ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
*A Peer Reviewed Open Access International Journal*

# Design of a Parallel Self-Timed Adder Using Recursive Approach

**Koti Reddy Naru**
**Student,**
**SR International Institute of Technology.**

**Mr.K.Kotaiah, M.Tech**
**Asst Prof,**
**SR International Institute of Technology.**

## Abstract:

As technology scales down into the lower nanometer values power, delay, area and frequency becomes important parameters for the analysis and design of any circuits. This brief presents a parallel single-rail self-timed adder. It is based on a recursive formulation for performing multibit binary addition. The operation is parallel for those bits that do not need any carry chain propagation. Thus, the design attains logarithmic performance over random operand conditions without any special speedup circuitry or look-ahead schema. A practical implementation is provided along with a completion detection unit. The implementation is regular and does not have any practical limitations of high fanouts. A high fan-in gate is required though but this is unavoidable for asynchronous logic and is managed by connecting the transistors in parallel. Simulations have been performed using anindustry standard toolkit that verify the practicality and superiority of the proposed approach over existing asynchronous adders.

## Keywords:
CMOS design, digital arithmetic Binary adders, Recursive adder.

## I.INTRODUCTION:

Binary addition is the single most important operation that a processor performs. Most of the addershave been designed for synchronous circuits even though there is a strong interest in clock less circuits[1].Asynchronous circuits do not assume any quantization of time. Therefore, they hold great potentialfor logic design as they are free from several problems of clocked (synchronous) circuits. In principle, logic flow in asynchronous circuits is controlled by On the other hand, wave pipelining (or max-imal ratepipelining) is a technique that can apply pipelined inputs before the outputs are stabilized [7].

The proposed circuit manages automatic single-rail pipelining of the carry inputs separated by propagation and inertial delays of the gates in the circuit path. The remainder of this brief is organized as follows. Section II provides a review of self-timed adders. Section III presents the architectures of PSTA. Section IV presents CMOS implementation of PSTA. Section Vprovides simulation results,. Section VI draws the conclusion.

## II. SELF-TIMED ADDERS:

Self timed refers to logic circuits that depend on timing assumptions for the correct operation. Self-timed adders have the potential to run faster averaged for dynamic data, as early completion sensing can avoid the need for the worst case bundled delay mechanism of synchronous circuits.

### A.Pipelined Adders Using Single-Rail Data Encoding:

The asynchronous Req/Ack handshake can be used to enable theadder block as well as to establish the flow of carry signals. In mostof the cases, a dual-rail carry convention is used for internal bitwiseflow of carry outputs. These dual-rail signals can represent more thantwo logic values (invalid, 0, 1), and therefore can be used to generatebit-level acknowledgment when a bit operation is completed. Finalcompletion is sensed when all bit Acksignals are received (high).The carry-completion sensing adder is an example of a pipelinedadder [8], which uses full adder (FA) functional blocks adapted fordual-rail carry. On the other hand, a speculative completion adder isproposed in [9]. It uses so-called abort logic and early completion toselect the proper completion response from a number of fixed delaylines. However, the abort logic implementation is expensive due tohigh fan-in requirements.

## B.Delay Insensitive Adders Using Dual-Rail Encoding:

Delay insensitive (DI) adders are asynchronous adders that assertbundling constraints or DI operations. Therefore, they can correctlyoperate in presence of bounded but unknown gate and wire delays [2].There are many variants of DI adders, such as DI ripple carryadder (DIRCA) and DI carry look-ahead adder (DICLA). DI addersuse dual-rail encoding and are assumed to increase complexity.Though dual-rail encoding doubles the wire complexity, they canstill be used to produce circuits nearly as efficient as that of thesingle-rail variants using dynamic logic or nMOS only designs. Anexample 40 transistors per bit DIRCA adder is presented in [8] whilethe conventional CMOS RCA uses 28 transistors.Similar to CLA, the DICLA defines carry propagate, generate, andkill equations in terms of dual-rail encoding [8]. They do not connectthe carry signals in a chain but rather organize them in a hierarchicaltree. Thus, they can potentially operate faster when there is long carrychain.A further optimization is provided from the observation that dualrailencoding logic can benefit from settling of either the 0 or 1 path.Dual-rail logic need not wait for both paths to be evaluated. Thus, it is possible to further speed up the carry look-ahead circuitry tosend carry-generate/carry-kill signals to any level in the tree. Thisis elaborated in [8] and referred as DICLA with speedup circuitry(DICLASP).

