INTERNATIONAL JOURNAL FOR INNOVATIVE RESEARCH IN MULTIDISCIPLINARY FIELD ISSN: 2455-0620 Volume - 5, Issue - 11, Nov– 2019
Monthly, Peer-Reviewed, Refereed, Indexed Journal with IC Value: 86.87 Impact Factor: 6.497
Received Date: 15/11/2019 Acceptance Date: 28/11/2019 Publication Date: 30/11/2019

Implementation of Parallel Single Rail Self Time Adder using Recursive Approach

[1]Ms.T.Vijayalakshmi, [2]Mr.P.Prabakaran, [3]Mr.G.Manikannan

[1]Assistant Professor, [2] Assistant Professor, [3] Assistant Professor
[1, 2, 3] Department of Electronics and Communication Engineering,
[1, 2, 3] CK College of Engineering and Technology, Cuddalore,Tamilnadu.
Email – [1] vijayalakshmi.aec@gmail.com.

Abstract: *This paper presents the VLSI Implementation of self time adder(parallel single-rail self-time adder). It is based on the recursive formula for performing multibit binary additions. This functionality is parallel to those bits that do not require any carry chain propagation. Therefore, the design achieves logarithmic performance under random operating conditions without any special speed circuitry or look-ahead schema. The practical implementation is provided with a complete diagnostic unit. Implementation has no conventional and high fan practical limitations. Although more fan-gate is required, it is inevitable for asynchronous logic and is governed by parallelizing transistors. Simulations have been performed using industry standard toolkits that verify the practicality and superiority of the proposed approach on existing asynchronous recruiters.*

Key Words: *Asynchronous circuits, binary adders, CMOS design, digital arithmetic.*

1. INTRODUCTION:

Most of today's digital systems are clock-based or synchronous, considering that signals are binary and time independent. Typically, synchronization systems include multiple subsystems that switch from one state to another depending on the global clock signal, and flip-flops (registers) are used to store different levels of the subsystem. A typical synchronization system is depicted by Figure 1.1. State updates within the registry are carried out either on the rising edge (positive edge) or fall edge (negative edge) of the global clock triggering the single edge. The status of the global clock allows for data loading or data storage. Dual edge induced flip-flops were proposed later in the literature with the aim of increasing system performance since the overall clock usage is only 50% for single edge induced systems and can load data on both rising and falling clock edges. The clock signal does not change [1] [2]. However, this usually comes at a huge cost.

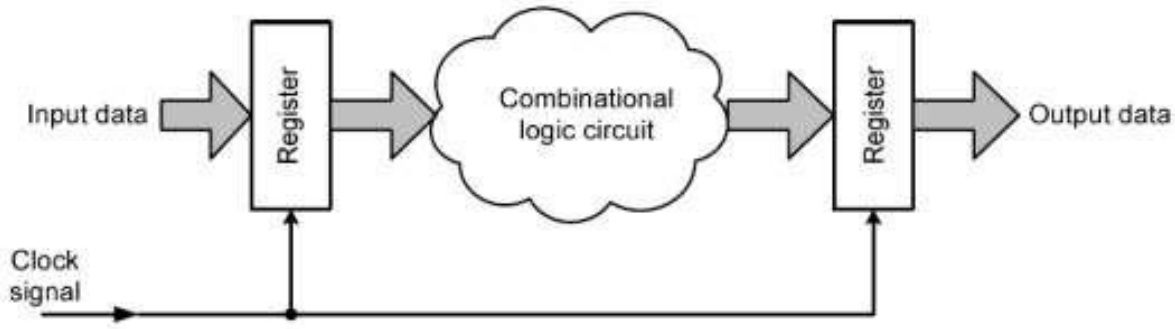

Fig 1.1: A typical synchronous system stage

The silicon footprint triggered flip-flop due to the large number of transistors and high connections to the double edge, resulting in higher power consumption. Preserving the original data rate on the single edge may help reduce dynamic power dissipation as changes can be halved when operating at half the system clock frequency, but in the end it can be offset by more leaks. Energy dissipation [2], which dominates deep submicron technologies. Furthermore, this mechanism tends to avoid the benefits associated with single-edge stimulation, in which its setup and capture times are large compared to conventional flip-flops, and any deviation from its 50% duty cycle can lead to time failures on complex paths [3]. In addition, it is more sensitive to noise than introducing a problem with system design, and the specification of tremor tolerance is very severe, which complicates the design of the system phase lock ring. As a result, cohesive designs that dominate the rising or falling edge are the main stream of digital system architectures; Nevertheless, it is becoming increasingly difficult to overcome some of the fundamental limitations inherent in this approach.

The International Technology Roadmap for Semiconductors (IDRS) predicts that computer-level synchronization is becoming impossible due to the complexity of silicon. A clock-based system can only work properly if all parts of the system are looking at the clock simultaneously, which can only occur if the clock wire is too low. However, with the advancement of technology, the number of transistors is greatly increased by the systems, and as a result the delay of the clock wires is no longer neglected. The problem of clock curve is a major disruption for many computer designers. Because the clock signal stores all flip-flops synchronously with their input data, it becomes more overloaded, and the problem becomes more acute. A widely preferred solution is to distribute the global clock and thereby control the clock curve using a clock network (clock tree) with clock buffers. As a result, this leads to an increase in the clock's net capacity, and suffers from increased activity (usually two changes per net per cycle), even ignoring the risk functions in such nets. The main factors that govern the clock curve in a typical coherent digital system are:

- Distribution network architecture, buffer schemes and used clock buffer
- Fabrication process variation in chip area.
- Restoration, Capacity and Induction of Interconnected Materials Used for Clock Distribution Network.
- The number of processing elements in the system and the load supplied by each element in the clock distribution network.
- Rise and fall times of clock frequency

