International Journal of Advance Research in Science and Engineering
Vol. No.5, Issue No. 03, March 2016
www.ijarse.com
IJARSE
ISSN 2319 - 8354

DESIGN OF HIGH SPEED PASTA

Ms. V.Vivitha[1], Ms. R.Niranjana Devi[2], Ms. R.Lakshmi Priya[3]

[1,2,3]M.E(VLSI DESIGN) , Theni Kammavar Sangam College of Technology, Theni,(India)

ABSTRACT

Parallel Asynchronous Self-Timed Adder (PASTA) is performing multi-bit binary addition based on the recursive formulation. PASTA does not need any carry chain propagation. The design is provided with the completion and detection unit and the design attains the logarithmic performance without any speedup circuitry or look-ahead schema. The implementation of PASTA has no limitations for high fan-outs. For asynchronous logic the high fan-in gate is required and which is managed by connecting the transistors in parallel. The simulations have been performed in CADENCE 180nm technology.

Index Terms*: *Asynchronous circuits, binary adders, CMOS design, digital arithmetic.

I. INTRODUCTION

Addition is an primary operation for any digital system. Binary addition is the primary operation that processor performs. The performance of the processor is significantly influenced by the speed of their adders. The Ripple Carry Adder and the Carry Look-ahead Adders are synchronous adders. The synchronous circuits need clock pulses to perform the operation. The clockless processors are also called as self-timed or asynchronous processors [1], refers to the logic circuit that depend on and/or engineer assumptions for the correct operation. The asynchronous circuits hand-off the results of their work, rather than waiting for a clock tick, as soon as synchronous circuits are finished their process. Logic flow in asynchronous circuit is controlled by a request-acknowledgment handshaking protocol to establish a pipeline in the absence of clocks. Explicit hand shaking blocks for small elements such as bit adders, are expensive. Therefore, it is implicitly and efficiently managed by using dual rail carry propagation in adders. Thus the asynchronous adders are either based on the full dual-rail encoding of all signals or the pipelined operation using single-rail data encoding and dual-rail carry representation for acknowledgments. The Delay Insensitive (DI) adders are asynchronous adders. There may variants of DI adders, such as Delay Insensitive Ripple Carry Adder (DIRCA), Delay Insensitive Carry look-Ahead Adder (DICLA)[8] Which uses dual-rail encoding. It add robustness to the circuit while construction.

Thus an Asynchronous Parallel Self-Timed Adder (PASTA) [3] is proposed and designed. The design of PASTA is uses half adders along with multiplexers requiring minimal interconnections. The design works in a parallel manner for the independent carry chain blocks.

The conventional adder like single rail cyclic asynchronous sequential adder, the feedback employs through XOR logic gates [4]. Wave pipelining is a technique that can apply pipelined inputs before the outputs are stabilized [7]. The proposed circuit uses the single-rail wave-pipelined approach for the carry input, which is separated by the propagation and inertial delay of gates in the circuit path.

II. BACKGROUND

Asynchronous circuits refer to logic circuits do not depend on an external clock for circuit operation. They depend on and/or engineer timing assumption for the correct operation. The worst case bundled delay mechanism of synchronous circuits has been avoided by asynchronous adders, as early completion sensing in these circuits have the potential to run faster for dynamic data.

2.1 Single-Rail Data Encoding in Pipelined Adders

To enable the adder block and establish the flow of carry signal the asynchronous Req/Ack handshake can be used. For internal bitwise flow of carry outputs, a dual-rail carry convention is used. More than two logic values (invalid,0,1) can represent these dual-rail signals and therefore can be used to generate a bit-level acknowledgment when a bit operation is completed. When all bit Ack signals are received (high), final completion is sensed. Examples include the carry-completion sensing adder [8] and a speculative completion adder is proposed in [9]. The adder in [8] is an of a pipelined adder, which uses full adder (FA) functional blocks adapted for dual-rail carry and the adder in [9] uses so-called abort logic and early completion to select the proper completion response from a number of fixed delay lines. The drawback is the expensive implementation of the abort logic due to high fan-in requirements.