## III.PARALLEL SELF TIME ADDRS:

In this section, the architecture and theory behind PASTA ispresented. The adder first accepts two input operands to perform halfadditionsfor each bit. Subsequently, it iterates using earlier generatedcarry and sums to perform half-additions repeatedly until all carry bitsare consumed and settled at zero level.

## A. Architecture of PASTA:

The general architecture of the adder is shown in Fig. 1. Theselection input for two-input multiplexers corresponds to the Reqhandshake signal and will be a single 0 to 1 transition denoted bySEL. It will initially select the actual operands during SEL = 0 andwill switch to feedback/carry paths for subsequent iterations usingSEL = 1. The feedback path from the HAs enables the multipleiterations to continue until the completion when all carry signals willassume zero values.

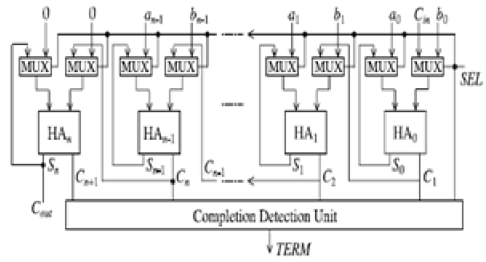



*Fig. 1.Block diagram of PASTA.*

## B.State Diagrams:

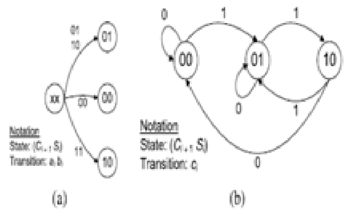



Fig. 2. State diagrams for PASTA. (a) Initial phase. (b) Iterative phase.

In Fig. 2, two state diagrams are drawn for the initial phase and theiterative phase of the proposed architecture. Each state is representedby (Ci+1 Si) pair where Ci+1, Si represent carry out and sum values,respectively, from the ith bit adder block. During the initial phase, thecircuit merely works as a combinational HA operating in fundamentalmode. It is apparent that due to the use of HAs instead of FAs,state (11) cannot appear. During the iterative phase (SEL = 1), the feedback path throughmultiplexer block is activated. The carry transitions (Ci) are allowedas many times as needed to complete the recursion.From the definition of fundamental mode circuits, the presentdesign cannot be considered as a fundamental mode circuit as theinput–outputs will go through several transitions before producing thefinal output. It is not a Muller circuit working outside the fundamentalmode either as internally; several transitions will take place, as shownin the state diagram. This is analogous to cyclic sequential circuitswhere gate delays are utilized to separate individual states [4].

## C.Recursive Formula for Binary Addition:

Let S jiand C ji+1 denote the sum and carry, respectively, for ithbit at the j th iteration. The initial condition ( j= 0) for addition is formulated as follows:

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

$$S_i^0 = a_i \oplus b_i$$

$$C_{i+1}^0 = a_i b_i. \qquad (1)$$

The *j*Th iteration for the recursive addition is formulated by

$$S_i^j = S_i^{j-1} \oplus C_i^{j-1}, \quad 0 \le i < n \qquad (2)$$

$$C_{i+1}^j = S_i^{j-1} C_i^{j-1}, \quad 0 \le i \le n. \qquad (3)$$

The recursion is terminated at kth iteration when the followingcondition is met:

$$C_n^k + C_{n-1}^k + \cdots + C_1^k = 0, \quad 0 \le k \le n. \qquad (4)$$

Now, the correctness of the recursive formulation is inductivelyproved as follows.Theorem 1: The recursive formulation of (1)–(4) will producecorrect sum for any number of bits and will terminate within a finitetime. Proof: We prove the correctness of the algorithm by induction onthe required number of iterations for completing the addition (meetingthe terminating condition).Basis: Consider the operand choices for which no carry propagationis required, i.e., Coi= 0 for i, i [0..n]. The proposed formulationwill produce the correct result by a single-bit computationtime and terminate instantly as (4) is met.Induction: Assume that Cki+1_= 0 for some ith bit at kth iteration.Let l be such a bit for which Ckl+1= 1. We show that it will besuccessfully transmitted to next higher bit in the (k + 1)th iteration.As shown in the state diagram, the kth iteration of lth bit state(Ckl+1, Skl ) and (l + 1)th bit state (Ckl+2, Skl+1) could be in anyof (0, 0), (0, 1), or (1, 0) states. As Ckl+1= 1, it implies thatSkl= 0. Hence, from (3), Ck+1l+1= 0 for any input condition between0 to l bits.We now consider the (l + 1)th bit state (Ckl+2, Skl+1) for kthiteration. It could also be in any of (0, 0), (0, 1), or (1, 0) states.In (k+1)th iteration, the (0, 0) and (1, 0) states from the kth iterationwill correctly produce output of (0, 1) following (2) and (3). For(0, 1) state, the carry successfully propagates through this bit levelfollowing (3).Thus, all the single-bit adders will successfully kill or propagatethe carries until all carries are zero fulfilling the terminatingcondition.The mathematical form presented above is valid under the conditionthat the iterations progress synchronously for all bit levels andthe required input and outputs for a specific iteration will also being synchrony with the progress of one iteration. In the next section,we present an implementation of the proposed architecture which issubsequently verified using simulations.

## IV.IMPLEMENTATION OF PSTA:

A CMOS implementation for the recursive circuit is shown inFig. 3. For multiplexers and AND gates we have used TSMC libraryimplementations while for the XOR gate we have used the faster tentransistor implementation based on transmission gate XOR to matchthe delay with AND gates [4]. The completion detection following (4)is negated to obtain an active high completion signal (TERM). Thisrequires a large fan-in n-input NOR gate. Therefore, an alternativemore practical pseudo-nMOS ratio-ed design is used. The resultingdesign is shown in Fig. 3(d). Using the pseudo-nMOS design, thecompletion unit avoids the high fan-in problem as all the connectionsare parallel. The pMOS transistor connected to VDD of this ratio-eddesign acts as a load register, resulting in static current drain whensome of the nMOS transistors are on simultaneously. In additionto the Cis, the negative of SEL signal is also included for theTERM signal to ensure that the completion cannot be accidentallyturned on during the initial selection phase of the actual inputs.It also prevents the pMOS pull up transistor from being always on.Hence, static current will only be flowing for the duration of theactual computation.

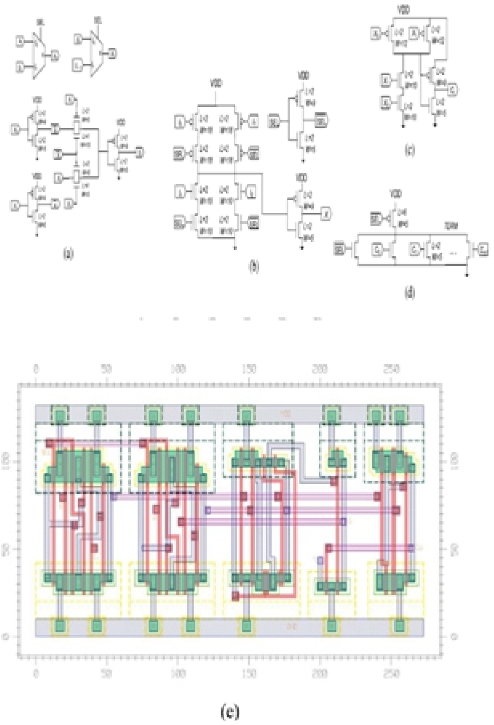


Fig.3.CMOSimplementation of PASTA. (a) Single-bit sum module. (b) 2×1 MUX for the 1 bit adder. (c) Single-bit carry module. (d) Completion signaldetection circuit.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
*A Peer Reviewed Open Access International Journal*