Various clock distribution strategies have been developed, and the most common and common approach is to use buffer trees for symmetric clock distribution. However, for distributing high-speed clock signals, symmetric trees such as H-tree are preferred compared to the asymmetric buffer clock distribution tree structure. The H-Tree network is the most widely used clock distribution network [4] - [6] to reduce the clock curve. For an N × N processing component in [7], the clock pulse rising time and the corresponding clock curve are shown as O (N3). Therefore, with the increase of N, the clockwise curvature is likely to increase rapidly. Therefore, a distributed buffer scheme is often sought for synchronized digital integrated circuits by introducing buffers into the clock distribution network. However, the disadvantages of this approach are the increase in design sensitivity to additional area overhead and process variations. It should also be noted that buffers are the primary source of the total clock curve within a well-balanced clock distribution network. Because global clock times are now generally less than half a nanosecond, variations in the delay of tens of thousands of pico seconds can severely reduce the efficiency and reliability of high-speed synchronization systems [8]. Moore's law [9] is a driving force through process generations, supported by continuous innovations in processes and devices [10], persistence after high integral circuit densities, and is highly significant with variations in process and device parameters [11] [12] dimensions The above problem may be exacerbated by the fact that they are scaled to s. The key to this is the increasing number of clock management

1.1 Motivation and Context

Clock curve and power dissipation are key drivers to rekindle global interest in asynchronous design - notable major projects include [15] - [29]. The design of clock-free or asynchronous systems has become attractive to digital system designers over the past two decades, although asynchronous logic has been explored from the very beginning of integrated circuit design [30] - [32]. But the cohesive design provided a more efficient vehicle for using technology in commercial applications. According to the 2006 Semiconductor Industry Association's (SIA's) IDRS report, the percentage of designs powered by the handshake clock (asynchronous signal) will rise from 11% in 2008 to 40% in 2020. The latest IDRS update on the design [33] re-use of design (as a percentage of all logic) will increase the current number from 38% to 55%. During this period, the parameter uncertainty (as a result of the percentage of login delay) is expected to increase from 10% to 25%. In fact, reliability has been named as one of the five shortcut design challenges, bringing home the point that design robustness is becoming an increasingly important priority in deep submicron technologies.

The above forecasts tend to predict and necessitate a significant change in the design paradigm from conventional synchronous logic to non-asynchronous logic because of the latter advantages due to its ability to tolerate supply voltage, process parameters and temperature variations [15]. In the absence of universal clock reference, asynchronous circuits have better noise and electro-magnetic compatibility characteristics than synchronous circuits [34]. In addition, they have greater modulation that allows for convenient design reuse [30]. Asynchronous operation does not automatically mean low power, but it is often suggested that low power possibilities are based on observing when and where asynchronous circuits are active [35].

2. EXISTING SYSTEM :

Binary additions are the single most important function that a processor performs. Although there is a strong interest in lock-free asynchronous processors circuits [1], most combinations are designed for synchronous circuits. Asynchronous circuits assume no time scale. Therefore, they have great potential for logic design as they are free from the many problems of clock (coherent) circuits. In principle, the absence of logic flow clocks in asynchronous circuits

is constrained by the request-approved handshaking protocol for setting up a pipeline. Transparent handshaking modules are expensive for small elements such as bit joiners. Therefore, it can be managed indirectly and efficiently using a double-rail carry campaigns in combinations. The valid dual-rail carries output also receives approval from the single-bit combination module. Therefore, asynchronous combinations of all signals of the entire dual-rail encryption based (Boolean logic, rather than symbolic terms, the logic uses null convention logic [2] use) or single-rail data encryption and dual-rail carry using a pipe embedded in the process of approval For the representation. While these constructs add robustness to circuit designs, they also introduce significant overheads to the average case performance benefits of asynchronous adders. Therefore, an efficient alternative approach to solving these problems is worth considering.

This summary provides asynchronous parallel self-time insertion (PASTA) using the method originally proposed in [3]. Pasta's design is conventional and uses semi-adders (HAs) with multiplexes that require minimal connections. Therefore, it is suitable for VLSI implementation. The design works in parallel with the independent carry chain modules. Implementation in this summary is unique because it uses the concepts of XOR logic gates as a single-train cycle asynchronous continuous adder [4]. Rotational circuits are more resource efficient than their acyclic counterparts [5], [6]. On the other hand, the wave pipelining (or maximum rate pipelining) is a technique that can use piped embedded inputs before outputs are confirmed [7]. The proposed circuit governs the automatic single-rail pipe version of the carrier inputs separated by the propagation of the gates and the idle delays. Therefore, this is effectively a single-train wave tube based approach and is completely different from conventional tube adders using dual-rail encoding to indirectly represent the tube version of the carrier signals.

2.1 BACKGROUND

There are countless designs of binary adders, and we focus here on asynchronous self-time combinations. Self-time refers to logic circuits that depend on the correct operation and / or engineer timing assumptions. Self-time additives have the ability to run on average faster than dynamic data because the worst case of early completion sensitivity synchronous circuits can avoid the need for a compressed delay mechanism. They can be classified as follows.

