International Journal of Computer Applications (0975 – 8887)
Volume 65– No.11, March 2013

FPGA-based New Hybrid Adder Design with the Optimal Bit-Width Configuration

Mahmoud A. M. Alshewimy
Computer Engineering Dept.
Istanbul University, Turkey

Ahmet Sertbas
Computer Engineering Dept.
Istanbul University, Turkey

ABSTRACT
This paper presents FPGA-based design of hybrid adder with the optimal bit-width configuration(out of alarge number of possible configurations) of each of the sub-adders constitute the proposed hybrid adder using a high level automated methodology. Algebraicoptimization model for the hybrid adderis built to produce the best choice of types and bit-widths of the sub-adders. In context of this work, several classes of parallel adders are designed and its performance is evaluated to serve as sub-adders inside the hybrid adder. The results show that the proposed model gains a high flexibility in allowing design tradeoffs between the performance criteria delay and areaand successfully to generate the optimalbit-width configurations of the hybrid adder.

Keywords
Hybrid adder, FPGA, optimal, algebraic,delay, area.

1. INTRODUCTION
The addition in digital systems has been a subject of extensive study for many years. It is the slowest among processor operations and very often the addition delay defines the maximum frequency of operation of the chip.Thus, the performance of processors is significantly influenced by the speed of their adders. As a consequence, a fast addition can easily increase the overall chip performance.Much of research has been done in order to design an efficient adder circuits in terms of high speed, small area and low power consumption. However, it has not yet been possible to integrate the various performance criteria delay, area, and power in a single cost function. It is often required to make some compromise, which is always very difficult task, between performance criteria depending on demand or application in digital system design.Ripple carry adders exhibits the most compact design but the slowest in speed. Whereas carry look ahead is the fastest one but consumes more area. Carry select adders act as a compromise between the two adders. In 1992, a new concept of hybrid adders is presented to speed up addition process by Lynch et al., that provided hybrid carry look-ahead/carry select adder design [1]. A different hybrid carry look-ahead/carry select adder design was developed by Wang et al.,[2]. A design methodology for hybrid carry lookahead/carry select adder with reconfigurability is presented in [3]. In2006, Lakshmananet al.,provided a high-speed hybrid parallel-prefix carry-select adder using Ling's algorithm [4]. In these works, the designshave been targeted to a specific technology and some implementation challenges have been encountered whileimplementing these designs in the specified technologies and their authorswere busy with the low level circuit design issues. So, the present paperis concerned with providing a high level automated methodology for designing hybrid adders with high performance without being aware of the low level circuit issues. Algebraic optimization model for FPGA-based new hybrid adder design that combines several types of individual fast parallel adders as sub-adders is proposed. These adders are the linear time ripple carry adder (RCA), the square root time carry skip (CSKA) and carry select (CSLA) adders, and the logarithmic time carry lookahead adder (CLA) and its variations Skalansky and Brent & kung parallel prefix adders. A standard software package "LINGO" is used inwriting and solving the proposed mathematical model.For all designs in this paper, Xilinx ISE 9.1 EDA tool is used for simulation and synthesis purposes.

The remainder of the paper is organized as follows: In Section 2, the design of parallel adders constituting the proposed hybrid adder is presented. In Section 3, the proposed hybrid adder architecture is presented. Section 4 presents the experimental results. Conclusions are drawn in Section 5.

2. PARALLEL ADDERS
In this section, the design of the individual parallel addersrepresenting sub-adders constitute the proposed hybrid adder is presented. The description of most of parallel addersthat have been designed in the present paper can be found in literature or in computer arithmetic books. However, for the sake of completeness a brief description of the different parallel adder types is presented with appropriate references.

2.1 Ripple Carry Adder (RCA)
Ripple carry adders (RCA) provide one of the simplest types of carry-propagate adder (CPA) designswith O(n) area and O(n) delay [5]. An n-bit RCA is formed by concatenating n full adders. The carryout from the i^{th} full adder is used as the carry in of the $(i+1)^{th}$ full adder, as shown in Fig.1.

