

High Speed Pipeline Architecture Using Mixture of Domino Logic Gates

A. Subasri¹, B. G. Gopal², Dr. P. G. Kuppusamy³ J. T. Arun Ragesh⁴

¹PG Scholar, ^{2,3,4} Professors

^{1,2,3} Department of Electronics and Communication Engineering,
^{1,2,3} Veltech Multitech Engineering College, Avadi, Chennai, India.

Abstract

The present study aims to design method of asynchronous logic pipeline which focuses on improving the quality of the circuit efficient and it is used for a wide range of applications. This method is to construct a single rail domino gate and dual rail domino gate to perform synchronizing logic gate, to synchronize logic gate with a latch function and also to analyze the delay voltage and threshold voltage. This feature offers reduced critical delays, smaller silicon area and low power consumption. Single-rail domino logic alone cannot be used because it would break the domino data path as only non-inverting logic can be implemented. Dual-rail domino gates are limited to build a fixed critical data path. The method discussed in this paper saves power by reducing the overhead of logic circuits. This approach includes a highly-concurrent handshake protocol, with fewer synchronization points between next pipeline stages than almost all existing asynchronous dynamic pipelining approaches. The evaluation results shows that the proposed design has better performance than a bundled data asynchronous domino logic pipeline.

Keywords: Asynchronous domino logic, Single rail, Dual rail.

Introduction

Asynchronous design is considered as a promising solution for dealing with the issues that relate to the global clock since it uses local handshake instead of externally supplied global clock. Handshake circuits generate local clock pulses and use delay

matching to indicate valid signal. It leads to the most efficient circuits due to the extensive use of timing assumptions. The four-phase dual-rail protocol design is implemented in a detailed way that the handshake signal is combined with the dual-rail encoding of data.

This project presents a design method of asynchronous domino logic pipeline which focuses on improving the circuit efficiency and make in asynchronous domino logic pipeline design more practical for a wide range of applications. This design method combines the advantages of the four-phase dual-rail protocol and the four-phase bundled-data protocol to improve the efficiency of area and power in asynchronous domino logic pipeline.

The latch less feature provides the benefits of reduced critical delays, less silicon area and lower power consumption. Asynchronous domino logic pipeline has a problem that dual-rail domino logic has to be used to compose the domino data path. Only single-rail domino logic cannot be used as it would break the domino data path as only non-inverting logic can be implemented. As a result, the domino data path has a dual-rail encoding above the level of head that consumes a lot of silicon area and power consumption.

Several related designs is also simply introduced. Synchronizing logic gates (SLGs) and synchronizing logic gates with latch function (SLGLs) are introduced to construct a fixed critical data path. The constructed critical data path is analyzed. Then, the complex pipeline structures are further discussed. In the extreme case, with a slow or stalled output environment data items has become bunched or stalled at close intervals. In all cases, input data items are processed instantly as they arrive, even with an unknown or irregular arrival rate without waiting for a clock edge. Hence, the inter-token spacing and the throughput rate are determined dynamically. Third asynchronous pipelines automatically provide flow control. Handshaking protocols inherently offer underflow and overflow protection, even speed environments. In contrast, synchronous pipelines are default include no flow control. Synchronous flow control is typically supported using explicit credit-based techniques involving extra registers or complex decoupled latch control. A signal, used for back pressure, must also be synchronized to the clock at every stage. It cause low circuit efficiency and put limit to the application area of four-phase dual-rail protocol design.

Domino Gates

Single Rail Domino Gate:

The main goal for use of dynamic logic styles is to use only faster n-transistors in the logic trees, to achieve higher performance and lesser area. The problem of directly connecting two dynamic stages is that the pre-charged high output node will be the result in conducting n-transistors in the following logic tree. At the beginning of evaluation phase, this connection to be grounded and it leadsto a discharge of the following output and in an erroneous state. DOMINO logic solve this issue with an inverter, following n-transistors are off and no flow out can occur. Therefore, single-rail logic styles like DOMINO realize only non-inverting functions. When the inverted outputs arrive first, they has enough time to switch off all n-transistors. The

non-inverted (DOMINO) outputs arrive later and all other transistors are already in correct evaluation state and no erroneous discharging can appear.