2.1.1 Pipelined Adders Using Single-Rail Data Encoding

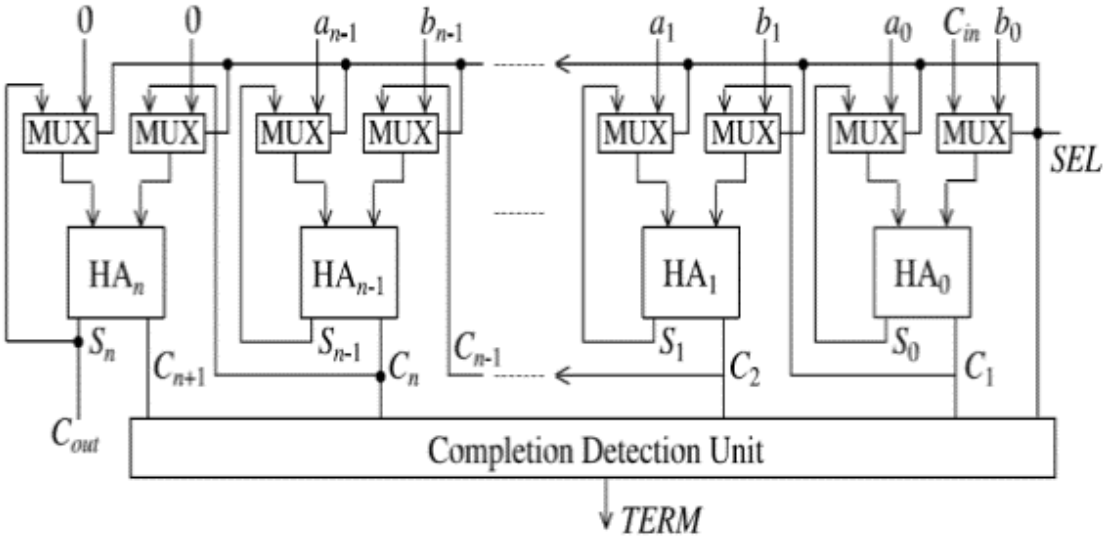

Fig. 2.1 General block diagram of PASTA.

The asynchronous Req/Ack handshake combination can be used to activate the volume and establish the flow of carry signals. In most cases, the double-rail carry assembly is used for the internal bitwise flow of the carry outputs. These double-train signals may represent two logic values (false, 0, 1), so that a bit function can be used to generate bit-level acknowledgment. The (higher) endpoint will be realized when all bit ac signals are received. Carry-filled Sensing Order An example of a tube insertion [8] is the use of full additive (FA) operating modules suitable for dual-rail carry. On the other hand, a speculative combination is proposed in [9]. This is called abortion logic and early completion to select the correct closing response from multiple standard delay lines. However, abandoned logic implementation is expensive due to high fan requirements.

2.1.2 Delay Insensitive Adders Using Dual-Rail Encoding

Delay insensitive (DI) adders are asynchronous combinations that emphasize compilation constraints or DI functions. Therefore, they can function properly in the presence of the bounded but unknown gate and wire delays [2]. There are several types of DI additives, namely DI ripple carry adder (DIRCA) and DI carry lookforward adder

(DICLA). DI adders use double-rail encryption and are thought to increase complexity. Although double-rail encryption doubles the wire problem, they can be used to produce circuits almost identical to single-rail variants using dynamic logic or NMOS only designs. As an example, 40 transistors are provided for a bit of TRCA insertion [8] using the usual CMOS RCA 28 transistors.

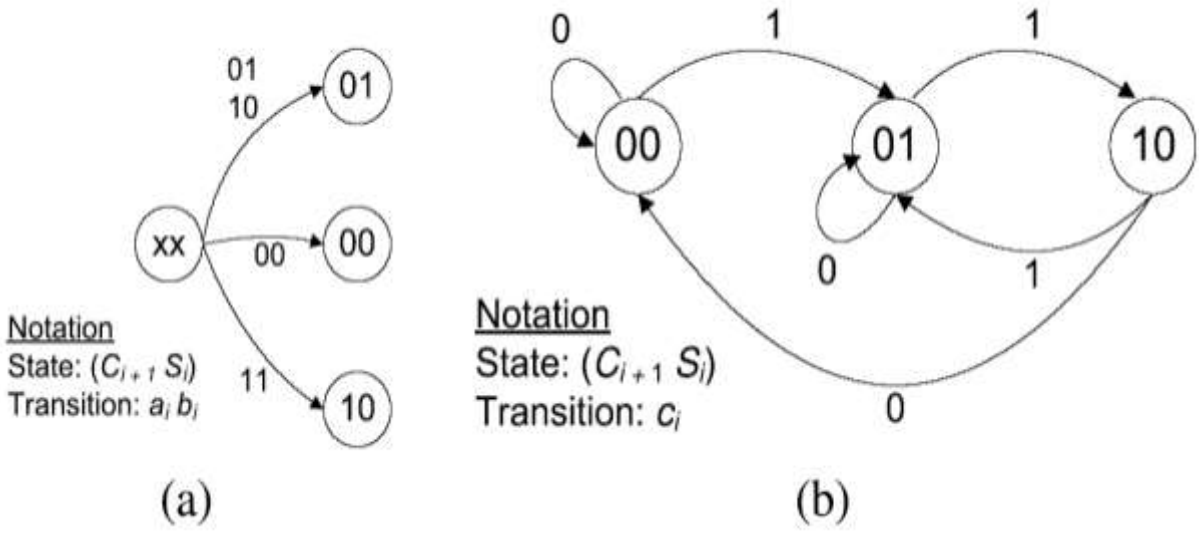

Fig. 2.2. State diagrams for PASTA. (a) Initial phase. (b) Iterative phase.