2.2 Dual-Rail Encoding in Delay Insensitive Adders

Bundling constraints are asserted in Delay insensitive (DI) asynchronous adders. In presence of bounded but unknown gate and wire delays [2] they can work correctly. DI ripple carry adder (DIRCA) and DI carry look-ahead adder (DICLA) are examples for DI adders. The conventional CMOS RCA uses 28 transistors whereas DIRCA adder is presented in [8] uses 40 transistors per bit. The DICLA defines carry propagate, generate, and kill an equation in terms of dual-rail encoding [8] that is similar to CLA. In DICLA carry signals are organized as a hierarchical tree. Therefore for long carry chain, they can potentially operate faster.

III. DESIGN OF PASTA

3.1 Architecture

The architecture and theory behind Asynchronous Parallel Adder using recursive approach is presented. The general architecture of the adder is shown in Fig. 1. Req handshake signal is required for the selection input for two-input multiplexers from a single 0 to 1 transition denoted by SEL. During the initial phase SEL = 0, the actual operands are initially selected and will switch to feedback/carry paths for iterative phase using SEL = 1.Multiple recursions occur through the feedback path and continue until all the carry values assume zero values to achieve completion.

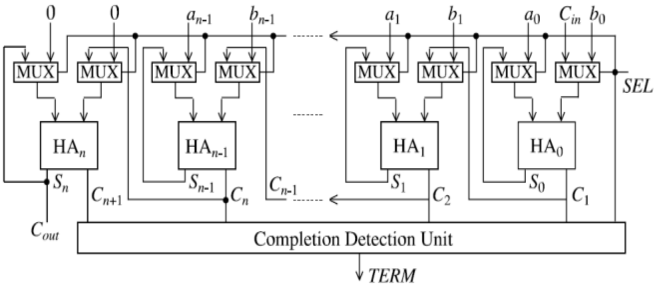


Fig.1.General block diagram of PASTA.

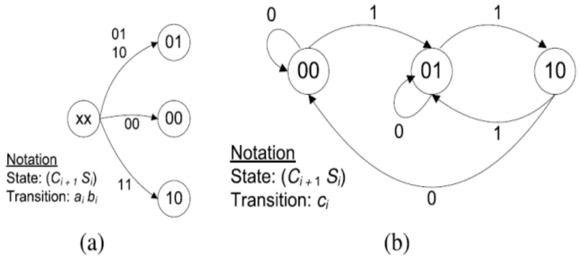


Fig. 2. State diagrams for PASTA. (a) Initial phase. (b) Iterative phase.

3.2 State Diagmas

State diagrams are drawn for the initial phase and the Recursive phase in Fig. 2. Representation of each state is given by (Ck+1 Sk) pair where Ck+1, Sk represent carry out and sum values, respectively, from the kth bit adder block. The circuit merely works as a combinational HA operating in fundamental mode, during the initial phase. State (11) cannot appear, due to the use of HAs instead of FAs. When SEL = 1, the feedback path through multiplexer block is activated. To complete the recursion, the carry transitions (Ck) are allowed as many times as needed. This design cannot be considered as a fundamental mode circuit as the input–outputs will go through several transitions before producing the final output.

This is analogous to cyclic sequential circuits where gate delays are utilized to separate individual states [4].

3.3 Recursive Formula for Binary Addition

Let $S_j{}^i$ and $C_j{}^{i+1}$ denote the sum and carry, respectively, for *i*th bit at the *j* th iteration. The initial condition (*j* = 0) for addition is formulated as follows:

$$S^0{}_i = a_i \oplus b_i$$

$$C_0{}^{i+1} = a_i \,.\, b_i \,. \qquad (1)$$

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

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

$$C_j{}^{i+1} = S_{j-1}{}^i \, C_{j-1}{}^i, \; 0 \le i \le n. \qquad (3)$$

The recursion is terminated at *k*th iteration when the following condition is met:

$$C_k{}^n + C_k{}^{n-1} + \cdot \;\; \cdot \;\; \cdot \; + C_k{}^1 = 0, \; 0 \le k \le \text{n}. \; (4)$$

Fast adder will now be designed using the formulae presented in equations(1)-(4)

Theorem 1: The recursive formulation of (1)–(4) will produce correct sum for any number of bits and will terminate within a finite time.

Proof: We prove the correctness of the algorithm by induction on the required number of iterations for completing the addition (meeting the terminating condition).

Basis: Consider the operand choices for which no carry propagation is required, i.e., $C^0_i= 0$ for $\forall\, i$, $i \in$ [0..*n*].

The proposed formulation will produce the correct result by a single-bit computation time and terminate instantly as (4) is met.