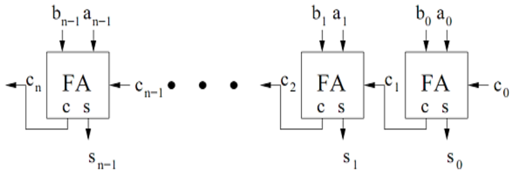

Figure 1:Schematic of RCA

2.2 Carry Lookahead Adder (CLA)

Carry look-ahead adder is designed to overcome the latency introduced by the rippling effect ofthe carry bits. It is the most commonly used scheme for accelerating carry propagation chain. This implementation has a logarithmic ordered delay at the expense of larger area. If A and B are two inputs, "c_i" is carry, "S" and "C" are output sum and carry respectively, then Boolean expression for calculating next carry and addition is:

pi=aixor bi --- Carry Propagation (1)

gi= ai and bi --- Carry Generate (2)

Ci+1= gi or (pi and ci) --- Next Carry (3)

Si+1= pixor ci (4)

In general, an n-bit CLA with a maximum fan-in of r requires $\log_r$ (n) CLA logic levels [6].

2.3 Modified Carry Skip Adder(CSKA)

In the conventional CSKA, the operands are divided into blocks of r-bit blocks.Each blockrepresents RCA utilized to produce the sum bits and a carry out bit for the block. CSKA has O (√n) delay provides a good compromise in terms of delay, along with a simple and regular layout [6].To gain more improvement in the speed of CSKA, we modified the CSKA architecture via using CLA blocks rather than RCA.Schematic of 16-bit MCSKA adder is shown in Fig.2.

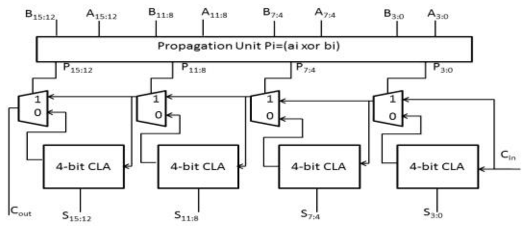

Figure2:Schematic of 16-bit modified CSKA

2.4 Carry Select Adder(CSLA)

A carry-select adder is divided into blocks, each of which – except for the least-significant – performs two additions in parallel, one assuming a carry-in of zero, the other a carry-in of one. A four bit carry select adder generally consists of two ripple carry adders and a multiplexer. Thecarry-select adder is simple but rather fast, having a gate level depth of(√n) [5].Schematic of 16-bit CSLA adder is shown in Fig.3.

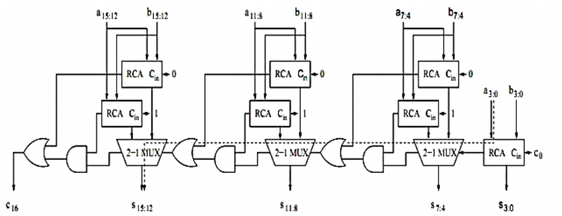

Figure3:Schematic of 16-bit CSLA

2.5 SKLANSKY Parallel Prefix Adder

SKLANSKY's prefix algorithm (PPA-SK), first used forconditional-sum addition [7], is one of the most common prefix algorithms. This algorithm has minimaldepth but the fan-out increases exponentially towardsthe final stages. The maximum fan-out is O(n) that is linear to thenumber of operand bits. Each black cell represents an arbitrary associative operator o is defined in (5):

$$(g_{out}, p_{out}) = (g_i, p_i) o (g_j, p_j) = (g_j + p_i g_j, p_i p_j) \quad (5)$$

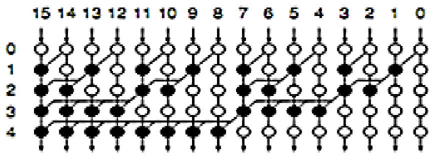

Figure4:Schematic of 16-bit PPA-SK

2.6 BRENT &KUNG Parallel Prefix Adder

BRENT and KUNG's prefix algorithm(PPA-BK) has low fan-out (i.e. O(log n) instead of O(n)) but twice the depth of the SKLANSKY algorithm [8]. It is quite area efficient due to the small number of black cells (remember that the white cells contain nologic) and due to the low wiring requirements. Schematicof 16-bit PPA-BK is shown in Fig. 5.