Like CLA, DICLA defines the carrying, generating and killing of equations based on double-rail encryption [8]. They do not enclose carry signals in a chain, but rather organize them into a hierarchical tree. Thus, they may be faster when there is a longer carry chain. Further optimization is provided by the observation that dual-rail encryption logic may benefit from resolving the 0 or 1 path. No need to wait for double-rail logic to evaluate both routes. Therefore, the Gary Look-Forward circuitry can be further accelerated to send Gary Generated / Gary-Gill signals to any level of the tree. This is described in detail in [8] and is referred to as DICLA with the speed circuit (DICLASP).

3. PROPOSED SYSTEM:

3.1 Design of pasta

The architecture and theory behind the pasta are presented. The adder first accepts two input functions to make half additions to each bit. Then, iterates again to make half-additions again until all the carry bits have been consumed using the previous carries and amounts and settled at zero level.

A. Architecture of Pasta

The general configuration of the joiner is shown in Figure 1.. The choice for the two input multiplexers is similar to the input reg handshake signal and is a single 0 to 1 conversion represented by the SEL. It initially selects the actual movements during SEL = 0, and switches to feedback / carrying paths for subsequent iterations using SEL = 1. The feedback path from HAs enables multiple iterations to be completed until all the carrier signals have zero values. .

B. State maps

In Figure 2, two state diagrams are drawn for the initial phase and the implementation phase of the proposed structure. Each state is represented by a pair of (Ci + 1 Si), where Ci + 1 and Si denote the activation and joint values from the ith bit adder set, respectively. During the initial phase, the circuit simply acts as a composite HA in the base mode. It is apparent that the use of HAs instead of FAs does not appear in stage (11).

During the repetition phase (SEL = 1), the feedback path is activated via the multiplexer module. Carry transitions (CI) are allowed multiple times as needed to complete the recurrence. From the definition of basic mode circuits, the current format cannot be considered a basic mode circuit because the input-outputs go through many changes before the final output is generated. This is not a Mueller circuit that operates locally outside the base system; Many changes will occur as shown on the state map. This is analogous to rotational sequence circuits where gate delays are used to separate individual states.

C. Recursive formula for binary additions

Theorem 1: The recursive creation of (1) - (4) will produce the correct amount for any bits and will be stopped within a specified time. Source: We demonstrate the correctness of the algorithm by completing the number of iterations required to complete the meeting (satisfying the stopping condition). Basic: Consider operational choices that do not require any carry propagation, that is, C0 i = 0 for ∀i, i ∈ [0..n]. The proposed creation will provide the correct result by a single bit computation time and (4) will cease immediately upon completion. Therefore, all single-

bit recruiters will be successfully terminated or propagated. The mathematical form presented above is valid under the condition that it progresses synchronously for all bit conditions, and that the input and output required for a particular iteration are consistent with the progress of a iteration. In the next section, we present the implementation of the proposed framework, which is then validated using simulations.

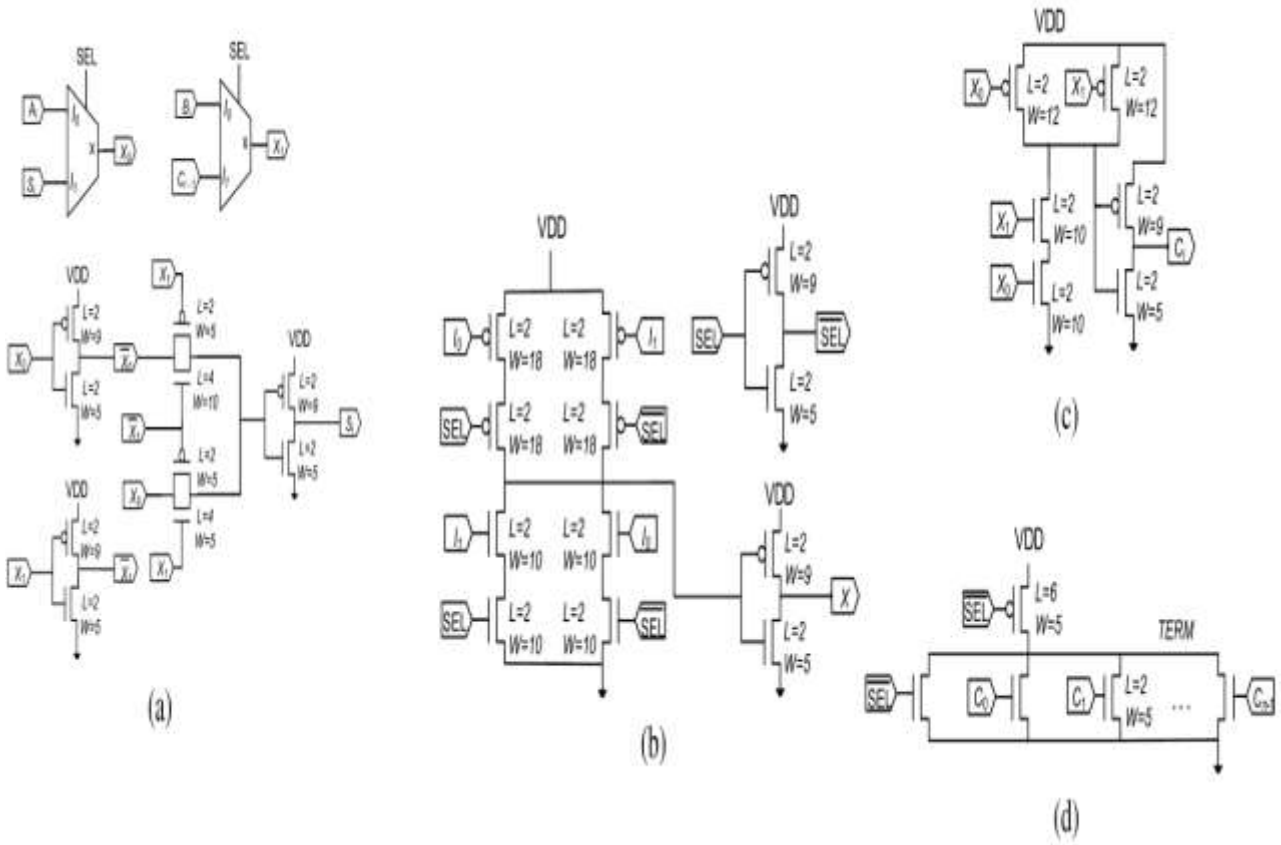

Fig. 3.1 CMOS implementation of PASTA. (a) Single-bit sum module. (b) 2×1 MUX for the 1 bit adder. (c) Single-bit carry module. (d) Completion signal detection circuit.

