I. J. Engineering and Manufacturing, **2020, 4, 1-12**
Published Online August 2020 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijem.2020.04.01

Power-Time Efficient Hybrid Adder Design Based on LP with Optimal Bit-Width Generation

Mahmoud A. M. Alshewimy
Computer and Control Engineering Department, Tanta University, Egypt
E-mail: mahmoud_elshwaimi@f-eng.tanta.edu.eg

Received: 08 January 2020; Accepted: 28 January 2020; Published: 08 August 2020

Abstract: This paper presents a systematic method for a hybrid adder design through allocating the optimal bit-widths and types of classical adders constituting a hybrid adder. The proposed optimization scheme considers two aspects design delay and power. It is based on a mathematical modeling of the proposed hybrid adder architecture following the principle of LP (Linear Programming). Two models, delay optimization under power constraint and power optimization under delay constraint, are introduced. Various experiments are presented to demonstrate the effectiveness and applicability of the proposed design scheme. The results indicate that the proposed scheme successfully allocates simultaneously and in a systematic way the optimal bit-widths of the sub-adders constituting a hybrid adder; providing an improvement in (power x delay) performance reaching 71.6%. The results obtained also indicate that the proposed design scheme introduces a high flexibility in making a compromise between delay and power of the adder design.

Index Terms: Delay, hybrid adder, linear programming, optimal, power.

1. Introduction

The addition in digital systems has been a subject of extensive study for many years. Arithmetic operations are the slowest among processor operations and very often the addition delay defines the maximum frequency of operation of the chip. In addition to explicit arithmetic (such as addition, subtraction, multiplication, and division) performed in a program, additions are performed to increment program counters and calculate effective addresses [1]. Statistics presented in [2] show that, in a prototypical RISC machine (DLX), 72 percent of the instructions perform additions (or subtractions) in the data path. The statistics reported in ARM processors even reaches 80 percent [3]. 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. Various techniques have been proposed in order to improve the speed of large adders. In every case the target is to compute the input carry more quickly for the high order bits.

The simplest adder architecture is the ripple carry adder (RCA). For this adder the worst-case delay increases linearly with the number of bits [4]. Whereas carry look-ahead adders based on parallel prefix computation methods yield the fastest adders but consumes more area [4]. In 1992, the first hybrid adder has been introduced by Lynch et al. that have provided hybrid carry look-ahead/carry-select adder design to speed up addition process [5]. The adder employs the carry-select method for calculating the sums; however the carries are generated by a carry look-ahead tree. A different hybrid carry look-ahead/carry-select adder design has been introduced by Wang et al., [6]. A design methodology for hybrid carry look-ahead/carry-select adder with re-configurability is presented in [7]. In 2006, Lakshmanan et al., had provided a high-speed hybrid parallel- prefix carry-select adder using Ling's algorithm [8]. In 2019, design of delay efficient hybrid adder for high speed applications is introduced [9]. The design employed utilizes the advantage of parallel prefix architecture (PPA), Hardware description language (HDL), and the concept of "Hybrid Variable Latency" technique”. In 2019 also, a novel hybrid adder was designed using the Gate diffusion Input (GDI) technique to obtain a new low power high speed adder [10].

For decades, much of research has been developed in order to design an efficient adder circuits in terms of high speed and small area. However, nowadays low power design became significant due to the spread of wireless communication and portable computing systems. Lower power dissipation is translated into longer-battery life for portable systems, smaller and less expensive package and higher integration density for any VLSI system. Therefore, this paper is interested in designing of a high performance adder in terms of low delay-power efficiency and giving the capability to make a compromise according to the application demand.

In the present work, the implementation of a hybrid adder is carried out by combining different types of classical adders with different bit-widths. In the early sections, we have evaluated the performance of different types of classical adders with various bit-widths that could contribute in building a hybrid adder. Two algebraic optimization models for a new hybrid adder design that combines several types of classical adders as sub-adders are described. These adders are ripple carry adder (RCA), carry skip (CSKA) and carry select (CSLA) adders, and carry look-ahead adder (CLA) and its variations Skalansky (SK) and Brent Kung (BK) parallel prefix adders.

