNTUK, NH-16, East-Gonagudem,
Rajanagaram, Rajamahendravaram,
Andhra Pradesh - 53329, India.

Shri. R. VINAY KUMAR, M. Tech.,
Assistant Professor, Department of ECE
International School of Technology and Sciences (Women),
Affiliated to JNTUK, NH-16, East-Gonagudem, Rajanagaram,
Rajamahendravaram,
Andhra Pradesh - 53329, India

***Abstract*:** This project proposes delay efficient architecture for shift registers using pulsed latches instead of flip flops. Area and power can be reduced greatly by using latches then flip flops. The timing problem exhibited by the latches is reduced by taking necessary delays in pulses for latches. This includes a counter for generating pulses with delays. For obtaining these delays counter has to incremented by 1. The proposed Kogge stone architecture reduces the delay to maximum extent, and produces numerous variations between conventional adder architecture.

Keywords: Flip flops, Latches, KOGGE stone adder.
Software used: Xilinx 12.3
Language used: Verilog HDL.

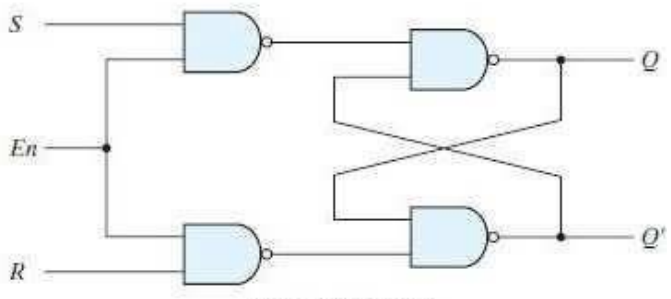
I.INTRODUCTION

Dynamic power is consumed across all elements of a chip. The clock network is one of the large consumers of dynamic power. Therefore, reducing power in the clock network can impact the overall dynamic power significantly. Designers already use a variety of techniques to reduce the clock power using smaller clock buffers. Even with these techniques, the dynamic power of clock network can be large since registers are used as state elements in the design. In general, a flip-flop is used as the register. (2) The digital logic circuits whose outputs can be determined using the logic function of current state input are combinational logic circuits; hence, these are also called as time independent logic circuits. (3)The digital logic circuits whose outputs can be determined using the logic function of current state inputs and past state inputs are called as sequential logic circuits. These sequential digital logic circuits are capable to retain the earlier state of the system based on the current inputs and earlier state. Hence, unlike the combinational logic circuits, these sequential digital logic circuits are capable of storing the data in a digital circuit. The sequential logic circuits contain memory elements.

II.LITERATURE REVIEW

2.1 S R latch

The *SR* latch with two cross-coupled NAND gates. It operates with both inputs normally at 1, unless the state of the latch has to be changed. The application of 0 to the *S* input causes output *Q* to go to 1, putting the latch in the set state. When the *S* input goes back to 1, the circuit remains in the set state. After both inputs go back to 1, we are allowed to change the state of the latch by placing a 0 in the *R* input. In comparing the NAND with the NOR latch, note that the input signals for the NAND require the complement of those values used for the NOR latch.



(a) Logic diagram

Fig 2.1: SR-Latch

Truth table is as follows:

En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

2.2 D-Latch:

One way to eliminate the undesirable condition of the indeterminate state in the SR latch is to ensure that inputs S and R are never equal to 1 at the same time. This is done in the D latch; shown this latch has only two inputs: D (data) and En (enable).

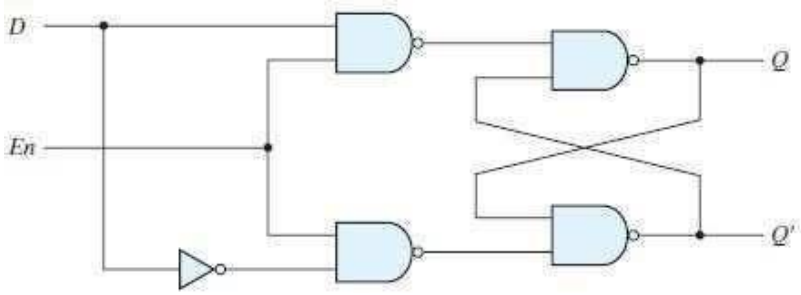


Fig 2.2: D-Latch

Truth table is as follows:

En	*D*	Next state of *Q*
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state

2.3 JK-Latch:

The JK latch is much less frequently used than the JK flip-flop. The JK latch follows the following state table:

JK latch truth table			
J	K	Q_{next}	Comment
0	0	Q	No change
0	1	0	Reset
1	0	1	Set
1	1	$\overline{Q}$	Toggle

Flip-flop circuits are constructed in such a way as to make them operate properly when they are part of a sequential circuit that employs a common clock. The problem with the latch is that it responds to a change in the level of a clock pulse.