3.2 Implementation

A CMOS implementation for the recursive circuit is shown in Figure 3. We used TSMC library implementations for multiplexers and AND gates, while the transmission gate to the XOR gate used the fastest ten transistor processing based on the XOR. [4]. The following (4) detection is denied to receive the active high completion signal (TERM). This requires a large fan-in n-input NOR gate. Therefore, an alternative is to use the most practical pseudo-nMOS rate-ed design. The resulting design is shown in Figure 3 (d). Using the pseudo-nMOS format, the closing unit avoids much fan-problem because all connections are parallel. The PMOS transistor attached to the VDT of this rate-ed design acts as a load record, resulting in a constant current drain when some NMOS transistors are in the same position. With the exception of Ci s, the negative of the SEL signal has been added to the TERM signal to ensure that the completion is not accidentally activated during the initial selection phase of the actual inputs. This prevents the pMOS transistor from ever running. Therefore, the static calculation will only flow for the duration of the actual calculation. The VLSI layout is also done [Fig. 3 (c)] for a stable cell environment using two metal layers. The layout for 1-bit occupies 270 × × 130, resulting in a 1.123 Mλ2 area per 32-bit. Pull-down transistors of completion detection logic are included in the single-bit layout (T-terminal), while the pull-up transistor is placed on the full 32-bit insert. This is almost twice the area required for RCA, and is slightly less than the addition of partially efficient prefix wood, i.e., the Brent-Kung Adder (BKA).

4. SIMULATION RESULT:

In this section, we present simulation results for different joiners using the Tanner ETA Tools version 14.3 running on a 32-bit Windows platform. We have used standard library implementations of base gateways to enable other recruiters. Custom combinations like DIRCA / DICLASP are implemented based on their highly efficient designs.

To begin with, we show how pasta's current design can effectively perform binary additions to different temperature and process corners to ensure robustness under production and operating variables. In Figure 4, the time graphs for the worst and mean cases associated with maximum and mean length highlight the chain propagation over

random input values. Gary propagates through adjacent bit adders like a pulse, as clearly seen from Figure 4 (a). The best case (not shown here) relating to the minimum length carrier chain does not involve any carrier propagation and therefore only causes a single bit insertion delay before generating the TERM signal. The worst case involves the maximum carry propagation layer delay due to the length of the entire 32-bit carrier chain.