Single-Rail Data paths:

Two distinct solutions are now proposed for gate-level pipelining of wide single-rail data paths. The first approach is again to use partitioning. In this case, the dual-rail completion detectors are replaced with single-rail completion generators and bundling signals. However, there is one important difference: in single-rail bundled data paths (unlike dual-rail), each data stream has its own distinct “request” signal. Therefore, the merging of multiple data streams now requires the explicit combining of the multiple request signals at the stage’s inputs. This merger is achieved by a simple modification of the stage completion generator: additional request signals are typically easily accommodated by adding extra transistors in series. Since most wide data paths have processing logic that typically converges into only a few bits in the rightmost pipeline stage, only a single request will typically be generated by the pipeline for the right environment.

However, in the event that there are multiple request signals at the right end of the pipeline, either the environment must have the capability to handle them, or they need to be combined into a single unified request signal. An efficient way of combining multiple request signals is to add a small number of additional pipeline stages, which recombine the requests in a tree-like fashion. A second approach to handling wide single-rail data paths is much simpler. Nopartitioning is used; instead, throughput overheads is reduced by earlier generation of the control signals along the critical path. The idea is that the control signal for the completion generator can be produced earlier than the control signal that pre-charges and evaluates the stage. As a result, much of the overhead of broadcasting the control signal to the dynamic gates can be hidden, occurring in parallel with other pipeline operations.

However, the delays caused by the coupling capacitance effect and the buffer can guarantee that the slowest data path is steady. Conventional logic gate has data-dependency problem, the slowest gate function block would vary from one to others depending on input data patterns. Synchronizing logic gates (SLGs) is used to solve this problem. Another problem is that the slowest gate in a pipeline stage might early triggered by outputs from the previous pipeline stage. Then, synchronizing logic gates with a latch function (SLGLs) is extended to solve it. As a result, a stable critical path is available with the help of SLGs and SLGLs.

Dual Rail Domino Gate:

However, asynchronous domino logic pipeline has a common problem that dual-rail domino logic has to be used to create the domino data path. Single-rail domino logic cannot be used because it would break the domino data path hence only non-inverting logic can be implemented. As a result, the domino data path has a dual-rail encoding overhead that it consumes more silicon area and power consumption. Such overhead almost cancels out the area and power benefits provided by the latch less feature. Another problem is overhead of handshake control logic. Conventional designs of asynchronous domino logic pipeline is based on four-phase dual-rail protocol rely on

domino data path to transfer the data and encoded handshake signal and it is used to detect the completion detectors and collect the handshake signal throughout the entire data paths.

Dual-Rail Data paths:

A wide dual-rail data path may be partitioned into several data streams. As in wide FIFOs, each completion detector only examines the bits belonging to its own data stream. As a result, completion detection is low cost, and it has to be partitioned among narrow data streams. Similarly, the output of a stage's completion detectors are fanned-out to only those sources whose outputs belong to the fan-in of that stage. Since this fan-out is typically less, the distribution of the control signals is also low cost. In contrast to FIFOs where separate data streams do not interact, data paths with logic processing may involve merging and forking of data streams at certain points in the pipeline, which can complicate the pipeline structures.

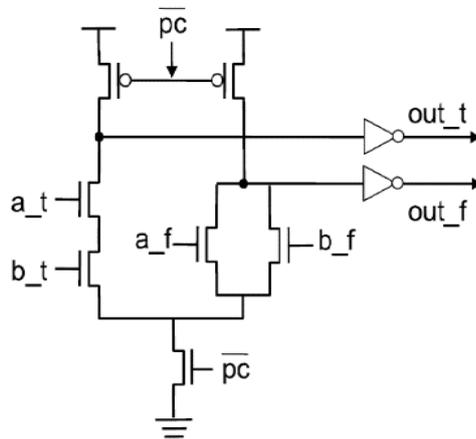


Figure 1: Dual-rail domino AND gate

As an example, a pipelined 32-bit ripple-carry adder was designed using the previous ideas. Each adder stage only depends on three inputs: two data bits and one carry-in. Therefore, in a gate-level pipeline implementation, each stage would simply have three inputs, two outputs, and a completion detector on only those outputs (and acknowledge only the corresponding input sources). This example illustrated how merged data streams can be effectively handled in a gate-level pipeline

Advanced Pipelines

1) Pre-Charge Half-buffer pipeline

A block diagram of pre-charge half-buffer pipeline (PCHB) shows in Figure 2. PCHB is a timing-robust pipeline style that uses quasi-delay-insensitive control circuits. Two completion detectors in a PCHB stage: one on the input side (Di) and one on the output side (Do). The complete cycle of events for a PCHB stage is quite similar to

that of PS0, except that a PCHB stage verifies its input bits. Because of input completion detector (Di), PCHB stage does not start evaluation until all input bits are valid. This design method absorbs skew across individual bits in the data paths.