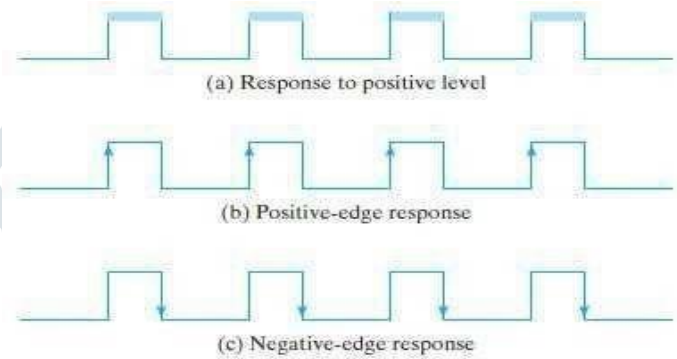


Fig 2.3: Positive and Negative clock edge cycle

2.4 SR-Flip-flop:

The **SR flip-flop**, also known as a *SR Latch*, can be considered as one of the most basic sequential logic circuit possible. This simple flip-flop is basically a one-bit memory bi-stable device that has two inputs, one which will "SET" the device (meaning the output = "1"), and is labeled S and another which will "RESET" the device (meaning the output = "0"), labeled R.

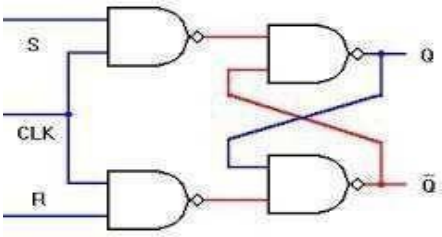


Fig2.4: SR-flip-flop circuit

2.5 D Flip-flop:

The D flip-flop is widely used. It is also known as a "data" or "delay" flip-flop. The D flip-flop captures the value of the D-input at a definite portion of the clock cycle (such as the rising edge of the clock).

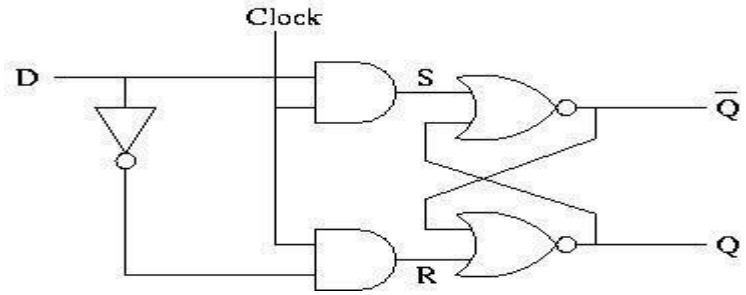


Fig2.5: D-Flip-flop

2.6 JK Flip-flop:

The **JK flip flop** is basically a gated SR flip-flop with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs S and R are equal to logic level "1". Due to this additional clocked input, a JK flip-flop has four possible input combinations, "logic 1", "logic 0", "no change" and "toggle". The symbol for a JK flip flop is similar to that of seen in the previous tutorial except for the addition of a clock input.

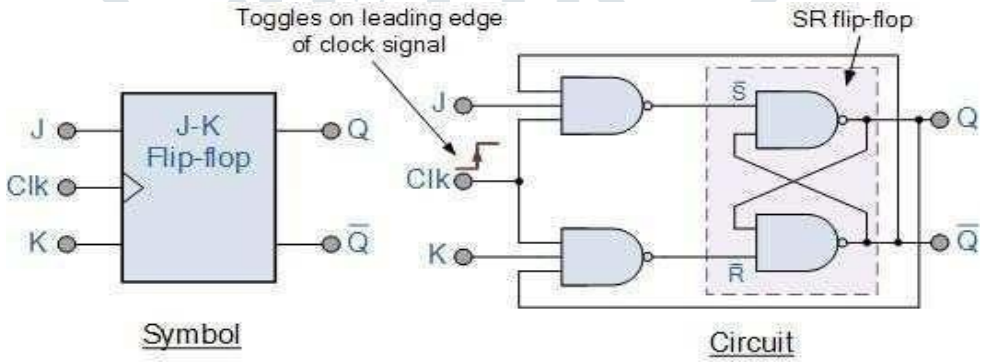


Fig2.6: JK Flip-flop Circuit and Logic Symbol

2.7 The Master-Slave JK Flip-flop:

The **Master-Slave Flip-Flop** is basically two gated SR flip-flops connected together in a series configuration with the slave having an inverted clock pulse.