A set of various experimental results has been provided to show how much the proposed hybrid adder scheme is effective in obtaining the best performance compared with those of the classical adders. For coding and solving the proposed hybrid adder models, the software package "LINGO 13.0" from LINDO Systems Inc. is utilized. The Xilinx ISE 9.1 tool is utilized in VHDL coding, simulating and synthesizing all adder designs.

The rest of our paper is organized as follows. In Section 2, classical parallel adders design constituting the proposed hybrid adder is described. The proposed hybrid adder architecture is developed in Section 3. Section 4 presents the results obtained using the optimization models of the proposed hybrid adder architecture. Conclusions are drawn in Section 5.

2. Classical Parallel Adders

In this section, a brief description of classical parallel adders used as sub-adders constituting a hybrid adder is introduced. This description can be found in literature. However, for the sake of completeness we briefly explain the concept of the different classical parallel adders associated with appropriate references.

A. Ripple Carry Adder (RCA)

A ripple carry adder is the most compact adder topology consisting of several full adders connected in series. The implementation of ripple carry adder has O (n) area and O (n) delay [11], which allows for fast design time. However, the ripple carry adder is relatively slow, since each full adder must wait for the carry bit to be calculated from the previous full adder as shown in Fig. 1.

If A and B are two inputs, "C_i" is carry, "S" and "C" are output sum and carry respectively.

The sum and carry are computed as follows:

$$S_i=A_i \text{ xor } B_i \text{ xor } C_i \quad (1)$$

$$C_{i+1}=(A_i \text{ and } B_i) \text{ or } (C_i \text{ and } (A_i \text{ or } B_i)) \quad (2)$$

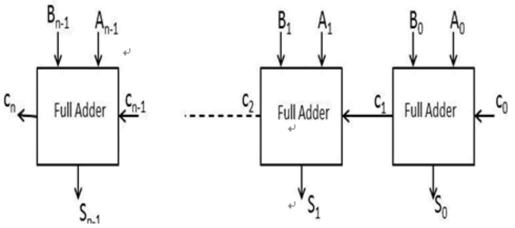


Fig 1. Architecture of RCA adder

B. Carry Look-Ahead Adder (CLA)

Weinberger and Smith proposed this scheme in 1958[12]. Carry look-ahead adder is designed to overcome the latency introduced by the rippling effect of the carry bits. Its implementation has a logarithmic ordered delay at the expense of larger area. 16-bit CLA adder consisting of four 4-bit CLA is depicted in Fig. 2. If A and B are two inputs, "Ci" is carry, "S" and "C" are output sum and carry respectively, then Boolean expressions for calculating sum and carry outputs are as follows:

$$P_i = A_i \text{ xor } B_i \quad \text{--- Carry Propagation} \quad (3)$$

$$G_i = A_i \text{ and } B_i \quad \text{--- Carry Generate} \quad (4)$$

$$C_{i+1} = G_i \text{ or } (P_i \text{ and } C_i) \quad \text{--- Next Carry} \quad (5)$$

$$S_{i+1}= P_i \text{ xor } C_i \quad \text{--- Sum} \quad (6)$$

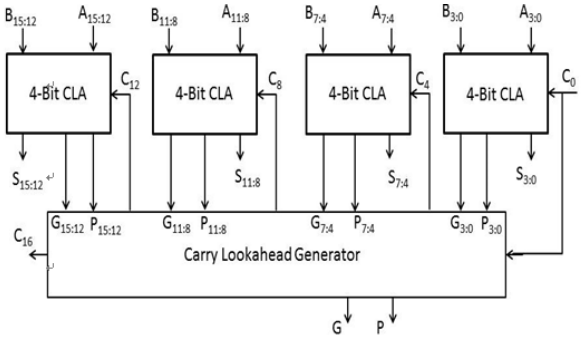


Fig 2. Architecture of 16-bit CLA adder

C. Carry Skip Adder (CSKA)

The conventional carry skip scheme is proposed by Kilburnet to accelerate the carry propagation [13]. The carry skip adder divides the words to be added into blocks and ripple carry adder is used to produce the sum bit and carry. CSKA has O (√n) delay provides a good compromise in terms of delay, along with a simple and regular layout [11]. The conventional CSKA architecture is modified to gain more improvement in the performance via using 4-bit CLA rather than RCA. A new 16-bit CSKA adder architecture based on four 4-bit CLA is depicted in Fig. 3.

