International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 13
September 2016

VLSI Implementation of Self Time Adder Using Recursive Approach

***A.RADHIKA** ****Mr.M.RANJITH**

*M.TECH ,Dept of ECE, VAAGDEVI COLLEGE OF ENGINEERING

Warangal

**Asst. prof Dept of ECE, VAAGDEVI COLLEGE OF ENGINEERING

Warangal

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:

A study of the operations performed by an ARM processor's ALU revealed that additions constituted nearly 80% of them. About 72% of the instructions of a prototype RISC machine resulted in addition/subtraction operations. In fact, addition was found to be the most frequently encountered operation amongst a set of real-time digital signal processing benchmarks. In general, integer addition plays a very important and dominant role in digital computer systems.

In this work, we shall extensively consider adder cells synthesised using various approaches that form the fundamental datapath elements, and evaluate their performance based on the carry-ripple or ripple carry adder (RCA) topology. The impact of a dual-bit adder cell in reducing the delay metric of the basic adder topology is investigated and the bottleneck in extending the hierarchy further is discussed. Single-bit17 and dual-bit18 adders based on homogeneous and heterogeneous DI data encoding schemes are considered and a comparative analysis is performed. Also, the usefulness of a hybrid combination of single-bit and dual-bit adders within a RCA structure is studied, which is found to

feature only a minor optimization potential. Finally, the concept of redundant logic insertion, introduced with the aim of further minimising the delay of the addition operation is elucidated through case studies. In general, it could help in minimising the latency of iterative logic circuits.

A majority of the present-day digital systems are clock based or synchronous, which assume that signals are binary and time is discrete. In general, synchronous systems comprise a number of subsystems that change from one state to another depending on a global clock signal, with flip-flops (registers) being used to store the different states of the subsystems. A conventional synchronous system is portrayed by figure 1.1. The state updates within the registers are carried out on the rising edge (positive edge) or falling edge (negative edge) of the global clock – single edge triggering. The state of the global clock permits either data loading or data storage. Since the overall clock utilization is only 50% for single edge triggered systems, double edge triggered flip-flops were subsequently proposed in the literature with the motive of increasing the system throughput as data can be loaded on both the rising and falling clock edges and data is retained when the clock signal does not toggle [1] [2]. However, this usually comes at the expense of a larger

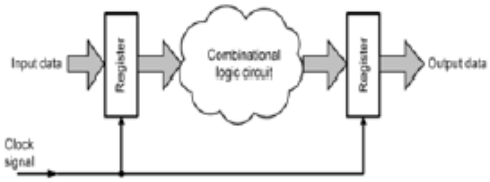

Fig 1.1: A typical synchronous system stage

Silicon footprint due to greater number of transistors and more interconnects for the dual edge triggered flip-flop and consequently leads to more power consumption. Preserving the original data rate as that of single edge triggered flip-flop designs whilst operating at half the system clock frequency might be helpful in

reducing the dynamic power dissipation as the transitions could be reduced by half, but eventually this may be offset by more leakage power dissipation [2], which is becoming dominant in deep submicron technologies Moreover, this mechanism tends to forego the advantages associated with single edge triggering in that its set-up and hold times are larger compared to conventional flip-flops and any deviation from its 50% duty cycle can lead to timing failures in critical paths upsetting the system behaviour [3]. In addition, it is more sensitive to noise apart from introducing complexity in system design and as such, the specification on jitter tolerance is more stringent which complicates the design of the system phase lock loop. As a result, synchronous designs with rising or falling edge triggering have been predominant, being the mainstream of digital system architectures; nevertheless, it is becoming increasingly difficult to overcome some fundamental limitations inherent in this approach.

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].Asyn-chronous 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.

The above projections tend to forecast and necessitate a considerable shift in the design paradigm from conventional synchronous logic to asynchronous logic, as the latter benefits owing to its ability to tolerate supply voltage, process parameter and temperature variations [15]. Due to the absence of a global clock reference, asynchronous circuits tend to have better noise and electro-magnetic compatibility properties than synchronous circuits. Also, they feature greater modularity permitting

convenient design reuse . Asynchronous operation by itself does not imply low power, but often suggests low power opportunities based on the observation that asynchronous circuits only consume power when and where active..The recent demonstration of the potential advantages of the world's first 8-bit physically flexible asynchronous microprocessor design over a synchronous flexible version in terms of power and noise figures by Karaki et al. from Seiko Epson's Technology Platform Research Centre, which utilizes 4-phase handshaking and quasi-delay-insensitive design style, endorses the future of self timed design techniques for even unconventional electronics.