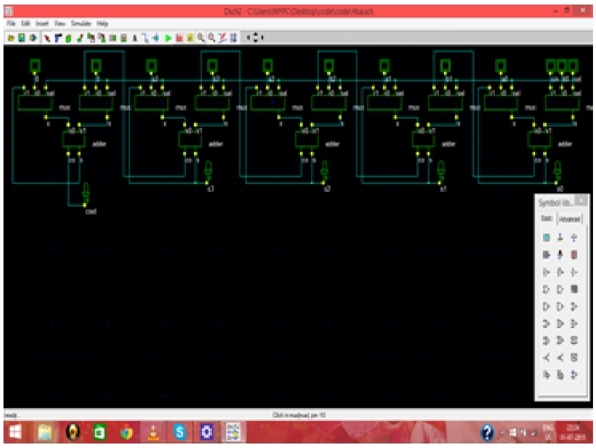
VLSI layout has also been performed [Fig. 3(e)] for a standardcell environment using two metal layers. The layout occupies270 λ ×130 λ for 1-bit resulting in 1.123 Mλ2 area for 32-bit. Thepull down transistors of the completion detection logic are includedin the single-bit layout (the T terminal) while the pull-up transistoris additionally placed for the full 32-bit adder. It is nearly doublethe area required for RCA and is a little less than the most of thearea efficient prefix tree adder, i.e., Brent–Kung adder (BKA).

## V.SIMULATION RESULTS:

In this section, we present simulation results for PASTA adders using DSCH and microwind tool version 3.0. For implementation of other adders, we have used standard library implementations of the basic gates. The custom adders such as DIRCA/DICLASP are implemented based on their most efficient designs from [8]. Initially, we show how the present design of PASTA can effectively perform binary addition for different temperatures and process corners to validate the robustness under manufacturing and operational variations.

The following figure shows the schematics, layouts and timing diagram of 4-bit Parallel Adder when initial phase: SEL=0 and iterative phase: SEL=1.On the other hand, PASTA performs best among the self-timed adders. PASTA performance is comparable with the best case performances of conventional adders. Effectively, it varies between one and four times that of the best adder performances. It is even shown to be the fastest for 0.35 µm process [11]. For average cases, PASTA performance remains within two times to that of the best average case performances while for the worst case, it behaves similar to the RCA.

Note that, PASTA completes the first iteration of the recursive formulation when “SEL = 0.” Therefore, the best case delay represents the delay required to generate the TERM signal only and of the order of picoseconds. Similar overhead is also present in dual-rail logic circuits where they have to be reset to the invalid state prior to any computation. The dynamic/ nMOS only designs require a precharge phase to be completed during this interval [12]. Another interesting observation is that the performances of the combinational adders and PASTA improve with the decreasing process width and VDD values while the performance of dual-rail adders decreases with scaling down the technology.



(a)

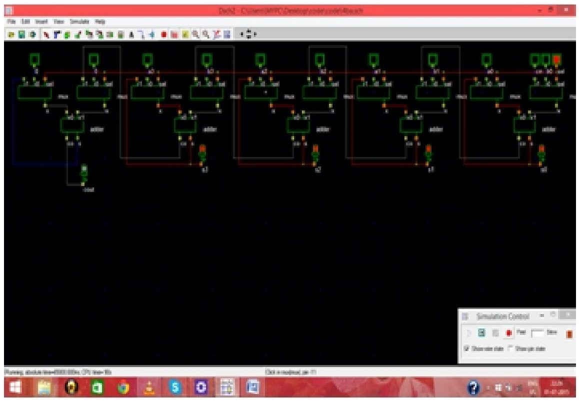


Fig. 4.Schematic implementation of PASTA. (a) 4-bit sum module. (b) 4-bit Simulated results