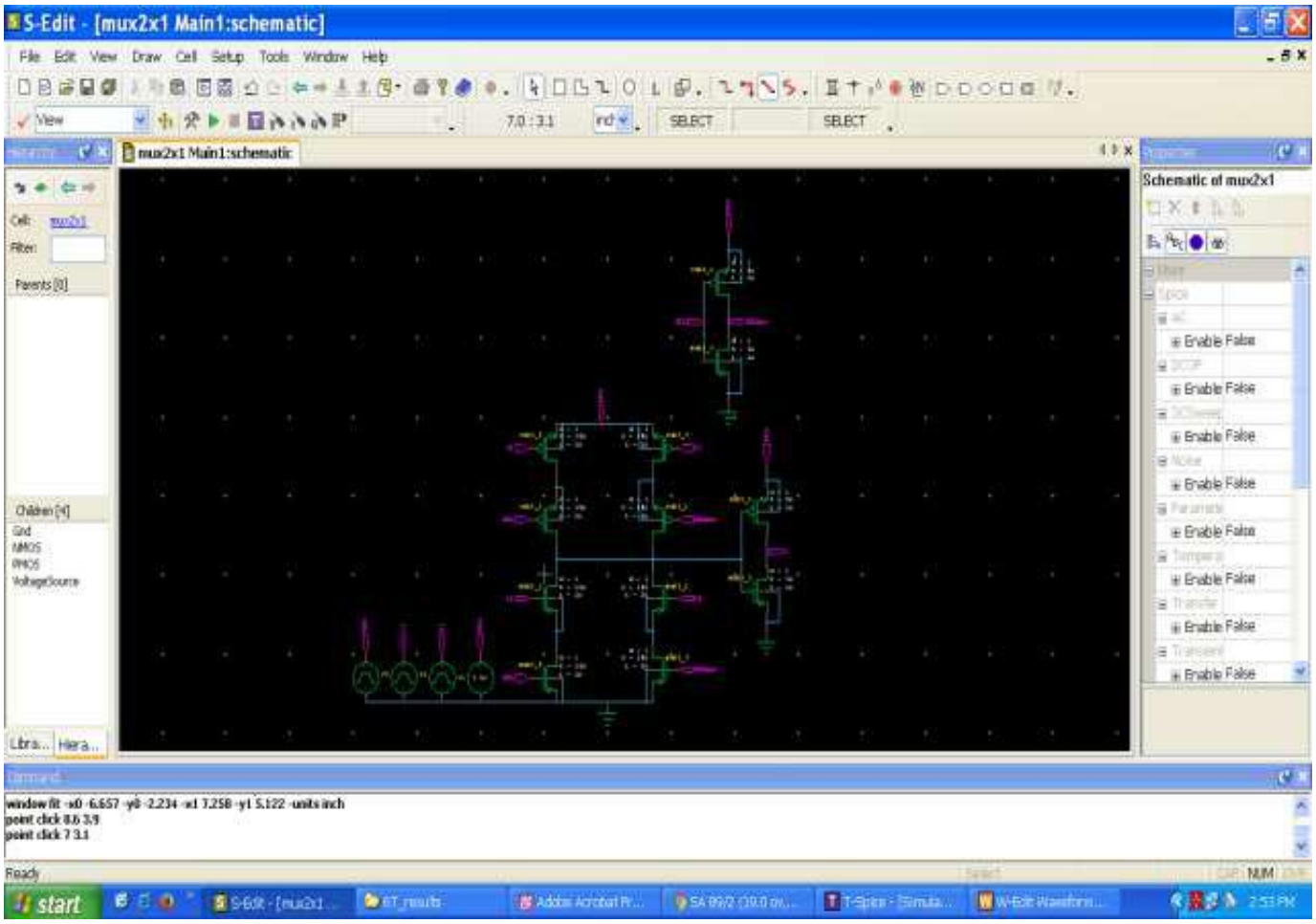

Fig 4.1: Mux 2 x 1 schematic

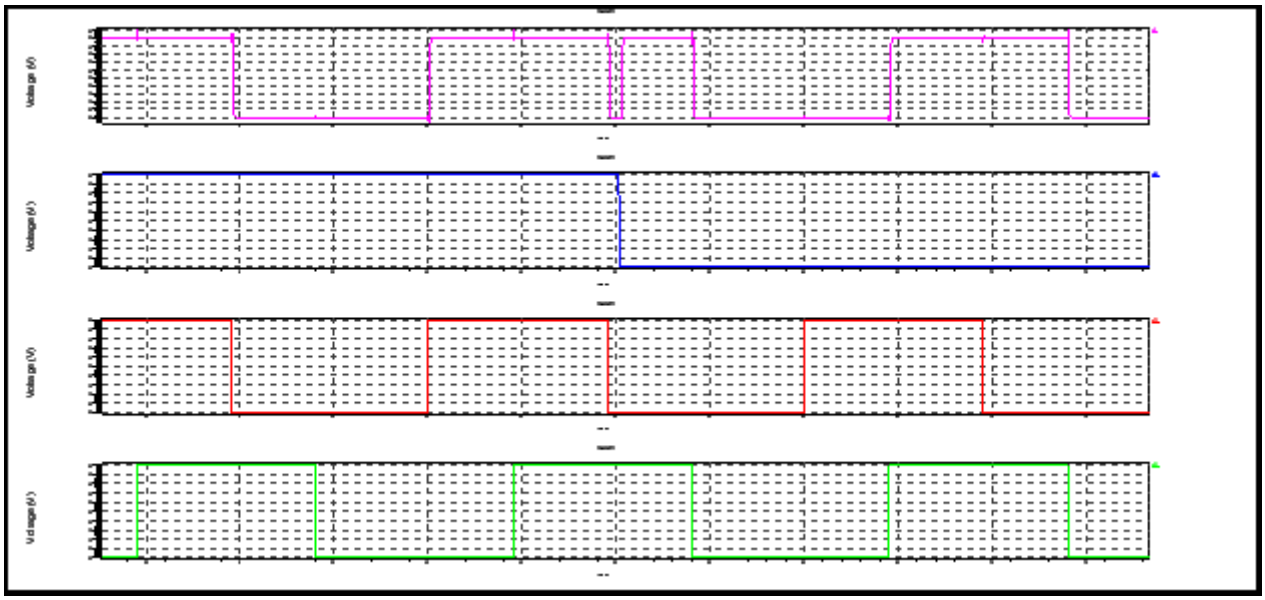

Fig 4.2 Input Signals

INTERNATIONAL JOURNAL FOR INNOVATIVE RESEARCH IN MULTIDISCIPLINARY FIELD ISSN: 2455-0620 Volume - 5, Issue - 11, Nov– 2019
Monthly, Peer-Reviewed, Refereed, Indexed Journal with IC Value: 86.87 Impact Factor: 6.497
Received Date: 15/11/2019 Acceptance Date: 28/11/2019 Publication Date: 30/11/2019