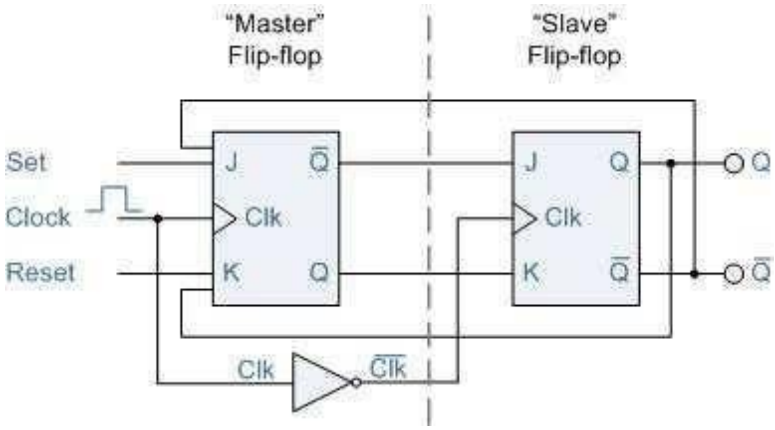


Fig2.7: Master-Slave JK Flip-flop Symbol

The input signals J and K are connected to the gated "master" SR flip flop which "locks" the input condition while the clock (Clk) input is "HIGH" at logic level "1". As the clock input of the "slave" flip flop is the inverse (complement) of the "master" clock input, the "slave" SR flip flop does not toggle. The outputs from the "master" flip flop are only "seen" by the gated "slave" flip flop when the clock input goes "LOW" to logic level "0". **Master- Slave JK Flip flop** is a "Synchronous" device as it only passes data with the timing of the clock signal.

2.8 Pulse Latch:

The pulsed latch requires pulse generators that generate pulse clock waveforms with a source clock. The pulse width is chosen such that it facilitates the transition.

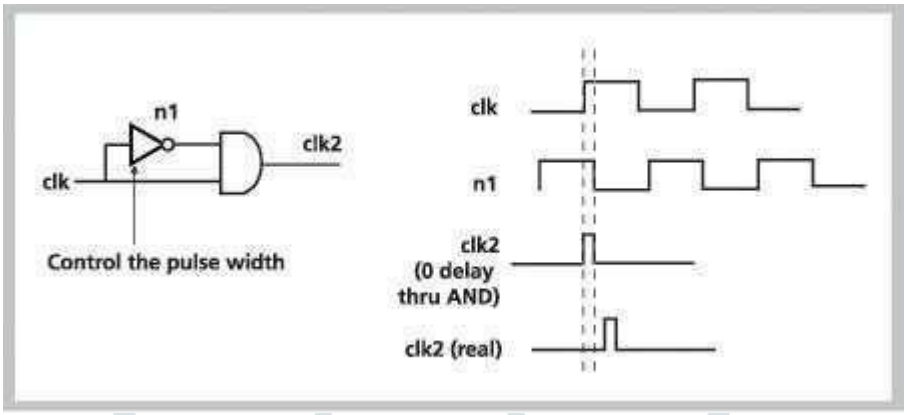


Fig2.8: Pulse Latch

III. SHIFT REGISTERS AND LATCH PULSE ARCHITECTURE :

Flip flops are use in constructing registers. Register is a group of flip flops used to store multiple bits of data. For example, if a computer is to store 16 bit data, then it needs a set of 16 flip flops. The input and outputs of a register are may be serial or parallel based on the requirement. A shift register is a sequential circuit which stores the data and shifts it towards the output on every clock cycle. Basically shift registers are of 4 types.

They are:

- Serial In Serial Out shift register
- Serial In parallel Out shift register
- Parallel In Serial Out shift register
- Parallel In parallel Out shift register

