International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 5, Issue 3 (May-June 2017), PP. 90-95

ADVANCED DIGITAL DESIGN OF CARRY SKIP ADDER WITH HYBRID METHOD FOR FIELD PROGRAMMABLE GATE ARRAY

[1] ELSA THOMAS, [2] NIDIYA HABEEB
[1,2] ELECTRONICS AND COMMUNICATION ENGINEERING
[1]MUSALIAR COLLEGE OF ENGINEERING AND TECHNOLOGY
[2]MUSALIAR COLLEGE OF ENGINEERING AND TECHNOLOGY
PATHANAMTHITTA, KERALA, INDIA
1elsathomas33@gmail.com

Abstract— **A hybrid variable latency carry skip adder (CSKA) structure that has a higher speed yet lower energy consumption compared with the conventional one is presented in this paper. Hybrid variable latency extension of the proposed structure, lowers the power consumption without considerably impacting the speed, is presented. This extension utilizes a modified parallel structure for increasing the slack time, and hence, enabling further voltage reduction. The modified structure make use of Han-Carlson adder. The proposed structures are assessed by comparing their speed, power, and energy parameters with those of other adders. The results that are obtained on simulations reveal, that improvements are there in the delay and energy compared with those of the conventional CSKA. In addition, the power–delay product was the lowest among the structures considered in this paper. Simulations on the proposed hybrid variable latency CSKA reveal reduction in the power consumption compared with the latest works in this field while having a reasonably high speed.**

Keywords— Carry skip adder (CSKA), energy efficient, high performance, hybrid variable latency adders, voltage scaling.

I. INTRODUCTION

Adders are a key building block in arithmetic and logic units (ALUs) and hence increasing their speed and reducing their power/energy consumption strongly affect the speed and power consumption of processors. It is highly desirable to achieve higher speeds at low-power/energy consumptions, which is a challenge for the designers of general purpose processors.

One of the effective techniques to lower the power consumption of digital circuits is to reduce the supply voltage due to quadratic dependence of the switching energy on the voltage. Moreover, the subthreshold current, which is the main leakage component in OFF devices, has an exponential dependence on the supply voltage level through the drain-induced barrier lowering effect . Depending on the amount of the supply voltage reduction, the operation of ON devices may reside in the super threshold, near-threshold, or subthreshold regions. Working in the super threshold region provides us with lower delay and higher switching and leakage powers compared with the near/subthreshold regions. In the subthreshold region, the logic gate delay and leakage power exhibit exponential dependences on the supply and threshold voltages. Moreover, these voltages are (potentially) subject to process and environmental variations in the nano scale technologies. The variations increase uncertainties in the aforesaid performance parameters. In addition, the small subthreshold current causes a large delay for the circuits operating in the subthreshold region.

Recently, the near-threshold region has been considered as a region that provides a more desirable tradeoff point between delay and power dissipation compared with that of the sub threshold one, because it results in lower delay compared with the subthreshold region and significantly lowers switching and leakage powers compared with the super threshold region. In addition, near-threshold operation, which uses supply voltage levels near the threshold voltage of transistors , suffers considerably less from the process and environmental variations compared with the sub threshold region.

The dependence of the power (and performance) on the supply voltage has been the motivation for design of circuits with the feature of dynamic voltage and frequency scaling. In these circuits, to reduce the energy consumption, the system may change the voltage (and frequency) of the circuit based on the workload requirement . For these systems, the circuit should be able to operate under a wide range of supply voltage levels. Of course, achieving higher speeds at lower supply voltages for the computational blocks, with the adder as one the main components, could be crucial in the design of high-speed, yet energy efficient, processors.

In addition to the knob of the supply voltage, one may choose between different adder structures/families for optimizing power and speed. There are many adder families with different delays, power consumptions, and area usages. Examples include ripple carry adder (RCA), carry increment adder (CIA), carry skip adder (CSKA), carry select adder (CSLA), and parallel prefix adders (PPAs). The RCA has the simplest structure with the smallest area and power consumption but with the worst critical path delay. In the

CSLA, the speed, power consumption, and area usages are considerably larger than those of the RCA. The PPAs, which are also called carry look-ahead adders, exploit direct parallel prefix structures to generate the carry as fast as possible . There are different types of the parallel prefix algorithms that lead to different PPA structures with different performances. As an example, the Kogge–Stone adder (KSA) is one of the fastest structures but results in large power consumption and area usage.