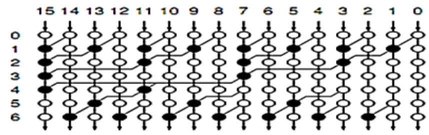

Figure5:Schematic of 16-bit PPA-BK

2.7 Implementation Results of Parallel Adders

All adders are designed using VHDL (Very High Speed Integration Hardware Description Language). To get delay and area report, we use XILINX ISE 9.1 i as synthesis tool and its built-in ISE simulator for simulation. FPGA-Virtex5 is used for implementation. The simulation result of parallel addersdesigns are shown in Fig. (6) to Fig. (11).

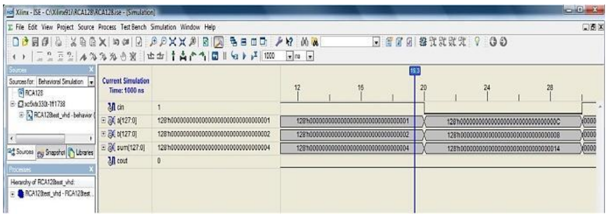

Figure6:Simulation chart of128-bit RCA

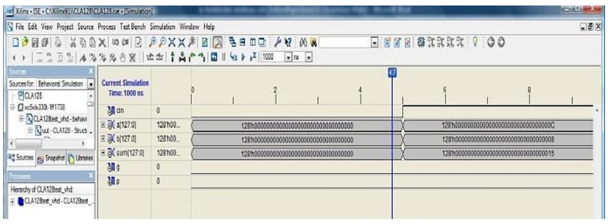

Figure7:Simulation chart of128-bit CLA

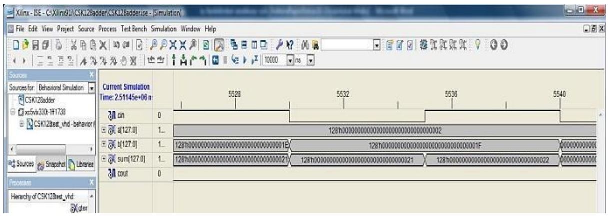

Figure8:Simulation chart of 128-bit CSKA

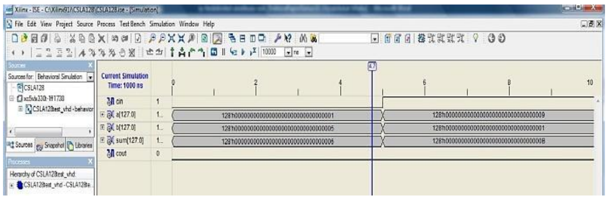

Figure9:Simulation chart of128-bit CSLA

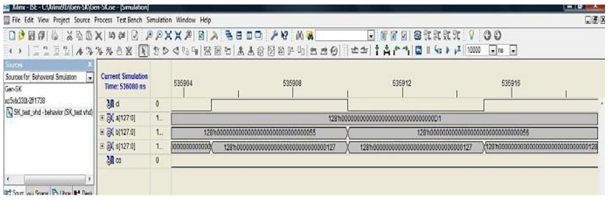

Figure10:Simulation chart of 128-bit PPA-SK

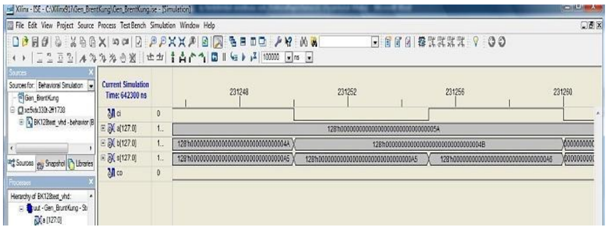

Figure11:Simulation chart of 128-bit PPA-BK

Summary of delay and area reports for different bit widths of the parallel adders is shown in Table 1.Rows"DS" and "DC"represent the worst case delays of sum and carry outs respectively. Rows "Slice" give the area of the designs.