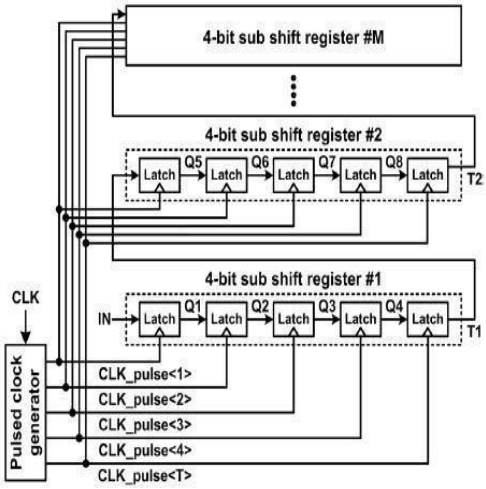


Fig3.1: Pulsed latch architecture

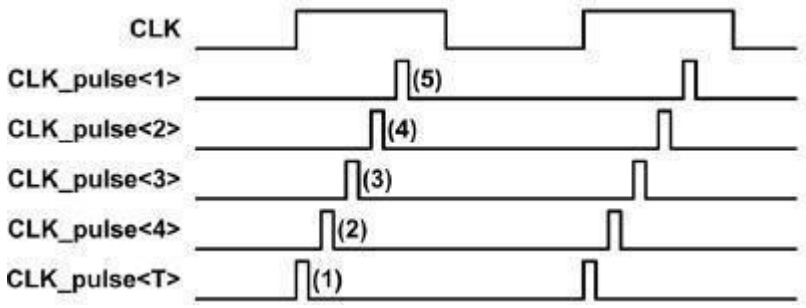


Fig3.2: Clock Signals

The power optimization is similar to the area optimization. The power is consumed mainly in latches and clock-pulse circuits. Each latch consumes power for data transition and clock loading. When the circuit powers are normalized with a latch, the power consumption of a latch and a clock-pulse circuit are 1 and, respectively. The total power consumption is also. An integer for the minimum power is selected as a divisor of, which is nearest to. In selection, the clock buffers are not considered. The total size of the clock buffers is determined by the total clock loading of latches.

In a long shift register, a short clock pulse cannot through a long wire due to parasitic capacitance and resistance. At the end of the wire, the clock pulse shape is degraded because the rising and falling times of the clock pulse increase due to the wire delay.

IV. DESIGN TOOLS

VLSI stands for "Very Large Scale Integration". This is the field which involves packing more and more logic devices into smaller and smaller areas. VLSI

- Simply we say Integrated circuit is many transistors on one chip.
- Integrated circuit (IC) may contain millions of transistors, each a few mm in size
- Applications wide ranging: most electronic logic devices

In VLSI technology design we use different languages like VHDL, Verilog, System verilog etc. Verilog language is used for the design.

4.1 Applications of VLSI:

Electronic systems now perform a wide variety of tasks in daily life. Electronic systems in some cases have replaced mechanisms that operated mechanically, hydraulically, or by other means; electronics are usually smaller, more flexible, and easier to service. In other cases electronic systems have created totally new applications. Electronic systems perform a variety of tasks, some of them visible, some more hidden.

- Personal entertainment systems such as portable MP3 players and DVD players perform sophisticated algorithms with remarkably little energy.

- Digital electronics compress and decompress video, even at high-definition data rates, on-the-fly in consumer electronics.

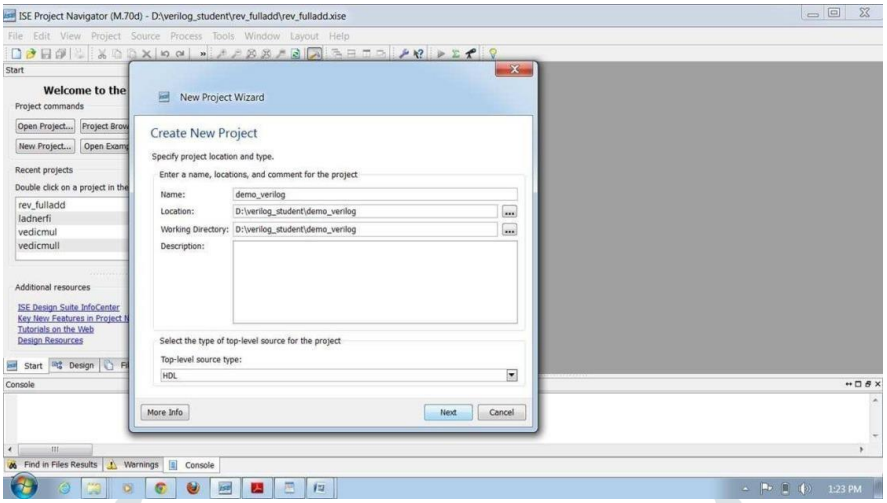
- Low-cost terminals for Web browsing still require sophisticated electronics, despite their dedicated function.