The CSKA, which is an efficient adder in terms of power consumption and area usage. The critical path delay of the CSKA is much smaller than the one in the RCA, whereas its area and power consumption are similar to those of the RCA. In addition, the power-delay product (PDP) of the CSKA is smaller than those of the CSLA and PPA structures . In addition, due to the small number of transistors, the CSKA benefits from relatively short wiring lengths as well as a regular and simple layout. The comparatively lower speed of this adder structure, however, limits its use for high-speed applications.

A. OBJECTIVE

In this paper, a structure, based on the variable latency technique, which in turn lowers the power consumption without considerably impacting the CSKA speed, is presented. To the best of our knowledge, no work concentrating on the design of (hybrid) variable latency CSKA structures modified using Han-Carlson parallel prefix adder have been reported in the literature. Hence, the contributions of this work can be summarized as follows.

- Proposing a hybrid variable latency CSKA structure based on the extension of the CI-CSKA, by replacing some of the middle stages in its structure with a PPA, which is modified in this work.
- The modification is achieved by making use of Han-Carlson parallel prefix adder .

II. EXISTING SYSTEM

A. simple strategy for optimized design of one-level carry-skip adders

The conventional structure of the CSKA consists of stages containing chain of full adders (FAs) (RCA block) and 2:1 multiplexer (carry skip logic). The RCA blocks are connected to each other through 2:1 multiplexers, which can be placed into one or more level structures[2] . The CSKA configuration (i.e., the number of the FAs per stage) has a great impact on the speed of this type of adder.

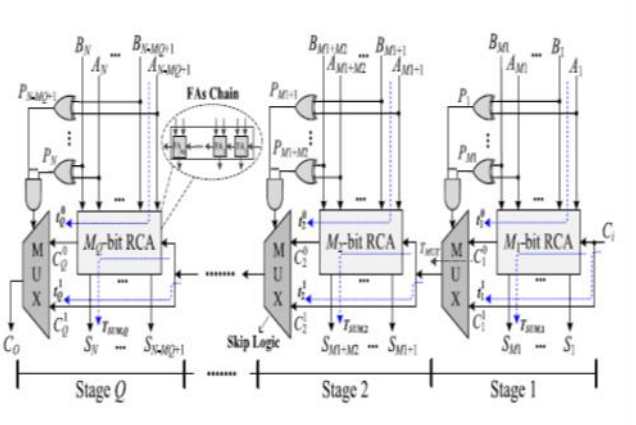

Fig.1 Structure of conventional CSKA[2]

The structure of an N-bit Conv-CSKA, which is based on blocks of the RCA (RCA blocks), is shown in Fig. 1. In addition to the chain of FAs in each stage, there is a carry skip logic. For an RCA that contains N cascaded FAs, the worst propagation delay of the summation of two N-bit numbers, A and B, belongs to the case where all the FAs are in the propagation mode. It means that the worst case delay belongs to the case where

$P_i = A_i \oplus B_i = 1$ for $i = 1, \ldots, N$

where Pi is the propagation signal related to Ai and Bi.The N FAs of the CSKA are grouped in Q stages. Each stage contains an RCA block with Mj FAs (j = 1, . . . , Q) and a skip logic. In each stage,the inputs of the multiplexer (skip logic) are the carry input of the stage and the carry output of its RCA block (FA chain). In addition, the product of the propagation signals (P) of the stage is used as the selector signal of the multiplexer. The CSKA may be implemented using FSS and VSS where the highest speed may be obtained for the VSS structure . Here, the stage size is the same as the RCA block size.

B. High-speed and energy-efficient carry skip adder operating under a wide range of supply voltage levels

The structure is based on combining the concatenation and the incrementation schemes with the Conv-CSKA structure, and hence, is denoted by CI-CSKA. It provides us with the ability to use simpler carry skip logics. The logic replaces 2:1 multiplexers by AOI/OAI compound gates (Fig. 2). The gates, which consist of fewer transistors, have lower delay, area, and smaller power consumption compared with those of the 2:1 multiplexer [8]. Note that, in this structure, as the carry propagates through the skip logics, it becomes complemented. Therefore, at the output of the skip logic of even stages, the complement of the carry is generated.