Induction: Assume that C_k^{i+1}_= 0 for some *i*th bit at *k*th iteration. Let *l* be such a bit for which $C_k^{l+1}= 1$. We show that it will be successfully transmitted to next higher bit in the *(k* + 1*)*th iteration.

As shown in the state diagram, the *k*th iteration of *l*th bit state (C_k^{l+1}, S_k^{l}) and *(l* + 1)th bit state (C_k^{l+2}, S_k^{l+1}) could be in any of *(*0, 0*)*, *(*0, 1*)*, or *(*1, 0*)* states. As $C_k^{l+1}= 1$, it implies that $S_k^{l}= 0$. Hence, from (3), $C_{k+1}^{\ l+1}= 0$ for any input condition between0 to *l* bits.

We now consider the *(l* + 1*)*th bit state $(C_k^{l+}2, S_k^{\ l+1})$ for *k*th iteration. 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 *k*th iteration

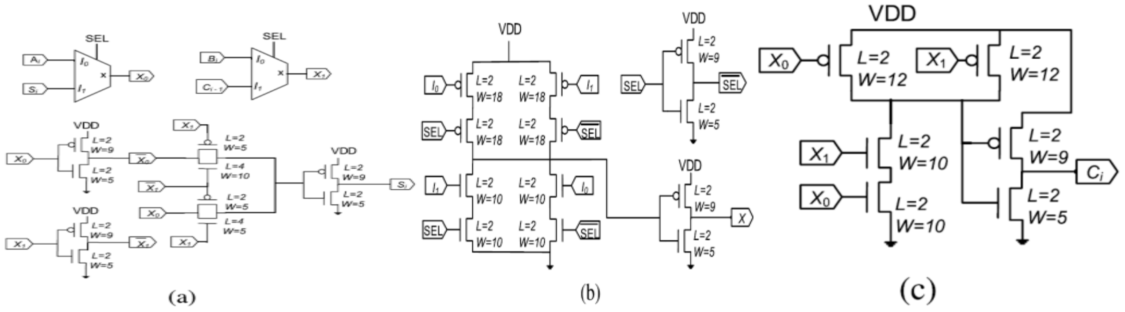
will correctly produce output of *(*0, 1*)* following (2) and (3). For*(*0, 1*)* state, the carry successfully propagates through this bit level following (3). Thus, all the single-bit adders will successfully kill or propagate the carries until all carries are zero fulfilling the terminating condition.

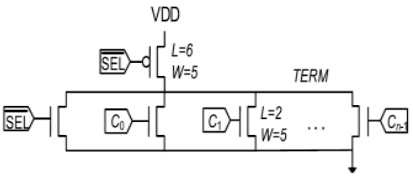
The mathematical form presented above is valid under the condition that the iterations progress synchronously for all bit levels and the required input and outputs for a specific iteration will also be in synchrony with the progress of one iteration. In the next section, we present an implementation of the proposed architecture which is subsequently verified using simulations.

IV. CMOS IMPLEMENTATION AND SIMULATION

A CMOS implementation for the recursive circuit is shown in Fig. 3. For multiplexers and AND gates we have used CADENCE Virtuoso library implementations while for the XOR gate we have used the faster ten transistor implementation based on transmission gate XOR to match the delay with AND gates [4]. The completion detection following (4) is negated to obtain an active high completion signal.

Using the pseudo-nMOS design, the completion unit avoids the high fan-in problem as all the connections are parallel.





(d)

Fig. 3. 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.

The pMOS transistor connected to *V*DD of this ratio-ed design acts as a load register, resulting in static current drain when some of the nMOS transistors are on simultaneously. The simulation of the proposed design is carried out by Using cadence virtuso spectre circuit simulator.

A.8-bit PASTA

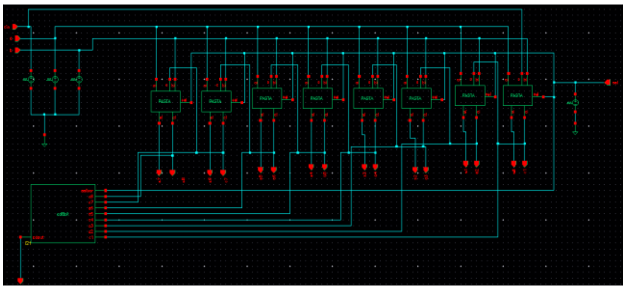


Fig 4(a). Design of 8-bit PASTA

Output waveform for 8-bit PASTAr is shown below.