Table 1. Summary of delay (ns) and area (slice) reports of paralleladders

		4 bit	8 bit	16 bit	32 bit	64 bit	128 bit
RCA	DS	7.4	9.6	13.8	19.7	34.4	83.1
	DC	6.8	9.5	12.8	19.6	34.1	82.9
	Slice	3	9	23	35	75	124
CLA	DS	6.6	8.78	9.96	10.9	15.6	20.8
	DC	6.3	6.69	7.32	8.68	11.5	19.2
	Slice	4	10	27	37	95	186
CSKA	DS	7.0	9.02	11.0	17.4	23.8	41.1
	DC	6.2	8.17	10.6	16.1	22.9	40.9
	Slice	5	11	22	37	69	137
CSLA	DS	6.5	9.3	11.3	16.9	25.1	53.7
	DC	5.6	8.6	10.9	16.3	24.8	53.4
	Slice	6	13	23	56	106	242
PPA-SK	DS	7.1	9.6	11.2	14.5	24.6	35.4
	DC	7.0	9.4	11.0	13.9	24.4	34.5
	Slice	5	8	21	48	103	242
PPA-BK	DS	7.0	9.1	12.9	16.6	25.2	49.4
	DC	7.0	8.92	12.87	16.62	23.51	46.31
	Slice	5	10	19	36	72	163

3. PROPOSED HYBRID ADDER ARCHITECTURE

A general architecture for N-bit hybrid adder is proposedin Fig.12. The hybrid adder consists of L sub-adders with variable bit-widths (n_i) where $0 \leq n_i \leq N$ and $1 \leq i \leq L$.The carry-out (Cout) of each sub-adder is connected to the carry-in (Cin) of the next sub-adder, making the connection between these sub-adders linear. The question now how tochoose thebest types of sub-adders and their orders in automated process?; the answer to this question can be found in Fig.13.

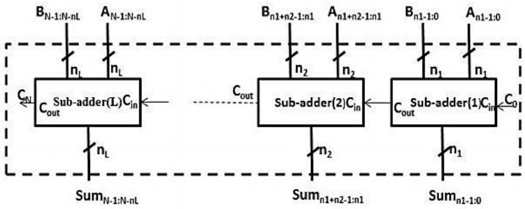

Figure 12: A proposed general architecture of N-bit hybrid adder

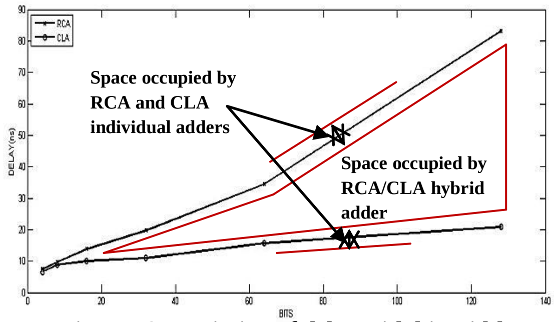

Figure 13: Variation of delay with bit-width

Fig.13 shows the space occupied by the implementations of RCA/CLA hybrid adder and itsindividual ones. The space occupied by this hybrid adder is due to its architecture that combines two different types of adders; RCA and CLA. The space occupied by the individual adders is owing to the design variations such as optimization efforts. It's apparent that the design space occupied by the hybrid adder is large than of individual ones gives more flexibility to make compromise between performance criteria delay, area of a design. To find an optimal design point in this space, we propose to develop algebraic optimization model using LINGO 13.0 [9]. It is a standard tool uses linear programming (LP) to describe the problem of concern. The adjective linear means that all the mathematical functions in this model are required to be linear. The word programming does not refer here to computer programming; rather, it is essentially a synonym for planning. Thus linear programming involves the planning of activities to obtain an optimal result, i.e., a result that reaches the specified goal best among all feasible alternatives.

3.1 Algebraic OptimizationModel of the Proposed Hybrid Adder

If the problem of design a hybrid adder with a minimum overall delay under area constraint is considered. It is required to write an algebraic mathematical model for a hybrid adder overall delay. The problem could be defined as the following:

Minimize [the overall delay (OD) of the hybrid adder]

Subject to the restrictions

1- The total area of the hybrid adder (TA) ≤ Maximum allowed area (MAA)