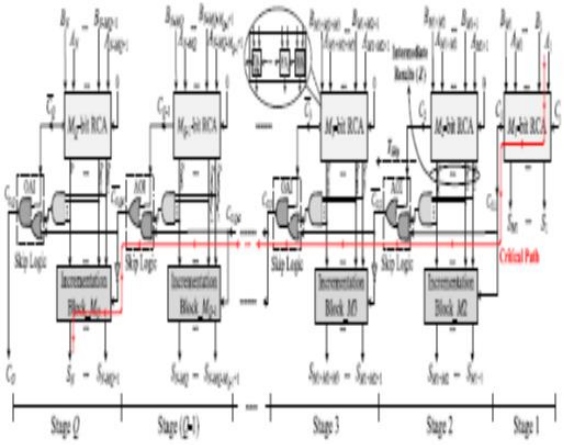

Fig 2 CI-CSKA structure[3]

The adder contains two N bits inputs, A and B, and Q stages. Each stage consists of an RCA block with the size of Mj (j = 1, . . . , Q). In this structure, the carry input of all the RCA blocks, except for the first block which is Ci , is zero (concatenation of the RCA blocks).Therefore, all the blocks execute their jobs simultaneously

III. PROPOSED SYSTEM

In this section, first, the structure of a generic variable latency adder, which may be used with the voltage scaling relying on adaptive clock stretching, is described. Then, a modified hybrid variable latency CSKA structure is proposed.

A. Variable latency adders relying on adaptive clock streching

The basic idea behind variable latency adders is that the critical paths of the adders are activated rarely. Hence, the supply voltage may be scaled down without decreasing the clock frequency. If the critical paths are not activated, one clock period is enough for completing the operation. In the cases, where the critical paths are activated, the structure allows two clock periods for finishing the operation. Hence, in this structure, the slack between the longest off-critical paths and the longest critical paths determines the maximum amount of the supply voltage scaling. Therefore, in the variable latency adders, for determining the critical paths activation, a predictor block, which works based on the inputs pattern, is required.

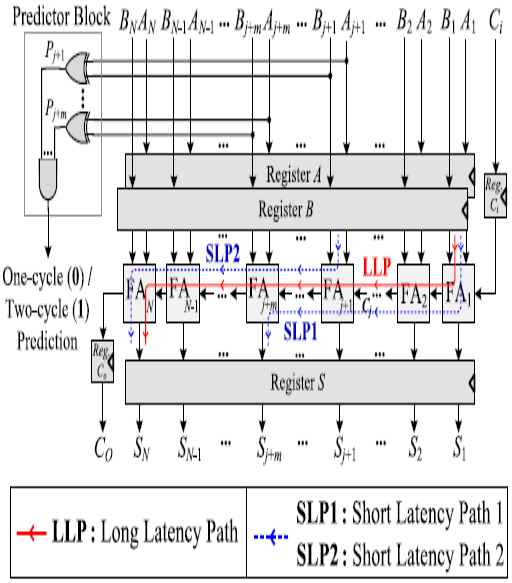

Fig. 3 Generic structure of variable latency adders based on RCA[3].

The concepts of the variable latency adders, adaptive clock stretching, and also supply voltage scaling in an N-bit RCA adder may be explained using Fig. 3. The predictor block consists of some XOR and AND gates that determines the product of the propagate signals of considered bit positions. Since the block has some area and power overheads, only few middle bits are used to predict the activation of the critical paths at price of prediction accuracy decrease. In Fig. 3.1, the input bits (j + 1)th–(j + m)th have been exploited to predict the propagation of the carry output of the j th stage (FA) to the carry output of (j +m)th stage. For this configuration, the carry propagation path from the first stage to the Nth stage is the longest critical path (which is denoted by Long Latency Path (LLP), while the carry propagation path from first stage to the (j+m)th stage and the carry propagation path from (j +1)th stage to the Nth stage (which are denoted by Short Latency Path (SLP1) and SLP2, respectively) are the longest off-critical paths. It should be noted the paths that the predictor shows are (are not) active for a given set of inputs are considered as critical (off-critical) paths. Having the bits in the middle decreases the maximum of the off-critical paths. The range of voltage scaling is determined by the slack time, which is defined by the delay difference between LLP and max(SLP1, SLP2). Since the activation probability of the critical paths is low , the clock stretching has a negligible impact on the throughput (e.g., for a 32-bit adder, m = 6–10 may be considered \. There are cases that the predictor mispredicts the critical path activation. By increasing m, the number of misprediction decreases at the price of increasing the longest off-critical path, and hence, limiting the range of the voltage scaling. Therefore, the predictor block size should be selected based on these tradeoffs.