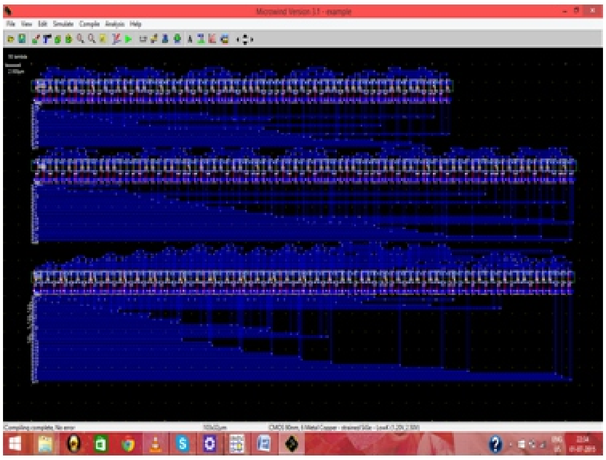


Fig. 5.Layout implementation of 4-bit PASTA.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

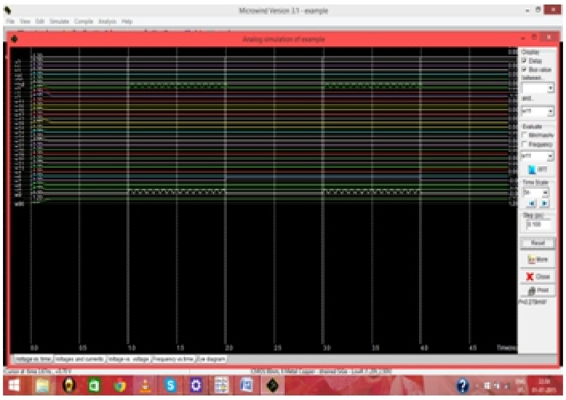


Fig. 6. Timing diagram of 4-bit PASTA.

Fig. 7. Eye diagram of 4-bit PASTA.

## VI. CONCLUSION:

This brief presents an efficient implementation of a 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] D. Geer, "Is it time for clockless chips? [Asynchronous processorchips]," IEEE Comput., vol. 38, no. 3, pp. 18–19, Mar. 2005.

[2] J. Sparsø and S. Furber, Principles of Asynchronous Circuit Design.Boston, MA, USA: Kluwer Academic, 2001.

[3] P. Choudhury, S. Sahoo, and M. Chakraborty, "Implementation of basicarithmetic operations using cellular automaton," in Proc. ICIT, 2008,pp. 79–80.

[4] M. Z. Rahman and L. Kleeman, "A delay matched approach forthe design of asynchronous sequential circuits," Dept. Comput.Syst.Technol., Univ. Malaya, Kuala Lumpur, Malaysia, Tech. Rep. 05042013,2013.

[5] M. D. Riedel, "Cyclic combinational circuits," Ph.D. dissertation, Dept.Comput. Sci., California Inst. Technol., Pasadena, CA, USA, May 2004.

[6] R. F. Tinder, Asynchronous Sequential Machine Design and Analysis:A Comprehensive Development of the Design and Analysis ofClock-Independent State Machines and Systems. San Mateo, CA, USA:Morgan, 2009.

[7] W. Liu, C. T. Gray, D. Fan, and W. J. Farlow, "A 250-MHz wavepipelined adder in 2-μm CMOS," IEEE J. Solid-State Circuits, vol. 29,no. 9, pp. 1117–1128, Sep. 1994.

[8] F.-C. Cheng, S. H. Unger, and M. Theobald, "Self-timed carry-lookaheadadders," IEEE Trans. Comput., vol. 49, no. 7, pp. 659–672, Jul. 2000.

[9] S. Nowick, "Design of a low-latency asynchronous adder using speculativecompletion," IEE Proc. Comput. Digital Tech., vol. 143, no. 5,pp. 301–307, Sep. 1996.

[10] N. Weste and D. Harris, CMOS VLSI Design: A Circuits and SystemsPerspective. Reading, MA, USA: Addison-Wesley, 2005.

[11] C. Cornelius, S. Koppe, and D. Timmermann, "Dynamic circuit techniquesin deep submicron technologies: Domino logic reconsidered," inProc. IEEE ICICDT, Feb. 2006, pp. 1–4.

[12] M. Anis, S. Member, M. Allam, and M. Elmasry, "Impact of technologyscaling on CMOS logic styles," IEEE Trans. Circuits Syst., AnalogDigital Signal Process., vol. 49, no. 8, pp. 577–588, Aug. 2002.