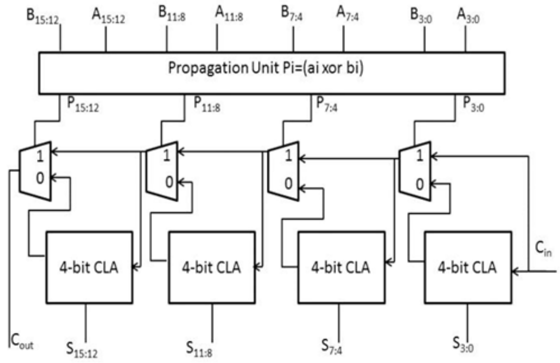


Fig 3. Architecture of 16-bit CSKA Adder

D. Carry Select Adder (CSLA)

Each block of a carry-select adder generates two sets of sums, one for the carry-in 0 and the other for 1. The carry-select logic selects the appropriate set of sums upon arrival of the carry bit. A four bit carry select adder generally consists of two ripple carry adders and a multiplexer. The carry-select adder is simple but rather fast, having a gate level depth of (√n) [11]. The architecture of 16-bit CSLA adder is depicted in Fig. 4.

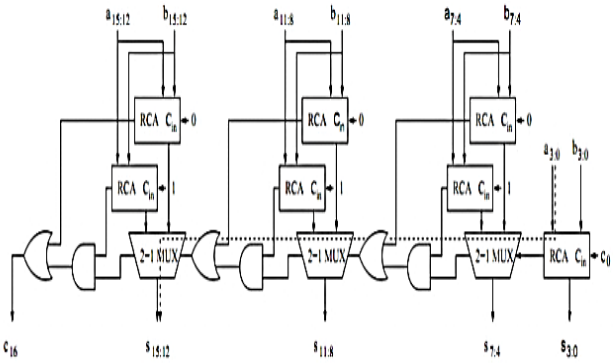


Fig 4. Architecture of 16-bit CSLA adder

E. Sklansky Parallel Prefix Adder (SK)

Sklansky prefix algorithm (PPA-SK), first used for conditional-sum addition [14], is one of the most common prefix algorithms. This algorithm has minimal depth but the fan-out from the inputs to outputs along the critical path increases significantly which introduces latency in the circuit [11]. The maximum fan-out is O(n) that is linear to the number of operand bits. Each black cell represents an arbitrary associative operator o is defined as:

$$(g_{out}, p_{out}) = (g_i, p_i) o (g_j, p_j) = (g_i p_i g_j, p_i p_j) \tag{7}$$

where the pairs (g_i, g_j) and (p_i, p_j) indicate generation bits and propagation bits, respectively. Taking into consideration, the pair (g_i, p_i) has a higher order than (g_j, p_j). The graph of 16-bit Sklansky adder is depicted in Fig. 5.

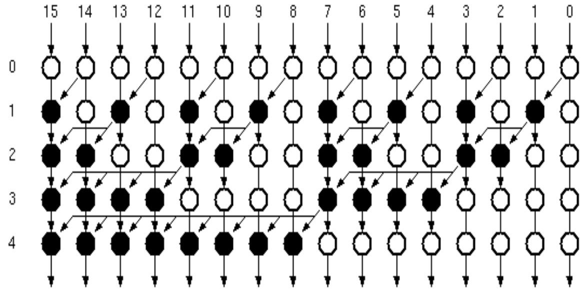


Fig 5. 16-bit SK adder graph

F. Brent Kung Parallel Prefix Adder (BK)

Brent Kung adder is oriented towards simpler tree structure with a fewer computation nodes. It has low fan-out (i.e. O (log n) instead of O (n)) but twice the depth of Sklansky adder [15]. It is quite area efficient due to the small number of black cells (remember that the white cells contain no logic) and due to the low wiring requirements. The graph of 16-bit Brent Kung adder is depicted in Fig. 6.