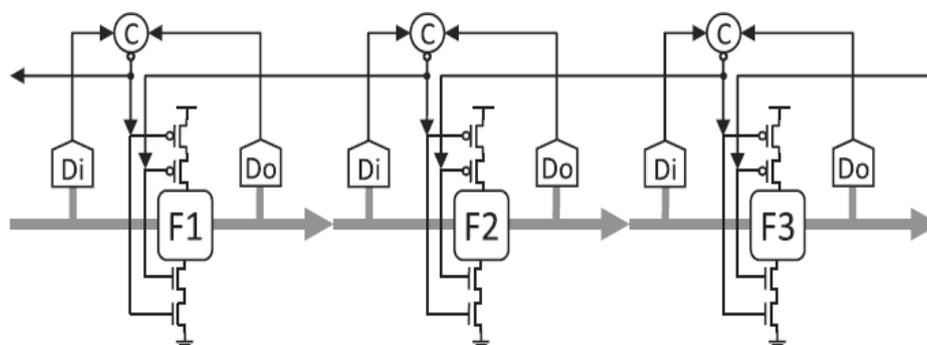


Figure 2: Block diagram of PCHB

2) LP2/2:

LP2/2 is a high-throughput pipeline style, which had both dual-rail protocol design and bundled-data protocol design based on the dual-rail protocol. LP2/2 improves the throughput of PS0 by optimizing the sequential of handshake events. However, they do not solve overhead problems in handshake control logic and function block logic. The handshake speed is accelerated by employing completion detectors placed ahead of function blocks. Although this pipeline structure reduces the handshake cycle time, the asymmetric completion detectors still consumes a lot of power since they have to detect the entire data paths.

3) Delay Assumptions

PCHB is a very robust pipeline that it requires no delay assumptions or calculations. However, the robustness of circuits comes at the expense of its performance. The complex handshake circuits slow down the handshake speed and consume more power. For fast handshake speed is achieved by designing PS0 using control circuits that are always correct for common conditions.

CriticalData path

Gate-delay data-dependence problem limits the construction of stable critical datapath using traditional logic gates. The critical Signal transition varies from one data path to others according to different input data. Since SLGs has solved the gate-delay data-dependence problem, a stable critical data path can be easily constructed by the following steps:

- 1) To find a gate (named as Lin gate) in each pipelined stage that has the largest number of inputs.
- 2) Changing these Lin gates to SLGs;

- 3) Linking SLGs together to form a stable critical data path. The idea for finding the critical signal transition is that embedding an SLG in each pipeline stage and making the SLG to the last gate to start and finish evaluation. First of all, the embedded SLG has the largest gate delay in a pipeline stage.

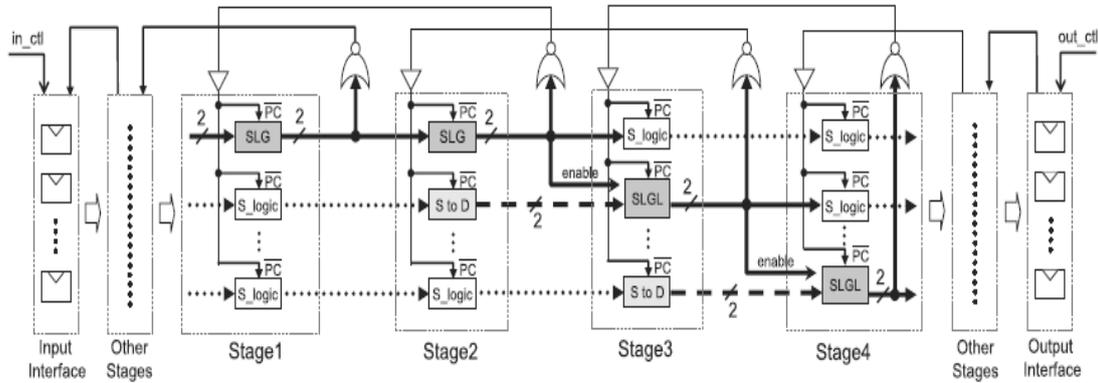


Figure 3: APCDP

After linking each pipeline stage's SLG together, the SLG in the following pipeline stage be the last gate to start evaluation since it always waits for the critical signal transition from previous SLG. As a result, the linked SLG data path becomes a stable critical data path. Linking each pipeline stage's SLG is partially done in the process of selecting Lin gate in each pipeline stage.