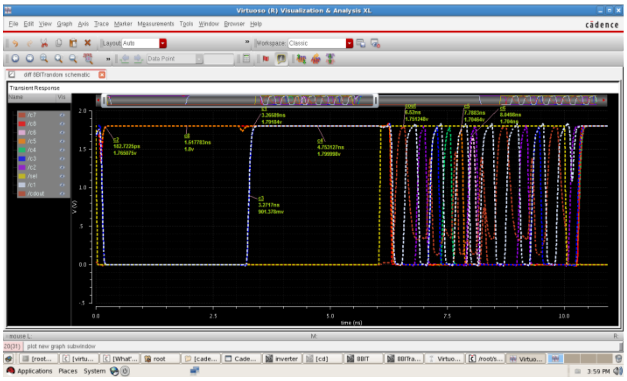

Virtuoso (R) Visualization & Analysis XL
File Edit View Graph Axis Trace Marker Measurements Tools Window Browser Help
cādence
Transient Response
V (V)
time (ns)
Applications Places System

Fig 4(b). Output for 8-bit PASTA

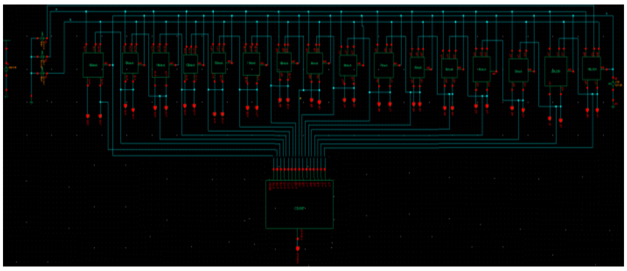


Fig 5(a). Design of 16-bit PASTA

Output waveform for 8-bit PASTA is shown below.

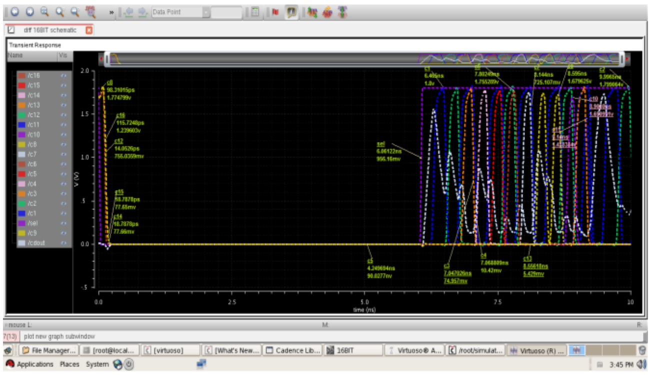


Fig 5(b). Output for 16-bit PASTA

Table 1: Parameter Analysis Comparison of PASTA

Adder	Bits	Delay in (ns)	Average power	Transistor count
DIRCA	8-bit	26.4	58.54 mW	336
	16-bit	68.8	120.4 mW	672
PASTA	8-bit	4.24	740.2 µW	236
	16-bit	9.13	958.9 µW	468

V. CONCLUSION

An efficient adder PASTA was implemented using various techniques in order to get better performance. For independent carry chains the circuit works in a parallel manner. Over random input values logarithmic average time performance is achieved. We also obtained practical and efficient completion detection unit for this proposed adder. Compare to DIRCA we could obtain the power to be reduced by 57.8mW for 8-bit PASTA and 119.44mW for 16-bit PASTA which contributes for a greater efficiency.

PASTA better performance in delay, power and reduction in area. The design achieves a very simple 16-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. Simulation results are used to verify the advantages of the proposed approach. The proposed method is implemented using digital CADANCE environment

REFERENCES

[1] D. Geer, "*Is it time for clockless chips? [Asynchronous processor chips]," 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 basic arithmetic operations using cellular automaton," in Proc. ICIT,* 2008, pp. 79–80.

[4] M. Z. Rahman and L. Kleeman, *"A delay matched approach for the 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 of Clock-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 wave pipelined 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 carrylookahead adders," IEEE Trans. Comput., vol. 49, no. 7, pp. 659–672,* Jul. 2000.

[9] S. Nowick, *"Design of a low-latency asynchronous adder using speculative completion," 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 Systems Perspective. Reading, MA, USA: Addison-Wesley,* 2005.

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

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