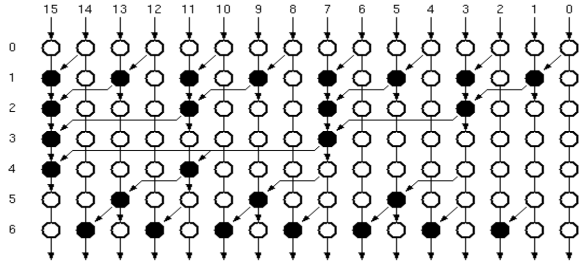


Fig 6. 16-bit BK adder graph

3. The Proposed Hybrid Adder Architecture

To design an efficient hybrid adder, it is essential to have a performance higher than that of classical adders. If a hybrid adder is represented using a combinations of classical adders. In that case, individual classical adders could be seen as a part of hybrid adders and hybrid adder could also be seen as an individual adder where containing only a single adder with 0-bit widths of other adders. Figures 7 and 8 show the delay and power values of the implementations of RCA, CSKA, and CLA classical adders obtained using Xilinx ISE9.1 software package.

The regions (C, D, E) shown on the figures specify the design space range covered by the implementations of classical adders. This design space range is owing to the design variations such as basic building components used and optimization efforts. However, the regions (A, B) are the design space range of hybrid adders consisting of these classical adders. The design space range occupied with the hybrid adders is due to its architecture that combines different types of classical adders. It's apparent that the design space covered by hybrid adders is larger than that of classical adders gives more flexibility in making compromise between performance criteria delay and power of a design. Besides that, the spaces covered by classical adders represent a small part of the whole design space occupied with hybrid ones.

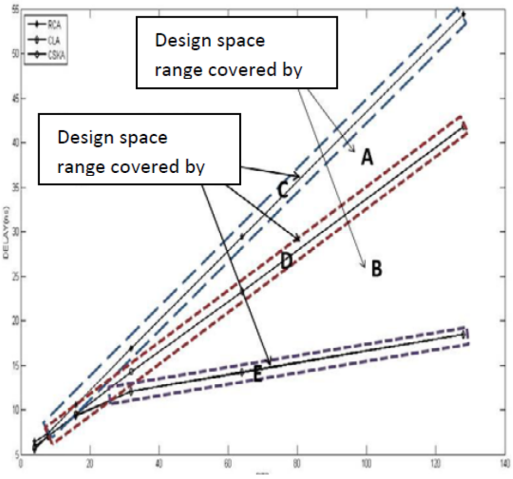


Fig 7. Delay space range covered by hybrid and classical adders

In Fig. 9, a general architecture for N-bit hybrid adder is proposed. 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 (C_{out}) of each sub-adder is connected to the carry-in (C_{in}) of the next sub-adder, making the connection between these sub-adders linear. The question now how to choose the best types of sub-adders and their bit-width topology in a systematic and optimal way to develop a hybrid adder with effective performance; to find an optimal design point in the design space range (A,B) covered by hybrids adder, we propose to develop mathematical optimization model for the general architecture of a hybrid adder depicted in Figure 9.

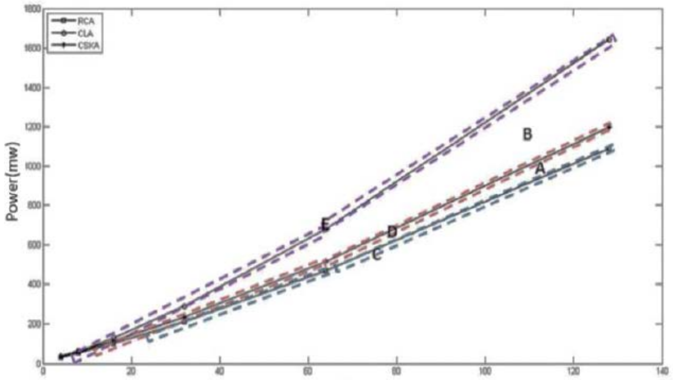


Fig 8. Power consumption space covered by hybrid and classical adders

This could be carried out using linear programming (LP) [16]. 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.

Optimization models for the proposed hybrid adders in terms of delay and power are created using LP and introduced in the following subsections:

A. Delay optimization model under power constraint
B. Power optimization model under delay constraint

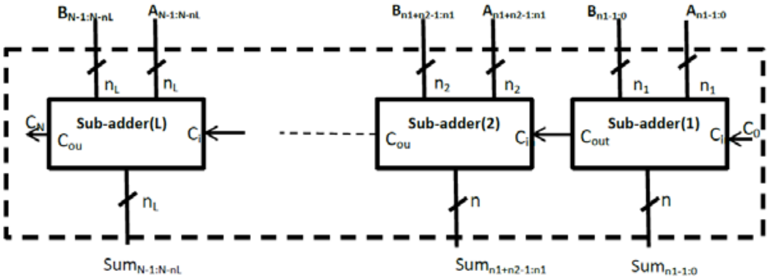


Fig 9. General architecture of the proposed hybrid adder

A. Delay Optimization Model Under Power Constraint

If the problem of design a hybrid adder with a minimum overall delay under power constraint is considered, taking into consideration the proposed general architecture of the hybrid adder depicted in Fig. 9 with a given order of L sub-adders. It is required to write an algebraic optimization model for a hybrid adder overall delay. The problem could be defined as the following:

> *Minimize [overall delay (OD) of a hybrid adder]*
> *Subject to the restrictions*
> *1- Total power of a hybrid adder (TP) ≤ Maximum allowed power (MAP)*
> *2-* $\sum_{i=1}^{L} n_i = N$

Here the objective is to allocate the optimal values of sub-adders bit widths (ni) so as to minimize the overall delay of a hybrid adder, subject to the constraints imposed on their values. It is obvious from the first constraint; it is required to model the total power of the hybrid adder. In doing so, assume that the total power of the hybrid adder is TP and the power of the ith sub-adder is Pi where i refer to the adder type. Using the proposed architecture in Fig. 9, the total power consumption equals to the sum of all power consumption of the individual sub-adders (Pi) that can be formulated as:

$$\mathrm{TP} = \sum_{i=1}^{L} \mathrm{P_i} \tag{8}$$

For allowing choosing between the diverse bit-widths of the sub-adders constituting the hybrid adder equation (8) should be placed in an optimized form. In doing so, a decision variable ($\mathrm{v_{n_i}}$) is inserted into (8) where $\mathrm{v_{n_i}}$ could be 1 or 0 representing a binary variable. For example, if $\mathrm{v_{n_i}}$= 0, where $\mathrm{n_i}$ =16, it means that the $\mathrm{i^{th}}$ type 16-bit sub-adder is not allowed to contribute into the structure of the hybrid adder. While $\mathrm{v_{n_i}}$=1, where $\mathrm{n_i}$ =8, it means that the $\mathrm{i^{th}}$ type 8-bit sub-adder has been selected to be a part of the hybrid adder. So, the power of the $\mathrm{i^{th}}$ type sub-adder that is the sum of the individual powers (Pi) is described as

$$\mathrm{P_i} = \sum_{\mathrm{n_i}=0}^{\mathrm{N}} \mathrm{P_{n_i}} \times \mathrm{v_{n_i}} \tag{9}$$

Using (8) and (9), the total power of a hybrid adder can be described as

$$\mathrm{TP} = \sum_{i=1}^{L} \sum_{\mathrm{n_i}=0}^{\mathrm{N}} \mathrm{P_{n_i}} \times \mathrm{v_{n_i}} \tag{10}$$

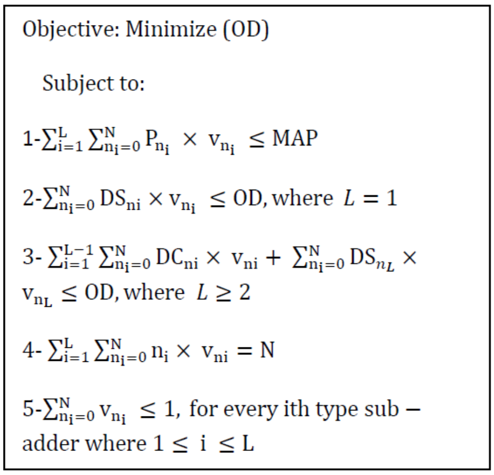
For a hybrid adder with the proposed architecture shown in Fig. 9, there are different delay paths from the input to the carry-out while the carry propagates from a sub-adder to the subsequent ones. The worst case delay of a carry propagation adder is described by the delay from the input to the sum (DS) or to carry-out (DC). The overall delay of a hybrid adder could be evaluated by finding the maximum delay between these delay paths.

So the worst case delay = Max $(\mathrm{DS_{n_1}}, \mathrm{DC_{n_1}} + \mathrm{DS_{n_2}}, \dots\dots\dots\dots, \mathrm{DC_{n_1}} + \mathrm{DC_{n_2}} + \cdots .. + \mathrm{DS_{n_L}})$, where the delay of sum is larger than the delay of carry for each type of sub-adder.