B. Proposed hybrid variable latency CSKA structure

The basic idea behind using VSS CSKA structures was based on almost balancing the delays of paths such that the delay of the critical path is minimized compared with that of the FSS structure. This deprives us from having the opportunity of using the slack time for the supply voltage scaling. To provide the variable latency feature for the VSS CSKA structure, we replace some of the middle stages in our proposed structure with a PPA modified in this paper. It should be noted that since the Conv-CSKA structure has a lower speed than that of the proposed one, in this section, we do not consider the conventional structure. The proposed hybrid variable latency CSKA structure is shown in Fig. 3.2 where an Mp-bit modified PPA is used for the pth stage (nucleus stage). Since the nucleus stage, which has the largest size (and delay) among the stages, is present in both SLP1 and SLP2, replacing it by the PPA reduces the delay of the longest off-critical paths. Thus, the use of the fast PPA helps increasing the available slack time in the variable latency structure. It should be mentioned that since the input bits of the PPA block are used in the predictor block, this block becomes parts of both SLP1 and SLP2.

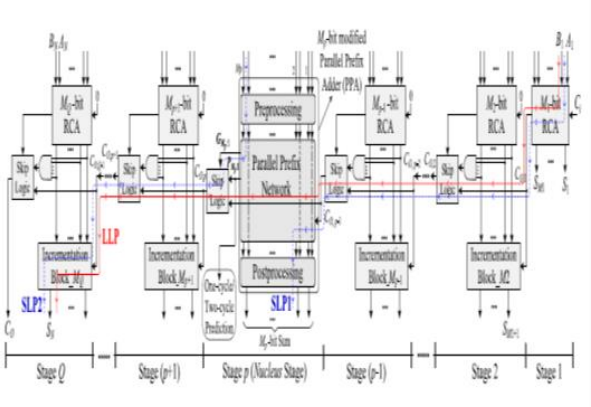

Fig. 4. Structure of the proposed hybrid variable latency CSKA[3]

Here, the parallel prefix network is modified using the Han-Carlson parallel prefix adder.

C. Modification using Han-Carlson Adder

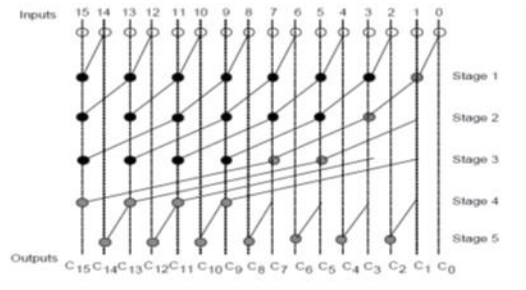

Fig. 5 Prefix Graph of a 16-bit Han-Carlson Adder

Figure 5 shows the prefix graph for a 16-bit Han-Carlson adder. It is a hybrid design combining stages from Brent-Kung and Kogge-Stone. It has five stages, the first stage resembles Brent-Kung adder and the middle three stages resemble Kogge-Stone adder. It possess wires with shorter span than Kogge-Stone. The dot operator was placed in the odd bit positions in the initial stages, but the dot operator was placed in the even bit positions in the final stage. The delay in this structure is given by [(log2 n) +1], while the computation hardware complexity is $\left[\frac{n}{2}(log_2 n)\right]$.The hardware complexity is reduced compared to Kogge-Stone adder.