4.2 Xilinx:

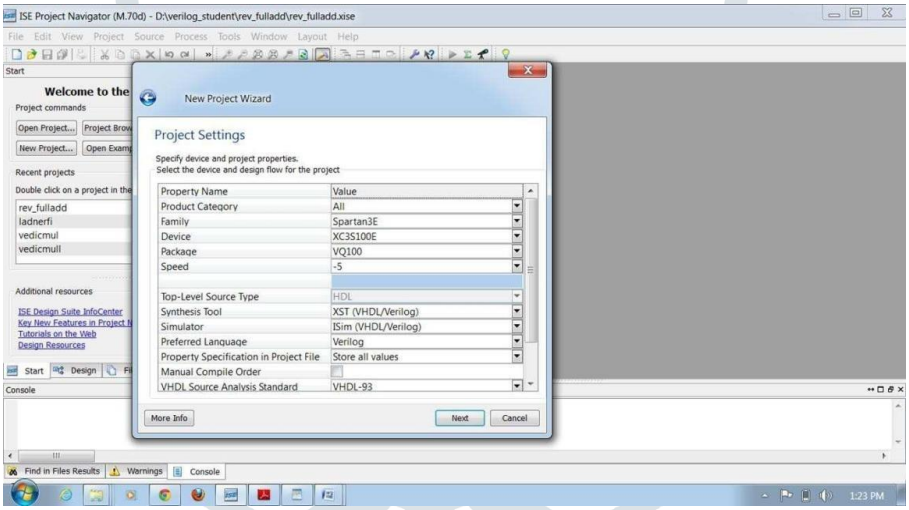
Xilinx software is used by the VHDL/VERILOG designers for performing Synthesis operation. Any simulated code can be synthesized and configured on FPGA. Synthesis is the transformation of VHDL code into gate level net list. It is an integral part of current design flows.

Algorithm:

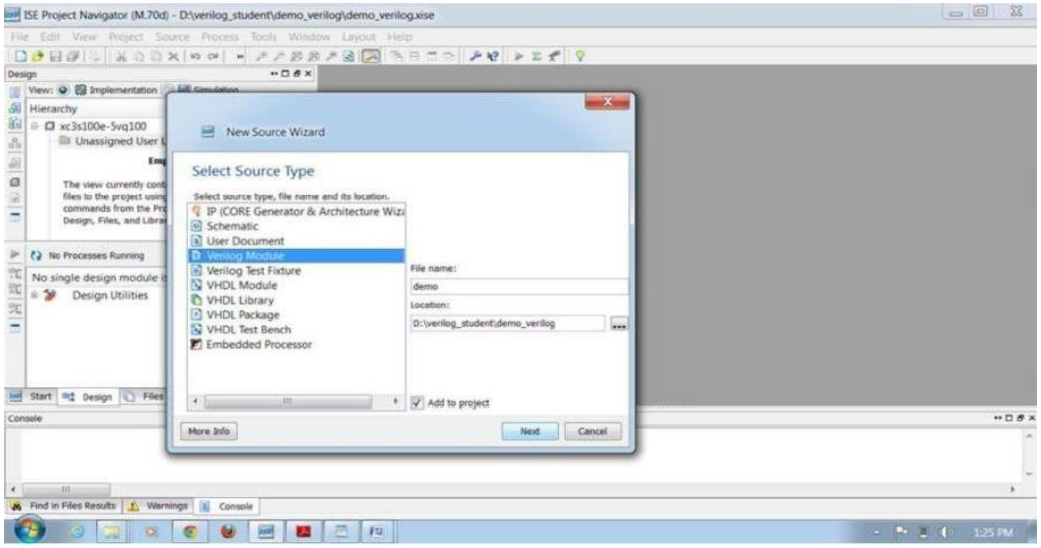
- Start the ISE Software by clicking the XILINX ISE icon.
- Create New Project