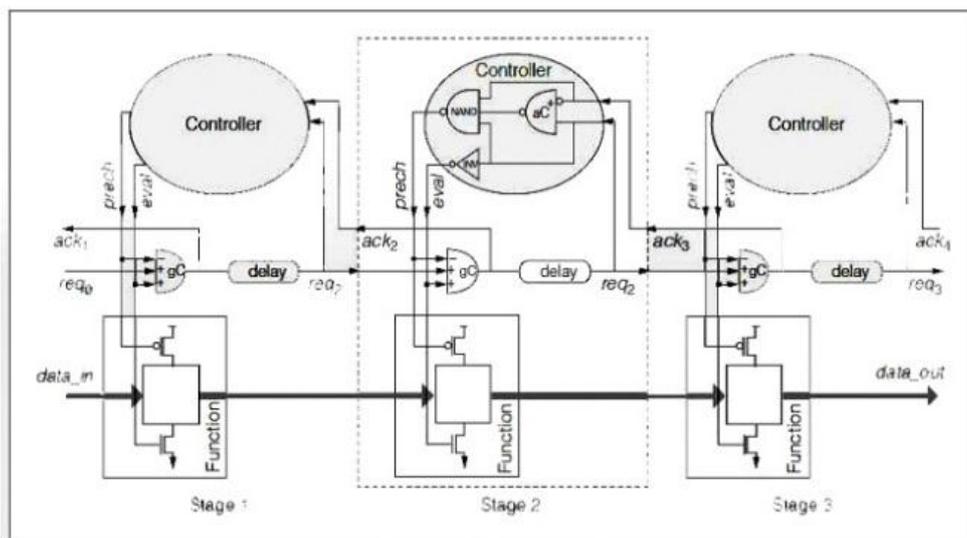


Figure 4: Architecture

Results and Analysis

The following figures indicates the delay analysis, threshold voltage analysis and different pipelined architecture.