2- $\sum_{i=1}^{L} n_i = N$

The objective here is to find the values of sub-adders bit widths (n_i) so as to minimize the overall delay of the hybrid adder, subject to the restrictions imposed on their values. It is obvious from the first constraint; it is required to model the total area of the hybrid adder. Assume that the total area of the hybrid adder is TA and the area of the i^{th} sub-adder is A_i where i refer to the adder type. Using the proposed architecture in Fig. 12, the total area equals to the sum of all areas of the individual sub-adders (A_i) that can be formulated as:

$$\mathrm{TA} = \sum_{i=1}^{L} A_i \qquad (6)$$

But to put (6) in a form that could be optimized allowing to choose between different types of the sub-adders. This can be achieved using a decision variable (v_{n_i}) that could be inserted into (6) where v_{n_i} could be 1 or 0. If v_{n_i}=0, where n_i =16, it means that the i^{th} type 16-bit sub-adder is not allowed to be a part of the hybrid adder. While v_{n_i}=1, where n_i =8, it means that the i^{th}type 8-bit sub-adder has been selected to be a part of the hybrid adder. So, the area of the i^{th} type sub-adder that is the sum of the individual areas (A_i) is described as

$$A_i = \sum_{n_i=0}^{N} A_{n_i} .\ v_{n_i} \qquad (7)$$

Using (6) and (7), the total area of a hybrid adder can be described as

$$\mathrm{TA} = \sum_{i=1}^{L} \sum_{n_i=0}^{N} A_{n_i} .\ v_{n_i} \qquad (8)$$

For a hybrid adder with the proposed architecture shown in Fig.12, there are different delay paths from the input to the carry-out while the carry propagates from a sub-adder to the subsequent ones. The worst case delay of a carry propagation adder is described by the delay from the input to the sum (DS)or to carry-out (DC) according to the order of a sub-adder. The overall delay of a hybrid adder could be evaluated by finding the maximum delay between these delay paths.

So the overall delay (OD) = Max (DS_1, DC_1+DS_2 , ……, DC_1+DC_2+……+DS_L).

The worst case delay could be determined using (9)

$$\sum_{i=1}^{L} \sum_{n_i=0}^{N} D_{ni} . v_{ni} \qquad (9)$$

D_{ni} represents adelay of the i^{th} type sub-adder that could be sum or carry delay according to the order of a sub-adder. These delays values could be acquired from our pre-designedparallel adders. Using (8) and (9) the problem of designing N-bit hybrid adder with the minimum overall delay subject to area constraint can be described as shown in Fig. 14.

4. EXPERIMENTAL RESULTS

The design of 128-bit hybrid adder composed of two sub-adders (L=2) is considered here. We are concernedwith the problem of designing a hybrid adder with delay optimization under area constraint. Possible bit-widths of sub-adders are in the range from 4 to 128 bits. The algebraic mathematical model of the proposed hybrid adder for L=2 is written and solved using "LINGO" software package.

Objective: Minimize (OD)

Subjectto:

1-$\sum_{i=1}^{L} \sum_{n_i=0}^{N} A_{n_i} . v_{n_i} \ \leq \mathrm{MAA}$

2-$\sum_{n_i=0}^{N} DS_{ni}. v_{n_i} \ \leq \mathrm{OD}$, where i = 1

3- $\sum_{i=1}^{L-1} \sum_{n_i=0}^{N} DC_{ni} . v_{ni} + \sum_{n_i=0}^{N} DS_{ni} . v_{n_i}|_{i=L} \leq \mathrm{OD}$

4- $\sum_{i=1}^{L} \sum_{n_i=0}^{N} n_i . v_{ni} = N$

5-$\sum_{n_i=0}^{N} v_{n_i} \ \leq 1$, for every ith type sub − adder where $1 \leq\ i\ \leq L$

Figure 14: Algebraic optimization model of N-bit hybrid adder

All DS, DC, and A_i values are stored in EXCEL data sheets that could be read by "LINGO". All possible combinations (2^6) of ordering of sub-adders constitute a hybrid adder are considered. The optimization model is written in a parameterized form to facilitate changing the bit-width (N) of a hybrid adder and the constraint area (MAA) given to the design. The characteristics of a hybrid adder obtained using "LINGO" are shown in Table 2. A symbol "|" is used to clarify that the sub-adders are connected in a linear way. For example, the form RCA(4)|CLA(124) illustrates that CLA is located to the 4-low order bits and that RCA is located to the 124-high order bits of the RCA|CLA hybrid adder.