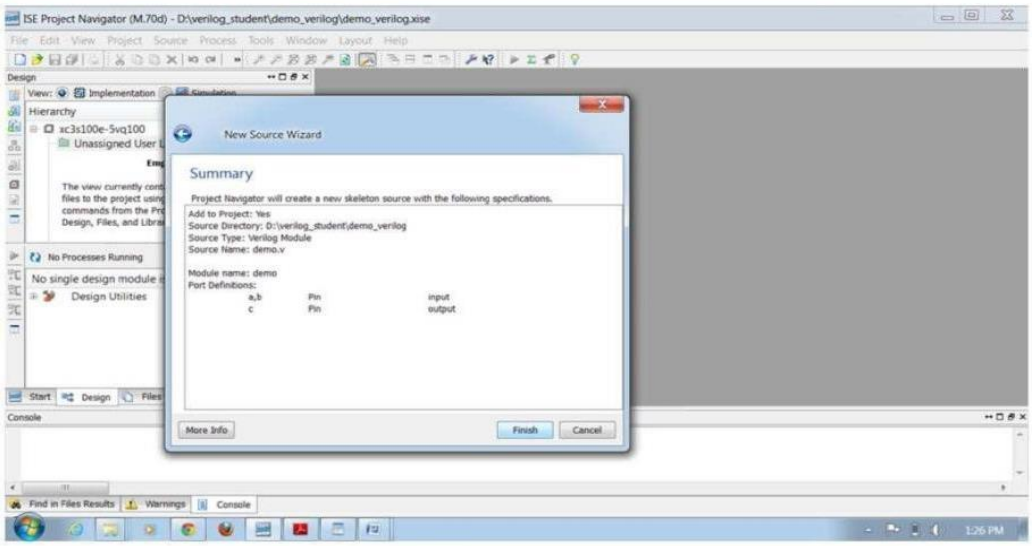


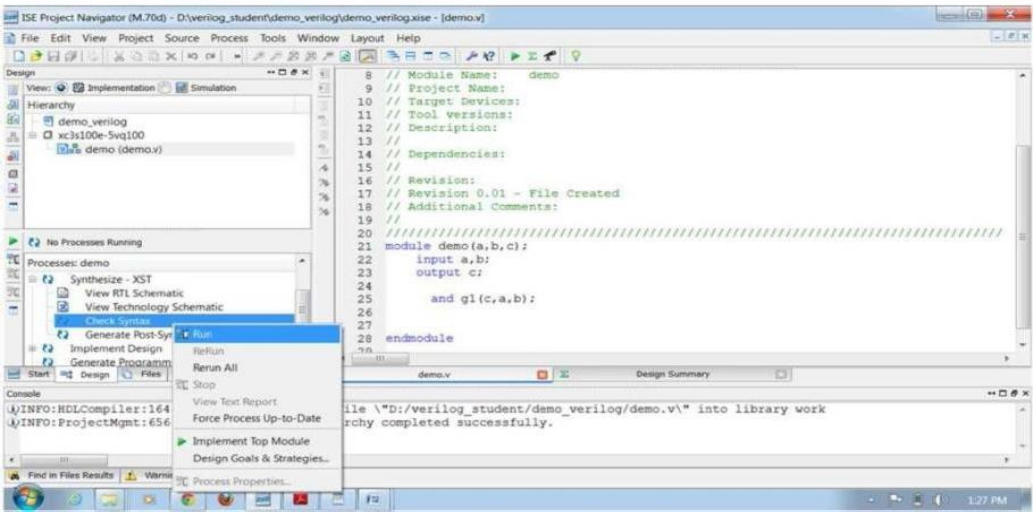
- And find the following properties displayed



- Create a DHDL source formatting all inputs, outputs and buffers


ISE Project Navigator (M.70d) - D:\verilog_student\demo_verilog\demo_verilog.xise
New Source Wizard
Select Source Type
Select source type, file name and its location.
IP (CORE Generator & Architecture Wiz
Schematic
User Document
Verilog Module
Verilog Test Fixture
VHDL Module
VHDL Library
VHDL Package
VHDL Test Bench
Embedded Processor
File name:
demo
Location:
D:\verilog_student\demo_verilog
Add to project
More Info
Next
Cancel


ISE Project Navigator (M.70d) - D:\verilog_student\demo_verilog\demo_verilog.xise
New Source Wizard
Summary
Project Navigator will create a new skeleton source with the following specifications.
Add to Project: Yes
Source Directory: D:\verilog_student\demo_verilog
Source Type: Verilog Module
Source Name: demo.v
Module name: demo
Port Definitions:
a,b Pin input
c Pin output
More Info
Finish
Cancel

- 
ISE Project Navigator (M.70d) - D:\verilog_student\demo_verilog\demo_verilog.xise - [demo.v]
demo_verilog
xc3s100e-5vq100
demo (demo.v)
Synthesize - XST
View RTL Schematic
View Technology Schematic
Check Syntax
Run
ReRun
Rerun All
Stop
View Text Report
Force Process Up-to-Date
Implement Top Module
Design Goals & Strategies...
// Module Name: demo
// Project Name:
// Target Devices:
// Tool versions:
// Description:
// Dependencies:
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
module demo(a,b,c);
input a,b;
output c;
and g1(c,a,b);
endmodule
INFO:HDLCompiler:164
INFO:ProjectMgmt:656
ile \"D:/verilog_student/demo_verilog/demo.v\" into library work
rchy completed successfully.