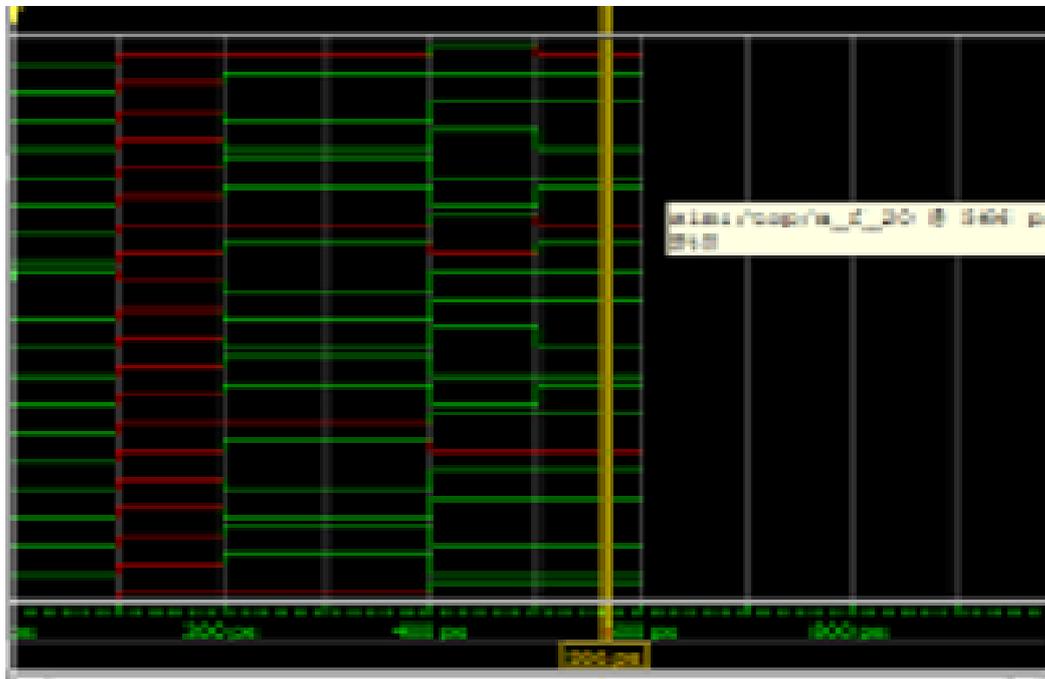


Figure 4: Asynchronous Pipeline Design Based on Constructed Data path

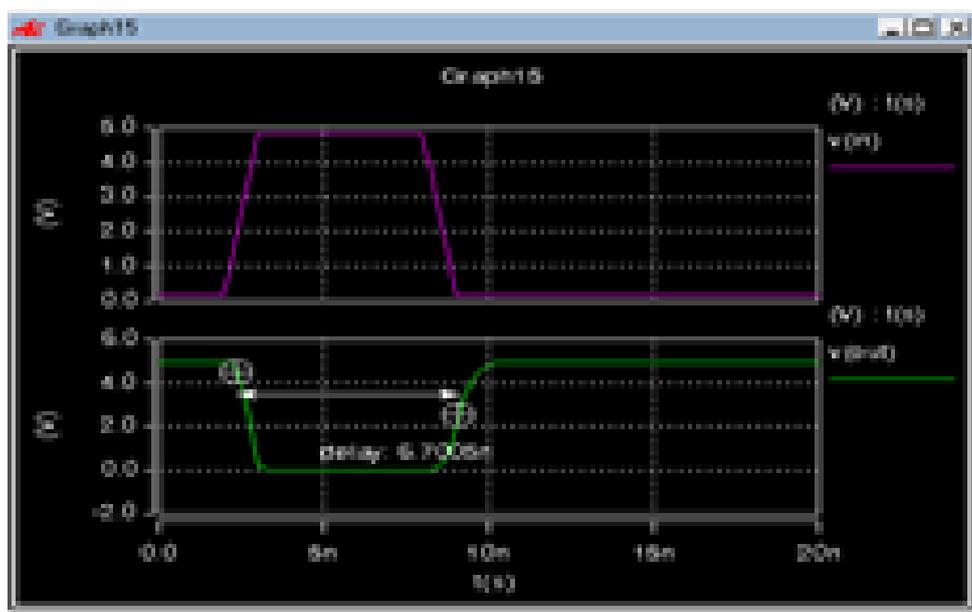


Figure 5: Delay Analysis

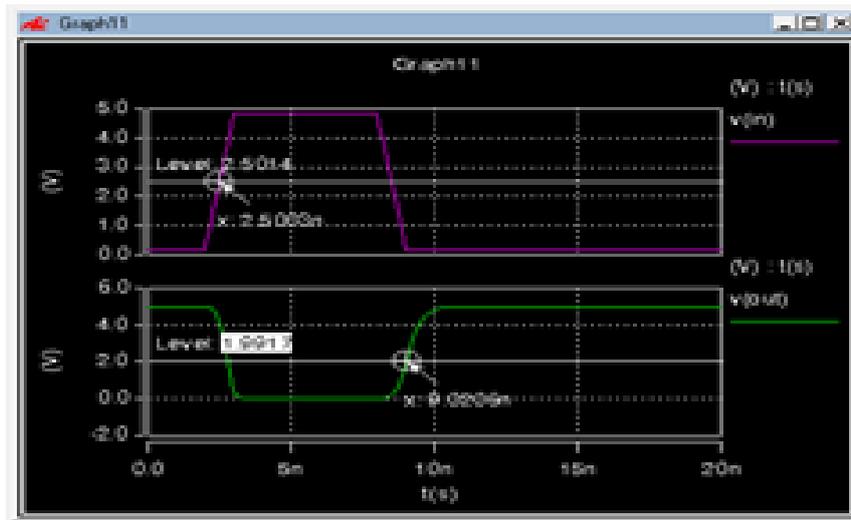


Figure 6: Threshold Voltage Analysis

Proposed Asynchronous Pipeline Design		Bundled data asynchronous pipeline	Synchronous pipeline
Logic Gate	Domino gate	Domino gate	Static Gates
Delay	6. 7005ns	8. 6500ns	9. 7500ns
Total memory	159 kb	248kb	278kb
Transistor Count	7375	9874	9010
Threshold voltage	2. 5v	3. 6v	4. 8v

Conclusion

This paper introduced a novel design method of asynchronous domino logic pipeline. The pipeline is realized based on a constructed critical data path. This design method greatly reduces the overhead of handshake control logic function block logic, not only increases the pipeline throughput but also decreases the power consumption.