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 pipe-linedadder [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 in-crease 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 transis-tors.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.

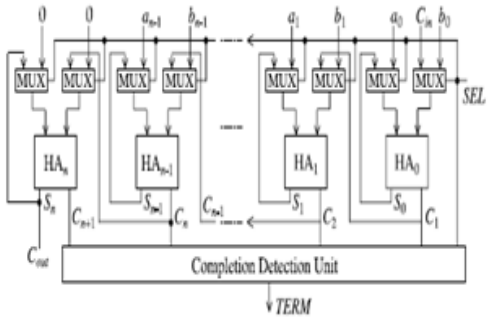

Fig. 1.Block diagram of PASTA.

B.State Diagrams:

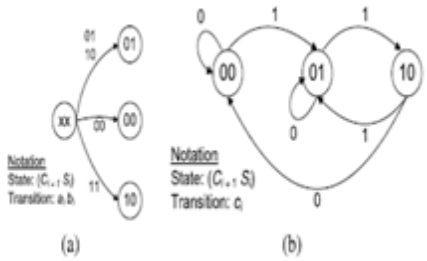

Fig. 2. State diagrams for PASTA. (a) Initial phase. (b) Iterative phase

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:

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

The jTh iteration for the recursive addition is formulated by

$$S_i^j = S_i^{j-1} \oplus C_i^{j-1}, \quad 0 \le i < n \quad (2)$$
$$C_{i+1}^j = S_i^{j-1} C_i^{j-1}, \quad 0 \le i \le n. \quad (3)$$

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

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

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., C0i= 0 for i, i [0..n]. The proposed formulationwill produce the correct result by a single-bit computationtime and ter-minate 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 besuccess-fully transmitted to next higher bit in the (k + 1)th itera-tion. 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 kthit-eration. 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 pre-sented above is valid under the conditionthat the itera-tions progress synchronously for all bit levels andthe required input and outputs for a specific iteration will also being synchrony with the progress of one itera-tion. In the next section,we present an implementation of the proposed architecture which issubsequently verified using simulations.

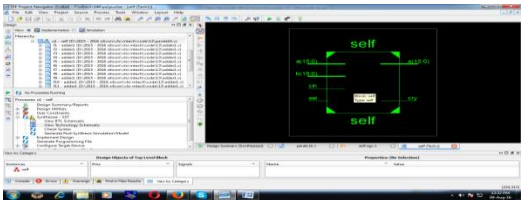

Fig. 3 Block Diagram

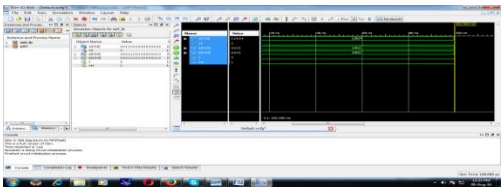

Fig. 4. Simulation Result

TABLE: COMPARISON OF DELAY IN BETWEEN EXISTING AND PROPOSED SYSTEM

SYSTEM	DELAY
Existing system	38.665ns
Proposed system	21.227ns

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 car-ry chains, and thus achieves

logarithmic average time performance over random input values. The comple-tion detection unit for the proposed adder is also prac-tical and efficient. Simulation results are used to verify the advantages of the proposed approach.

REFERENCES: [1] D. Geer, "Is it time for clockless chips? [Asynchro-nous 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, "Im-plementation of basicarithmetic operations using cel-lular 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. Tech-nol., 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. Sol-id- 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 Cir-cuits and SystemsPerspective. Reading, MA, USA: Ad-dison- Wesley, 2005.

AUTHOR 1:-

* A.RADHIKA completed her Btech in GANAPATHY ENGINEERING COLLEGE in 2014 and completed Tech in VAAGDEVI COLLEGE OF ENGINEERING

AUTHOR 2:-

**Mr.M.RANJITH is working as Assistant. professor in Dept of ECE, VAAGDEVI COLLEGE OF ENGINEERING