- VHDL code, to be synthesized
- Check Syntax after finally editing the VHDL source for any errors.
- After that, perform the RTL and TECHNOLOGY schematic for verifying synthesis

4.3 Verilog Language:

Verilog hdl is the one in every of the two commonest Hardware Description Languages (HDL) utilized by integrated circuit (IC) designers. Another coding language is VHDL. HDL is permits the design to be replicated advance within the design cycle so as to rectify mistakes or test with dissimilar design. Architectures represented in hdl are not depends on technology, simple to implement and correct, and are normally more understandable than schematics. Verilog are often describing styles at four levels of abstraction.

- Algorithmic level
- Register transfer level
- Gate level
- Switch level

The language additionally explains erects that may be worn to restraint the input and output of simulation. Some Verilog constructs don't seem to be producing electronically.Most readers can need to produce electronically their circuits, thus non synthesizable constructs ought to be used just for test benches. The words "not synthesizable" will be used for examples and constructs as needed that do not synthesize. The types of code in most HDLs are two types;

Structural, the diagrams in structural are verbal wiring which are with the absence of memory.

assign a=b | c & d; /* "|" is a OR */ assign p = (~q) & (r); The orders of the statements are not a matter. Change in **a**will occur by changing in **c**.

Procedural which is utilized to the circuits with memory, or a correct method to scribble conditional logic. always @(posedgeclk) // Execute the next statement on every rising clock edge.

Count <= count+1;

For composite, with flip-flop memory, this way of idea creates an excessive quantity of memory. But folks like procedural code as a result of it's typically abundant easier to put in writing, as an example, **if** and **case** statements square measure solely allowed in procedural code. As a result, the synthesizers are created which might acknowledge bound kinds of procedural code as truly combinatory. But if you digress from this vogue, beware.

4.3.1 Verilog modelling:

Verilog has four types of modelling. They are

- The switch level of modelling.
- The gate level of modelling.
- The Data flow modelling
- The Behavioral modelling