These delay paths could be determined using

$$\begin{cases} Max(\ \sum_{n_1=0}^{N} DS_{n_1} \times v_{n_1} for\ \ L = 1) \\ Max(\sum_{\mathrm{i=1}}^{\mathrm{L-1}} \sum_{\mathrm{n_i=0}}^{\mathrm{N}} \mathrm{DC_{ni}} \times \mathrm{v_{ni}} + \sum_{n_L=0}^{N} DS_{n_L} \times v_{n_L}\ \ \mathrm{for\ L} \ \geq 2) \end{cases} \tag{11}$$

The values of the sum delays (DS_{n_i}), the carry delays (DC_{n_i}) and power values (P_{n_i}) have been acquired for each classical adder using Xilinx ISE 9.1 after place and routing; where ni can be varied from 4 to 128 bits. Using equations of (10) and (11), the problem of modeling a hybrid adder with the minimum delay (OD) subject to power constraint is described in Fig. 10.



Objective: Minimize (OD)

Subject to:

1-$\sum_{i=1}^{L}\sum_{n_i=0}^{N} P_{n_i} \times v_{n_i} \leq \mathrm{MAP}$

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

3- $\sum_{i=1}^{L-1}\sum_{n_i=0}^{N} DC_{ni} \times v_{ni} + \sum_{n_i=0}^{N} DS_{n_L} \times v_{n_L} \leq \mathrm{OD}$, where $L \geq 2$

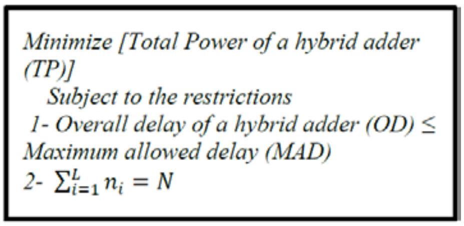
4- $\sum_{i=1}^{L}\sum_{n_i=0}^{N} n_i \times v_{ni} = N$

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

Fig 10. Delay optimization model of a hybrid adder

B. Power Optimization Model Under Delay Constraint

By the same way, if the problem of design a hybrid adder with a minimum power under delay constraint is considered. The problem could be defined as follows:



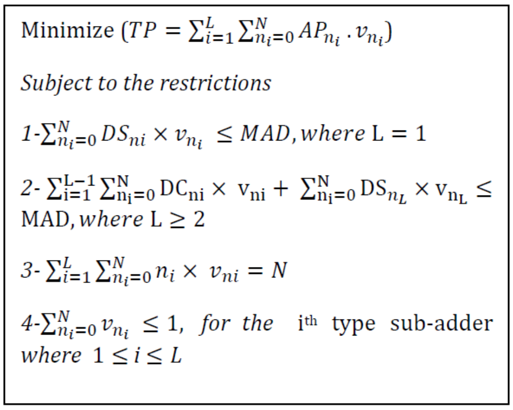
Minimize [Total Power of a hybrid adder (TP)]

Subject to the restrictions

1- Overall delay of a hybrid adder (OD) ≤ Maximum allowed delay (MAD)

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

Here the objective is to allocate the optimal bit-widths (ni) so as to minimize the power consumption of the hybrid adder, subject to the constraints imposed on their values. The total power (TP) of a hybrid adder is derived in equation (10), where Pni is the power of sub-adders for different bit-widths (ni). Using equations (10) and (11), the problem of modeling a hybrid adder with a minimum power under delay constraint is described in Fig.11.



Minimize $(TP = \sum_{i=1}^{L}\sum_{n_i=0}^{N} AP_{n_i} \cdot v_{n_i})$

Subject to the restrictions

1-$\sum_{n_i=0}^{N} DS_{ni} \times v_{n_i} \leq MAD$, *where* $\mathrm{L} = 1$

2- $\sum_{i=1}^{L-1}\sum_{n_i=0}^{N} DC_{ni} \times v_{ni} + \sum_{n_i=0}^{N} DS_{n_L} \times v_{n_L} \leq \mathrm{MAD}$, *where* $\mathrm{L} \geq 2$

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

4-$\sum_{n_i=0}^{N} v_{n_i} \leq 1$, *for the* ith type sub-adder *where* $1 \leq i \leq L$

Fig 11. Power optimization model of a hybrid adder

4. Experimental Results

All classical adders under consideration are designed with varied bit widths from 4 to 128 bits and coded in VHDL. Simulation of various adder designs is carried out with ISE simulator. The Xilinx ISE 9.1 tool is used to synthesize adder designs. All designs have been targeted to FPGA-Virtex5 (XC5LX330T).