As shown in the figure,4 in the preprocessing level, the propagate signals (Pi) and generate signals (Gi) for the inputs are calculated. In the next level, using Han-Carlson parallel prefix network, the longest carry (i.e., G8:1) of the prefix network along with P8:1, which is the product of the all propagate signals of the inputs, are calculated sooner than other intermediate signals in this network. The signal P8:1 is used in the skip logic to determine if the carry output of the previous stage (i.e., CO,p−1) should be skipped or not. In addition, this signal is exploited as the predictor signal in the variable latency adder. It should be mentioned that all of these operations are performed in parallel with other stages. In the case, where P8:1 is one, CO,p−1 should skip this stage predicting that some critical paths are activated. On the other hand, when P8:1 is zero, CO,p is equal to the G8:1. In addition, no critical path will be activated in this case. After the parallel prefix network, the intermediate carries, which are functions of CO,p−1 and intermediate signals, are computed (Fig. 3.2). Finally, in the postprocessing level, the output sums of this stage are calculated. It should be noted that this implementation is based on the similar ideas of the concatenation and incrementation concepts used in the CI-CSKA discussed earlier. It should be noted that the end part of the SPL1 path from CO,p−1 to final summation results of the PPA block and the beginning part of the SPL2 paths from inputs of this block to CO,p belong to the PPA block (Fig. 3.2). In addition, similar to the proposed CI-CSKA structure, the first point of SPL1 is the first input bit of the first stage, and the last point of SPL2 is the last bit of the sum output of the incrimination block of the stage Q. The larger size (number of bits), compared with that of the nucleus stage in the original CI-CSKA structure, leads to the decrease in the number of stages as well smaller delays for SLP1 and SLP2. Thus, the slack time increases further.

IV. IMPLEMENTATION RESULTS

The proposed hybrid variable latency adder is implemented in Verilog HDL. The simulations are performed using ModelSim.The results show that the highest power (energy) reduction of ~29% belongs to the RCA structure, which has due the highest slack time. In this case, the supply voltage reduction was 0.2 V. In the case of the standard

C2SLA, since the slack time was small, the voltage reduction was ~0.05 V, which led to a power reduction of 6%. For the hybrid C2SLA, the slack time was higher than that of the standard C2SLA and hence the voltage reduction of ~0.1 V became possible. This provided a higher power reduction (~13%). Finally, the hybrid CSKA had a larger slack compared with that of the hybrid C2SLA, and hence, the voltage reduction of ~0.15 V was possible. This provided the structure with a power reduction of ~23% . The very low delay of the proposed hybrid variable latency CSKA along with its lower power consumption result in the minimum PDP for this structure.

Timing Summary:

- Maximum combinational path delay: 20.145ns
- Total memory usage is 255140 kilobytes

Table 1.1Device utilization summary

Slice Logic Utilization	Used	Available
Number of Slice LUTs	73	9112
Number used as Logic	73	9112
Slice Logic Distribution	Used	Available
Number with an unused Flip Flop	73	73
Number with an unused	0	73
LUT Number of fully used LUT-FF pairs	0	73
IO Utilization	Used	Available
Number of bonded IOBs	97	232

CONCLUSION

In this paper, a hybrid variable latency CSKA was proposed, which exhibits a higher speed an lower energy consumption compared with those of the conventional one. It exploited a modified parallel adder structure at the middle stage for increasing the slack time, which provided us with the opportunity for lowering the energy consumption by reducing the supply voltage. The efficacy of this structure was compared with conventional structures. Again, the suggested structure showed the lowest delay and PDP making itself as a better candidate for high-speed low-energy applications.

ADVANTAGES

- Lowers the power consumption.
- This extension utilizes a modified parallel structure for increasing the slack time.
- Enabling further voltage reduction.
- Lowest delay and PDP.
- Better candidate for high-speed low-energy applications.

APPLICATIONS

- Used in high speed processors.
- Used in various computational devices

V. SOFTWARE DESCRIPTION

A. Xilinx ISE

The Xilinx ISE Simulator(ISim) is a Hardware Description Language(HDL) simulator that enables us to perform functional and timing simulations for VHDL, Verilog and mixed language designs. ISE Design suite14.7version is used in this project. Verilog is the language used for coding .The ISE software controls all aspects of the design flow. The main part of the tool is Project Navigator. Through the project navigator interface, we can access all of the design entry and design implementation tools. We can also access the files and documents associated with our project. This tool is used in the development and design of the system. It manages the creation of the design from the code editor, through to synthesizing, placing and routing and generation and implementation of program files.