In order to implement modules, control logic has been separated from the data path, while provision must be made for initialization and testing. If data bundling is used, the data path will look much the same as that in a synchronous implementation. The main differences are likely to be the routing of a data valid signal and the use of latches controlled by local signals, rather than registers controlled by a global clock signal. If double-rail coding is used, the logic blocks on the data path may resemble domino logic.

The evaluation results show that the proposed design has better performance than bundled-data asynchronous domino logic pipeline. It is even comparable with a synchronous pipeline with sequential clock gating.

References

- [1] Z. Xia, M. Hariyama, and M. Kameyama, "Asynchronous Domino Logic Pipeline Design Based on Constructed Critical Data Path," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 4, Apr. 2015.
- [2] B. H. Calhoun, Y. Cao, X. Li, K. Mai, L. T. Pileggi, and R. A. Rutenbar, "Digital circuit design challenges and opportunities in the era of nanoscale CMOS," *Proc. IEEE*, vol. 96, no. 2, pp. 343–365, Feb. 2008.
- [3] M. Krstic, E. Grass, F. K. Gurkaynak, and P. Vivet, "Globally asynchronous, locally synchronous circuits: Overview and outlook," *IEEE Des. Test Comput.*, vol. 24, no. 5, pp. 430–441, Sep./Oct. 2007.
- [4] M. Hariyama, S. Ishihara, and M. Kameyama, "Evaluation of a field-programmable VLSI based on an asynchronous bit-serial architecture," *IEICE Trans. Electron.*, vol. E91-C, no. 9, pp. 1419–1426, Sep. 2008.
- [5] S. M. Nowick and M. Singh, "High-performance asynchronous pipelines an overview," *IEEE Des. Test Comput.*, vol. 28, no. 5, pp. 8–22, Sep./Oct. 2011.
- [6] M. Singh and S. M. Nowick, "The design of high-performance dynamic asynchronous pipelines: Lookahead style," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 11, pp. 1256–1269, Nov. 2007.
- [7] M. Singh, J. A. Tierno, A. Rylyakov, S. Rylov, and S. M. Nowick, "An adaptively pipelined mixed synchronous-asynchronous digital FIR filter chip operating at 1.3 gigahertz," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 7, pp. 1043–1056, Jul. 2010.
- [8] S. Ahuja and S. Shukla, "MCBCG: Model checking based sequential clock-gating," in *Proc. High Level Des. Validation Test Workshop*, 2009, pp. 20–25.
- [9] L. Li, W. Wang, K. Choi, S. Park, and M. -K. Chung, "SeSCG: Selective sequential clock gating for ultra-low-power multimedia mobile processor," in *Proc. IEEE Int. Conf. EIT*, May 2010, pp. 1–6.
- [10] Z. Xia, S. Ishihara, M. Hariyama, and M. Kameyama, "Dual-rail/single-rail hybrid logic design for high-performance asynchronous circuit," in *Proc. IEEE ISCAS*, May 2012, pp. 3017–3020.
- [11] BG. Gopal, A. Lavanya, "Impley Logic Implementation of Carry Save Adder Using Memristors" *International Journal of Engineering Research and Applications*, 2015 vol. 5, issue. 5 pp: 105-109
- [12] Gopal BG Poomalai M, Design of Low Power Sequential System Using Multi Bit FLIP-FLOP with Data Driven Clock Gating, *International Journal of Engineering Research and Applications*, 2015 vol. 5, issue. 5
- [13] Gopal B. G, Jency Rubia J, Prabhu V, Analysis, Design and Implementation of 4-Bit Full Adder using FinFET, *Journal of Convergence Information Technology (JCIT)*, 2015, Vol. 10, Issue. 2
- [14] Dr. Kuppusamy P. G, "A New Power Gating Scheme Using Multi-Mode Power Switches For Static Power Reduction" *International Journal of Engineering Research and Applications*, 2015 vol. 5, issue. 5
- [15] Dr. Kuppusamy P. G, "Design of Level Shifter Using Dual Cascode Voltage Switch for Low Power Application" *International Journal of Engineering Research and Applications*, 2015 vol. 5, issue. 5