V. Results and Discussions

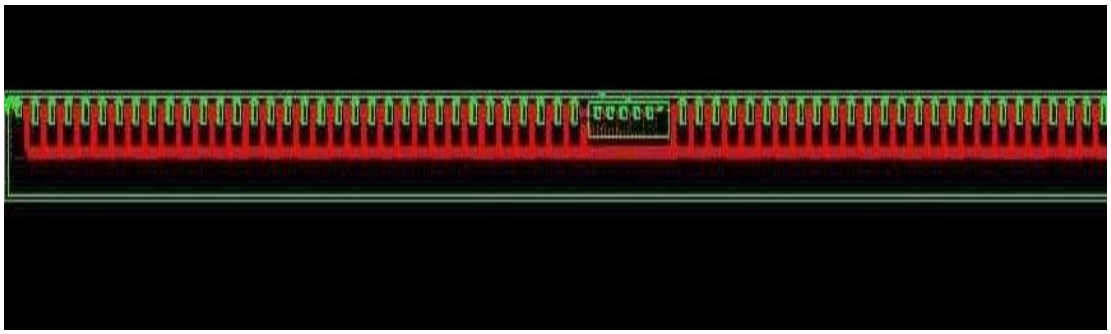


Fig: RTL Schematic

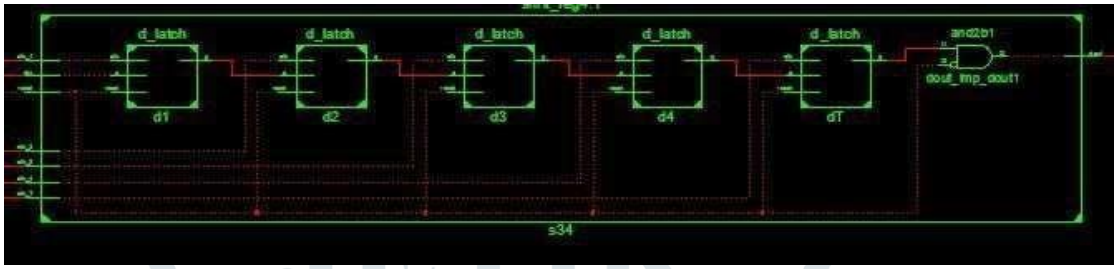


Fig: RTL internal structure

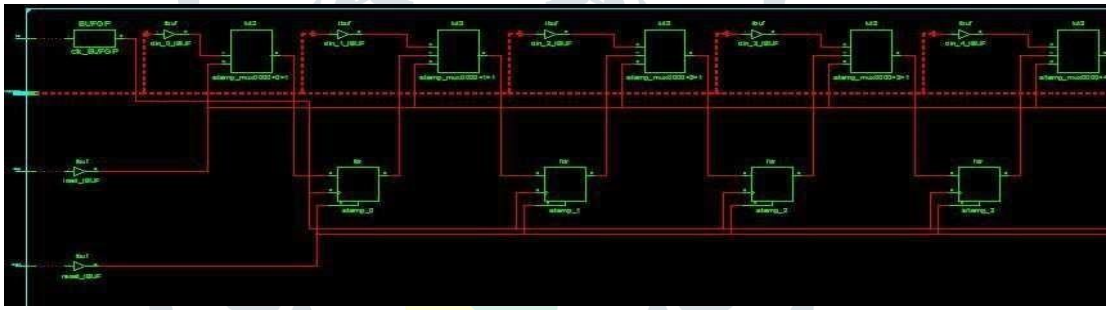


Fig: Technology Schematic

Fig: simulation results and related timing diagrams

VI. CONCLUSION:

This project proposes delay efficient shift register by using pulsed latches. Pulsed latches gives the same results similar to the flip flops but with less hardware resources. Flip flops need two latches. The same results can be obtained by using a clocked latch with pulse signal as clock input. This reduces the hardware requirement, in addition to this for producing different clock pulses for latches this project uses counter architecture for activating the latches. In this, counter is the circuit that increments its previous value by 1. In the increasing process several adders are implemented. This project proposes kogge stone adder which takes 0.932 nano seconds where as the conventional adder takes 3.497 nano seconds which reduces ¾ of the conventional adder delay.

REFERENCES

[1] P. Reyes, P. Reviriego, J. A. Maestro, and O. Ruano, "New protection techniques against SEUs for moving average filters in a radiation environment," IEEE Trans. Nucl. Sci., vol. 54, no. 4, pp. 957–964, Aug. 2007.

[2] M. Hatamian et al., "Design considerations for gigabit ethernet 1000 base-T twisted pair transceivers," Proc. IEEE Custom Integr. Circuits Conf., pp. 335–342, 1998.

[3] H. Yamasaki and T. Shibata, "A real-time image-feature-extraction and vector-generation vlsi employing arrayed-shift-register architecture," IEEE J. Solid-State Circuits, vol. 42, no. 9, pp. 2046–2053, Sep. 2007.

[4] H.-S. Kim, J.-H. Yang, S.-H. Park, S.-T. Ryu, and G.-H. Cho, "A 10-bit column-driver IC with parasitic-insensitive iterative charge-sharing based capacitor-string interpolation for mobile active-matrix LCDs," IEEE J. Solid-State Circuits, vol. 49, no. 3, pp. 766–782, Mar. 2014.

[5] S.-H. W. Chiang and S. Kleinfelder, "Scaling and design of a 16-megapixel CMOS image sensor for electron microscopy," in Proc. IEEE Nucl. Sci. Symp. Conf. Record (NSS/MIC), 2009, pp. 1249–1256.

[6] S. Heo, R. Krashinsky, and K. Asanovic, "Activity-sensitive flip-flop and latch selection for reduced energy," IEEE Trans. Very Large Scale Integer. (VLSI) Syst., vol. 15, no. 9, pp. 1060– 1064, Sep. 2007.

[7] S. Naffziger and G. Hammond, "The implementation of the nextgeneration 64 b itanium microprocessor," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2002, pp. 276–504.

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

[9] E. Consoli, M. Alioto, G. Palumbo, and J. Rabaey, "Conditional push-pull pulsed latch with 726 fJops energy delay product in 65 nm CMOS," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2012, pp. 482–483.

[10] V. Stojanovic and V. Oklobdzija, "Comparative analysis of masterslave latches and flip-flops for high-performance and low-power systems," IEEE J. Solid-State Circuits, vol. 34, no. 4, pp. 536–548, Apr. 1999.

[11] J. Montanaro et al., "A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor," IEEE J. Solid-State Circuits, vol. 31, no. 11, pp. 1703–1714, Nov. 1996.

[12] S. Nomura et al., "A 9.7 mW AAC-decoding, 620 mW H.264 720p 60fps decoding, 8-core media processor with embedded forwardbody- biasing and power-gating circuit in 65 nm CMOS technology," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2008, pp. 262–264.

[13] Y. Ueda et al., "6.33 mW MPEG audio decoding on a multimedia processor," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2006, pp. 1636–1637.

[14] B.-S. Kong, S.-S. Kim, and Y.-H. Jun, "Conditional-capture flip-flop for statistical power reduction," IEEE J. Solid-State Circuits, vol. 36, pp. 1263–1271, Aug. 2001.