Table 2.Summary of optimization results

Hybrid adder Configurations	MAA(slice)	OD(ns)
RCA(32)|CLA(96)	175	21.9
RCA(65)|CSLA(63)	180	59.3
CSKA(93)|CLA(35)	140	35.1
RCA(35)|SK(93)	205	50.2
RCA(4)|CSKA(124)	133	54.3
RCA(4)|BK(124)	143	53.2

Table 3 presents the performance criteria delay, area, and AT of the various adders. AT Product is normalized and shown in Fig. 15. It is found that RCA|CLA, CLA, and CSKA|CLA adders give a higher performance compared to other adder implementations.The results proved also that the proposed algebraic optimization model can be used easily to allow design tradeoffs between a hybrid adder performance criteria delay and area for efficient performance.

Table 3.Characteristics of various adders implementation

Adder Type	Area (slice)	Delay (ns)	AT	AT (normalized)
RCA	124	83.1	10304.4	2.69
CLA	186	20.80	3868.8	1.01
CSKA	137	41.1	5630.7	1.47
CSLA	242	53.7	12995.4	3.39
PPA-SK	242	35.4	8566.8	2.24
PPA-BK	163	49.4	8052.2	2.10
RCA(32)|CLA(96)	175	21.9	3832.5	1.00
CSKA(93)|CLA(35)	140	35.1	4914.0	1.28

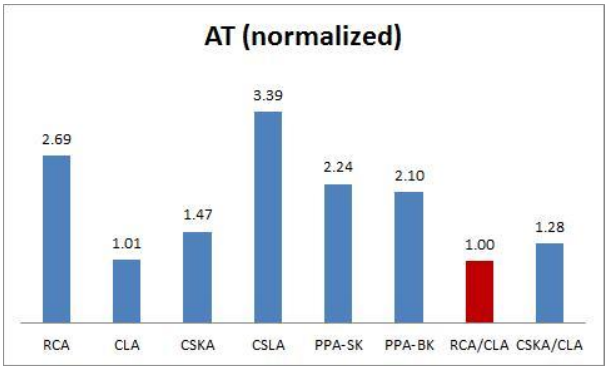

Figure 15: Performance comparison between various adders implementation

5. CONCLUSIONS and FUTURE WORK

This paper has been concerned with providing a high level automated methodology for designing hybrid adders with high performance without being aware of the low level circuit issues. So, a high level automated methodology for designing a hybrid adder has been introduced. Algebraic optimization model for FPGA-based new hybrid adder design that combines several types of individual fast parallel adders as sub-adders has been proposed. The proposed design method gains a great flexibility in allowing tradeoffs between the performance criteria delay and area of a design and has a performance advantage over other parallel adders. In a similar way, the problem of delay optimization under power constraint can be developed.

6. REFERENCES

[1] Lynch, T. and Swartzlander, E. E.,"A spanning Tree Carry Lookahead Adder", IEEE Trans. on Computers, 41 (August 1992), 931-939,1993.

[2] Wang, Y., Pai, C., and Song, X., "The Design of Hybrid Carry-Lookahead/Carry–Select Adders", IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 49, no. 1, 2002.

[3] Li, J., Yu, J. and Huang, Y., "A Design Methodology for Hybrid Carry-Lookahead/Carry-Select Adders with Reconfigurability", in Proc. 15th VLSI/CAD Symp., (Pintung), 2005.

[4] Lakshmanan, Meaamar,A., and Othman,M.,"High-Speed Hybrid Parallel-Prefix Carry-SelectAdder Using Ling's Algorithm", ICSE2006 Proc., Kuala Lumpur, Malaysia, 2006.

[5] Zimmermann, R.1997. Binary Adder Architectures for Cell – Based VLSI and their Synthesis, Thesis (Phd), Swiss Federal Institute of Technology, Zurich.

[6] PARHAMI, B. 2000. Computer Arithmetic: Algorithms and Hardware Designs, Oxford University Press, USA.

[7] Sklansky,J., "Conditional sum addition logic", IRE Trans. Electron. Comput., vol. EC-9, no. 6, pp. 226–231, June 1960.

[8] Brent,R. P. and Kung,H. T., "A regular layout for parallel adders", IEEE Trans. Comput., vol. 31, no. 3, pp. 260–264, Mar. 1982.

[9] Vanderbei, R. J. 2001. Linear Programming: Foundations and Extensions. 2nd Ed., USA.