Summary of delay and power reports for different bit widths of the classical 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 in terms of the number of used slices.

The applicability and effectiveness of the proposed design scheme are examined through various experiments. In doing so, several topologies of a hybrid adder consisting of two (L = 2) and three (L = 3) sub-adders are investigated using the optimization models to allocate the optimal bit-widths and sub-adders types. Possible bit-widths of the various sub-adders are up to 128 bits. The mathematical optimization models of the proposed hybrid adder architecture are coded and solved using “LINGO 13.0” software package from LINDO Systems. With changing maximum allowed power (MAP), the optimal hybrid adder topologies are found using the delay optimization model. Several topologies of the optimized hybrid adder are indicated in Table.2. A symbol “|” is used to clarify that the sub-adders are connected linearly. For example, the form RCA(31)|CSKA(97) illustrates that CSKA is located to the 97-low order bits and that RCA is located to the 31-high order bits of RCA|CSKA 128-bit hybrid adder.

Similarly, the delay constrained power optimization of the 128-bit hybrid adder is obtained by “LINGO 13.0”. With changing maximum allowed delay (MAD), the optimal hybrid adder topologies are found using the power optimization model. The obtained optimal hybrid adder topologies are shown in Table. 3. It is observed that the optimization results shown in Tables 2 and 3 are the same for the two optimization models. So, the optimal hybrid adder topology can be obtained by either using delay optimization model or power optimization model.

For all adder designs, power x delay product (P x T) are normalized (P x $T_{\text{other adders}}$ / P x $T_{\text{RCA/CLA}}$) and shown in Fig. 12. From the results obtained, we can infer that RCA(23)|CLA(105) hybrid adder gives the highest performance compared to other adder implementations with delay=23.5 ns and power=1434 mw. Improvements introduced by RCA(23)|CLA(105) hybrid adder, shown in Table.4 , in comparison to other adder designs are calculated using (P x T) Improvements of RCA(23)|CLA(105) = ((1 - (P x $T_{\text{RCA/CLA}}$ / P x $T_{\text{other adders}}$)) x100).

Table 1. Summary of delay (ns) and power (mw) reports of classical adders

		4 bit	8 bit	16 bit	32 bit	64 bit	128 bit
RCA	DS(ns)	7.4	9.6	13.8	19.7	34.4	83.1
	DC(ns)	6.8	9.5	12.8	19.6	34.1	82.9
	Power(mw)	31.16	59.77	135.96	234.46	557.45	1088.35
CLA	DS(ns)	6.6	8.78	9.96	10.9	15.6	20.8
	DC(ns)	6.3	6.69	7.32	8.68	11.5	19.2
	Power(mw)	33.97	63.25	154.54	244.97	700.47	1643.46
CSKA	DS(ns)	7.0	9.02	11.0	17.4	23.8	41.1
	DC(ns)	6.2	8.17	10.6	16.1	22.9	40.9
	Power(mw)	36.90	66.82	131.47	244.97	517.07	1197.54
CSLA	DS(ns)	6.5	9.3	11.3	16.9	25.1	53.7
	DC(ns)	5.6	8.6	10.9	16.3	24.8	53.4
	Power(mw)	39.96	74.23	135.96	354.02	784.21	2211.22
PPA-SK	DS(ns)	7.1	9.6	11.2	14.5	24.6	35.4
	DC(ns)	7.0	9.4	11.0	13.9	24.4	34.5
	Power(mw)	36.90	56.39	127.05	306.18	761.04	2211.22
PPA-BK	DS(ns)	7.0	9.1	12.9	16.6	25.2	49.4
	DC(ns)	7.0	8.92	12.87	16.62	23.51	46.31
	Power(mw)	36.90	63.25	118.40	239.69	537.11	1427.72