Fig 4.3 Output Signals

Fig 4.4 Input Signals

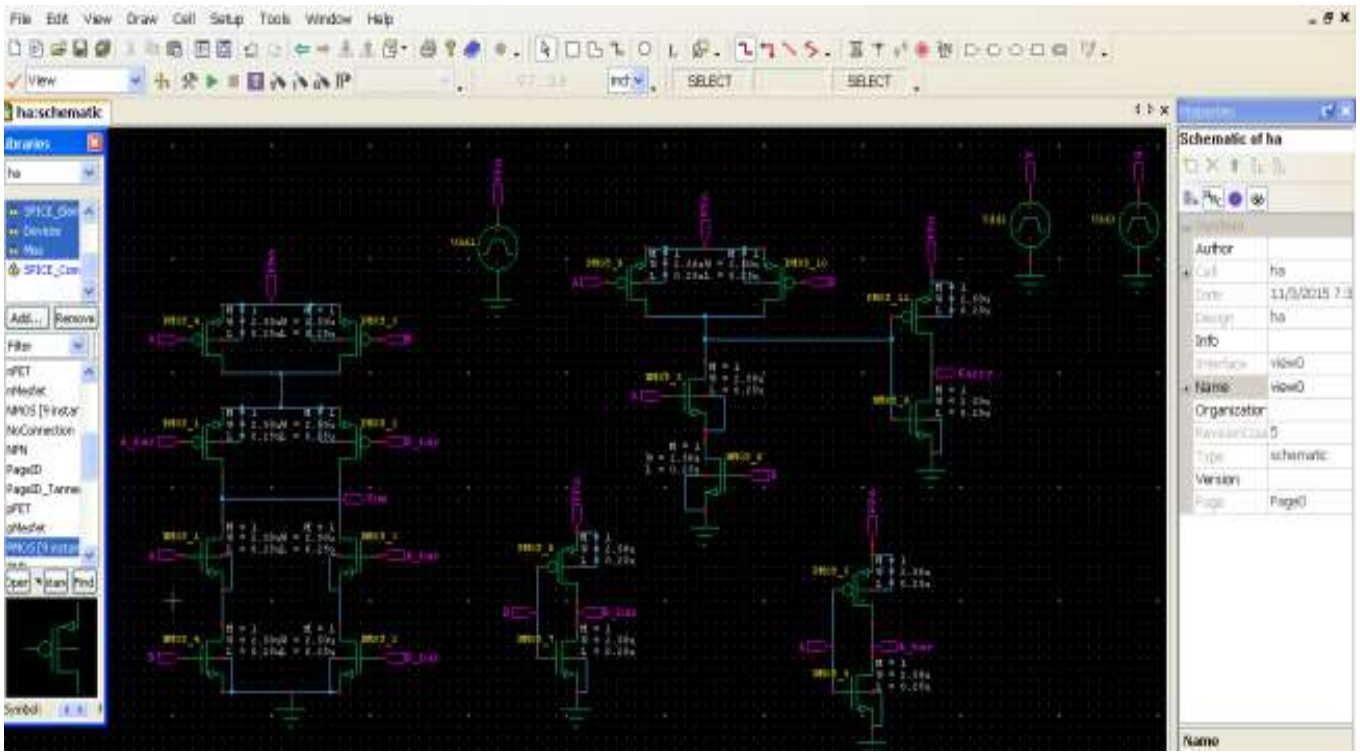

Fig 4.5 Schematic structure of the circuit

Fig 4.6 Output of Recursive adder

5. CONCLUSION:

An efficient implementation of PASTA. Initially, the theoretical foundation for a single-rail wave-pipelined adder is established. Subsequently, the architectural design and CMOS implementations are presented. The design achieves a very simple *n*-bit adder that is area and interconnection-wise equivalent to the simplest adder namely the RCA. Moreover, the circuit works in a parallel manner for independent carry chains, and thus achieves logarithmic average time performance over random input values. The completion detection unit for the proposed adder is also practical and efficient. Simulation results are used to verify the advantages of the proposed approach.

REFERENCES:

1. S.H. Unger, "Double-edge-triggered flip-flops," *IEEE Trans. on Computers*, vol. 30, no. 6,pp. 447-451, June 1981.

2. M. Pedram, Q. Wu and X. Wu, "A new design of double edge triggered flip-flops," *Proc. Asia and South Pacific Design Automation Conference*, pp. 417-421, 1998.
3. W. Chung, T. Lo and M. Sachdev, "A comparative analysis of low-power low-voltage dual-edge-triggered flip-flops," *IEEE Trans. on VLSI Systems*, vol. 10, no. 6, pp. 913-918, December 2002.
4. A.L. Fisher and H.T. Kung, "Synchronizing large VLSI processor arrays," *IEEE Trans. On Computers*, vol. C-34, no. 8, pp. 734-740, August 1985.
5. H.B. Bakoglu, J.T. Walker and J.D. Meindl, "A symmetric clock-distribution tree and optimized high-speed interconnections for reduced clock skew in ULSI and WSI circuit," *Proc. IEEE International Conf. on Computer Design*, pp. 118-122, 1986.
6. E.G. Friedman and S. Powell, "Design and analysis of a hierarchical clock distribution system for synchronous standard cell/macrocell VLSI," *IEEE Journal of Solid-State Circuits*, vol. SC-21, no. 2, pp. 240-246, April 1986.
7. S.Y. Kung and R.J. Gal-Ezer, "Synchronous versus asynchronous computation in VLSI array processors," *Proc. of the SPIE*, vol. 341, pp. 53-65, 1982.
8. E.G. Friedman, "Clock distribution networks in synchronous digital integrated circuits," *Proc. of the IEEE*, vol. 89, no. 5, pp. 665-692, May 2001.
9. G.E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, April 1965.
10. G.D. Wilk, R.M. Wallace and J.M. Anthony, "High-κ gate dielectrics: current status and materials properties considerations," *Journal of Applied Physics*, vol. 89, no. 10, pp. 5243-5275, 2001.
11. S.R. Nassif, "Design for variability in DSM technologies," *Proc. International Symp. On Quality Electronic Design*, pp. 451-454, 2000.
12. K. Bernstein, D.J. Frank, A.E. Gattiker, W. Haensch, B.L. Ji, S.R. Nassif, E.J. Nowak, D.J Pearson and N.J. Rohrer, "High-performance CMOS variability in the 65-nm regime and beyond," *IBM Journal of Research and Development*, vol. 50, no. 4/5, pp. 433-450, 2006.
13. D. Singh, "Prospects for low power microprocessor design," *Keynote Address, Proc. International Workshop on Low Power Design*, October 1994.
14. D. Brooks, V. Tiwari and M. Martonosi, "Wattch: a framework for architectural level power analysis and optimizations," *Proc. International Symp. on Computer Architecture*, pp. 83-94, 2000.
15. A.J. Martin, S.M. Burns, T.K. Lee, D. Borkovic and P.J. Hazewindus, "The first asynchronous microprocessor: the test results," *ACM SIGARCH Computer Architecture News*, vol. 17, no. 4, pp. 95-98, June 1989.
16. S.B. Furber, P. Day, J.D. Garside, N.C. Paver and J.V. Woods, "AMULET1: a micropipelined ARM," *Digest of Papers, IEEE Comp. Society International Conference (COMPCON)*, pp. 476-485, 1994.
17. T. Nanya, Y. Ueno, H. Kagotani, M. Kuwako and A. Takamura, "TITAC: design of a quasi-delay-insensitive microprocessor," *IEEE Design and Test of Computers*, vol. 11, no. 2, pp. 50-63, April 1994.
18. A.J. Martin, A. Lines, R. Manohar, M. Nystrom, P. Penzes, R. Southworth and U. Cummings, "The design of an asynchronous MIPS R3000 processor," *Proc. 17th Conf. on Advanced Research in VLSI*, pp. 164-181, 1997.
19. T. Nanya, A. Takamura, M. Kuwako, M. Imai, T. Fujii, M. Ozawa, I. Fukasaku, Y. Ueno, F. Okamoto, H. Fujimoto, O. Fujita, M. Yamashina and M. Fukuma, "TITAC-2: a 32-bit scalable-delay-insensitive microprocessor," *Proc. HOT Chips IX*, pp. 19-32, 1997.
20. H. van Gageldonk, K. van Berkel, Ad Peeters, D. Baumann, D. Gloor and G. Stegmann, "An asynchronous low power 80C51 microcontroller," *Proc. 4th International Symp. On Advanced Research in Asynchronous Circuits and Systems*, pp. 96-107, 1998.
21. M. Renaudin, P. Vivet and F. Robin, "ASPRO-216: a standard-cell QDI 16-bit RISC asynchronous microprocessor," *Proc. International Symp. on Advanced Research in Asynchronous Circuits and Systems*, pp. 22-31, 1998.
22. S.B. Furber, J.D. Garside, P. Riocreux, S. Temple, P. Day, J. Liu and N.C. Paver, "AMULET2e: an asynchronous embedded controller," *Proc. of the IEEE*, vol. 87, no. 2, pp. 243-256, February 1999.
23. S. Rotem, K. Stevens, R. Ginosar, P. Beerel, C. Myers, K. Yun, R. Kol, C. Dike, M. Roncken and B. Agapiev, "RAPPID: an asynchronous instruction length decoder," *Proc. International Symp. on Advanced Research in Asynchronous Circuits and Systems*, pp. 60-70, 1999.
24. S.B. Furber, D.A. Edwards and J.D. Garside, "AMULET3: a 100 MIPS asynchronous embedded processor," *Proc. International Conf. on Computer Design*, pp. 329-334, 2000.
25. J.D. Garside, W.J. Bainbridge, A. Bardsley, D.M. Clark, D.A. Edwards, S.B. Furber, J. Liu, D.W. Lloyd, S. Mohammadi, J.S. Pepper, O. Petlin, S. Temple and J.V. Woods, "AMULET3i – an asynchronous system-on-chip," *Proc. International Symp. on Advanced Research in Asynchronous Circuits and Systems*, pp. 162-175, 2000.

26. A. Semenov, A.M. Koelmans, L. Lloyd and A. Yakovlev, "Designing an asynchronous processor using Petri nets," *IEEE Micro*, vol. 17, no. 2, pp. 54-64, March-April 1997.
27. M. Lewis and L. Brackenbury, "CADRE: an asynchronous embedded DSP for mobile phone applications," *Design Automation for Embedded Systems*, vol. 6, pp. 451-475, 2002.
28. A.J. Martin, M. Nystrom, K. Papadantonakis, P.I. Penzes, P. Prakash, C.G. Wong, K.S. Ko, B. Lee, E. Ou, J. Pugh, E.-V. Talvala, J.T. Tong and A. Tura, "The Lutonium: a subnanojoule asynchronous 8051 microcontroller," *Proc. 9th International Symp. On Asynchronous Circuits and Systems*, pp. 14-23, 2003.
29. L.A. Plana, P.A. Riocreux, W.J. Bainbridge, A. Bardsley, S. Temple, J.D. Garside and Z.C. Yu, "SPA – a secure Amulet core for smartcard applications," *Microprocessors and Microsystems*, vol. 27, pp. 431-446, 2003.
30. D.E. Muller, "Asynchronous logics and application to information processing," in H. Aiken and W.F. Main (Eds.), *Switching Theory in Space Technology*, Stanford University Press, 1963.
31. R.E. Miller, *Switching Theory Volume 2: Sequential Circuits and Machines*, John Wiley & Sons, New York, 1965.
32. S.H. Unger, *Asynchronous Sequential Switching Circuits*, Wiley-Interscience, New York, 1969.
33. Semiconductor Industry Association's International Technology Roadmap for Semiconductors reports, Available: http://www.itrs.net/reports.html
34. G.F. Bouesse, G. Sicard, A. Baixas and M. Renaudin, "Quasi delay insensitive asynchronous circuits for low EMI," *Proc. 4th International Workshop on Electromagnetic Compatibility of Integrated Circuits*, pp. 27-31, 2004.
35. K. van Berkel, R. Burgess, J. Kessels, Ad Peeters, M. Roncken, and F. Schalij, "A fully
1. asynchronous low-power error corrector for the DCC player," *Proc. 41st IEEE International Solid State Circuits Conference*, pp. 88-89, 1994.