B. Verilog

Verilog is a Hardware Description Language, a textual format for describing electronic circuits and systems. It is most commonly used in the design and verification of digital circuits and systems. It is most commonly used in the design and verification of digital circuits at the register-transfer level of abstraction. It is also used in the verification of analog circuits and mixed-signal circuits. Applied to electronic design, Verilog is intended to be used for verification through simulation, for timing analysis, for test analysis(testability analysis and fault grading) and for logic synthesis. The Verilog HDL is an IEEE standard – number1364.

Hardware description languages such as Verilog differ from software programming languages because they include ways of describing the propagation time and signal strengths (sensitivity). There are two types of assignment operators; a blocking assignment (=), and a non-blocking (<=) assignment. The non-blocking assignment allows designers to describe a state-machine update without needing to declare and use temporary storage variables. Since these concepts are part of Verilog's language semantics, designers could quickly write descriptions of large circuits in a relatively compact and concise form. The designers of Verilog wanted a language with syntax similar to the C programming language, which was

already widely used in engineering software development. Like C, Verilog is case-sensitive and has a basic preprocessor (though less sophisticated than that of ANSI C/C++). Its control flow keywords (if/else, for, while, case, etc.) are equivalent, and its operator precedence is compatible with

C. Modelsim

ModelSim is an IDE for hardware design which provides behavioral simulation of a number of languages, i.e., Verilog, VHDL, and SystemC. The Verilog HDL is an industry standard language used to create analog, digital, and mixed-signal circuits. HDL's are languages which are used to describe the functionality of a piece of hardware as opposed to the execution of sequential instructions like that in a regular software application. Both of these tools are extensively in industry, so knowing how to use them can be beneficial later in the career.

REFERENCES

[1] Koren, Computer Arithmetic Algorithms, 2nd ed. Natick, MA, USA A K Peters, Ltd., 2002.
[2] M. Alioto and G. Palumbo, "A simple strategy for optimized design of one-level carry-skip adders," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 50, no. 1, pp. 141–148, Jan. 2003.
[3] High-Speed and Energy-Efficient Carry Skip Adder Operating Under a Wide Range of Supply Voltage Levels Milad Bahadori, Mehdi Kamal, Ali Afzali-Kusha, Senior Member, IEEE, and Massoud Pedram, Fellow, IEEE 2016.
[4] R. Zlatanovici, S. Kao, and B. Nikolic, "Energy–delay optimizationof 64-bit carry-lookahead adders with a 240 ps 90 nm CMOS design example," IEEE J. Solid-State Circuits, vol. 44, no. 2, pp. 569–583,Feb. 2009.
[5] S. K. Mathew, M. A. Anders, B. Bloechel, T. Nguyen, R. K. Krishnamurthy, and S. Borkar, "A 4-GHz 300-mW 64-bit integer execution ALU with dual supply voltages in 90-nm CMOS," IEEE J. Solid-State Circuits, vol. 40, no. 1, pp. 44–51, Jan. 2005
[6] V. G. Oklobdzija, B. R. Zeydel, H. Q. Dao, S. Mathew, and R. Krishnamurthy, "Comparison of high-performance VLSI adders in the energy-delay space," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 6, pp. 754–758, Jun. 2005.
[7] Ramkumar and H. M. Kittur, "Low-power and area-efficient carry select adder," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 2, pp. 371–375, Feb. 2012.
[8] M. Vratonjic, B. R. Zeydel, and V. G. Oklobdzija, "Low- and ultralow-power arithmetic units: Design and comparison," in Proc. IEEE
[9] Int. Conf. Comput. Design, VLSI Comput. Process. (ICCD), Oct. 2005,pp. 249–252.
[10] Nagendra, M. J. Irwin, and R. M. Owens, "Area-time-power tradeoffs in parallel adders," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 43, no. 10, pp. 689–702, Oct. 1996.
[11] Y. He and C.-H. Chang, "A power-delay efficient hybrid carrylookahead/carry-select based redundant binary to two's complement converter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 1, pp. 336–346, Feb. 2008