Table 2. Summary of delay optimization results obtained using LINGO

MAP(mw)	Hybrid adder topology	OD(ns)
1434	*RCA(23)|CLA(105)*	*23.50*
1434	CLA(105)|RCA(23)	25.20
1219	CSKA(16)|CLA(112)	34.48
1427	BK(23)|CLA(105)	29.56
1198	CSKA(128)|CSLA(0)	41.15
1198	CSKA(128)|BK(0)	41.15
1198	CSKA(128)|SK(0)	41.15
1096	RCA(31)|CSKA(97)	51.59
1342	RCA(64)|CSLA(64)	59.30
1200	RCA(78)|SK(50)	64.90
1098	RCA(63)|BK(65)	57.80
1403	RCA(23)|CSKA(4)|CLA(101)	30.70
1403	CSKA(4)|RCA(23)|CLA(101)	30.01
1403	CLA(101)|CSKA(4)|RCA(23)	33.15
1396	BK(4)|CSKA(24)|CLA(100)	36.27
1050	BK(39)|CSKA(64)|SK(25)	54.13
1298	SK(32)| RCA(40)|CSLA(56)	60.50
1078	BK(32)|RCA(32)|CSLA(64)	61.10

Table 3. Summary of power optimization results obtained using LINGO

MAD(ns)	Hybrid adder topology	TP (mw)
23.50	*RCA(23)|CLA(105)*	*1434*
25.20	CLA(105)|RCA(23)	1434
34.48	CSKA(16)|CLA(112)	1219
29.56	BK(23)|CLA(105)	1427
41.15	CSKA(128)|CSLA(0)	1198
41.15	CSKA(128)|BK(0)	1198
41.15	CSKA(128)|SK(0)	1198
51.59	RCA(31)|CSKA(97)	1096
59.30	RCA(64)|CSLA(64)	1342
64.90	RCA(78)|SK(50)	1200
57.80	RCA(63)|BK(65)	1098
30.70	RCA(23)|CSKA(4)|CLA(101)	1403
30.01	CSKA(4)|RCA(23)|CLA(101)	1403
33.15	CLA(101)|CSKA(4)|RCA(23)	1403
36.27	BK(4)|CSKA(24)|CLA(100)	1396
54.13	BK(39)|CSKA(64)|SK(25)	1050
60.50	SK(32)| RCA(40)|CSLA(56)	1298
61.10	BK(32)|RCA(32)|CSLA(64)	1078

Table 4. Power x delay (P x T) Performance comparison and the improvements introduced by RCA(23)|CLA(105) hybrid adder.

Adder topology	Power(mw)	Delay(ns)	(P x T) (normalized	*(P x T)* Improvements%
RCA(23)|CLA(105)	*1434*	*23.5*	*1.00*	*0.00*
CLA(105)|RCA(23)	1434	25.2	1.07	6.75
CSKA(16)|CLA(112)	1219	34.4	1.25	19.82
BK(23)|CLA(105)	1427	29.5	1.25	20.11
CSKA(128)|CSLA(0)	1198	41.1	1.46	31.64
CSKA(128)|BK(0)	1198	41.1	1.46	31.64
CSKA(128)|SK(0)	1198	41.1	1.46	31.64
RCA(31)|CSKA(97)	1096	51.5	1.68	40.40
RCA(64)|CSLA(64)	1342	59.3	2.36	57.65
RCA(78)|SK(50)	1200	64.9	2.31	56.73
RCA(63)|BK(65)	1098	57.8	1.88	46.90
RCA(23)|CSKA(4)|CLA(101)	1403	30.7	1.28	21.76
CSKA(4)|RCA(23)|CLA(101)	1403	30.0	1.25	19.96
CLA(101)|CSKA(4)|RCA(23)	1403	33.1	1.38	27.54
BK(4)|CSKA(24)|CLA(100)	1396	36.2	1.50	33.44
BK(39)|CSKA(64)|SK(25)	1050	54.1	1.69	40.71
SK(32)| RCA(40)|CSLA(56)	1298	60.5	2.33	57.09
BK(32)|RCA(32)|CSLA(64)	1078	61.1	1.95	48.84
RCA	1088	83.1	2.68	62.74
CLA	1643	20.8	1.01	1.42
CSKA	1197	41.1	1.46	31.53
CSLA	2211	53.7	3.52	71.62
SK	2211	35.4	2.32	56.95
BK	1427	49.4	2.09	52.22

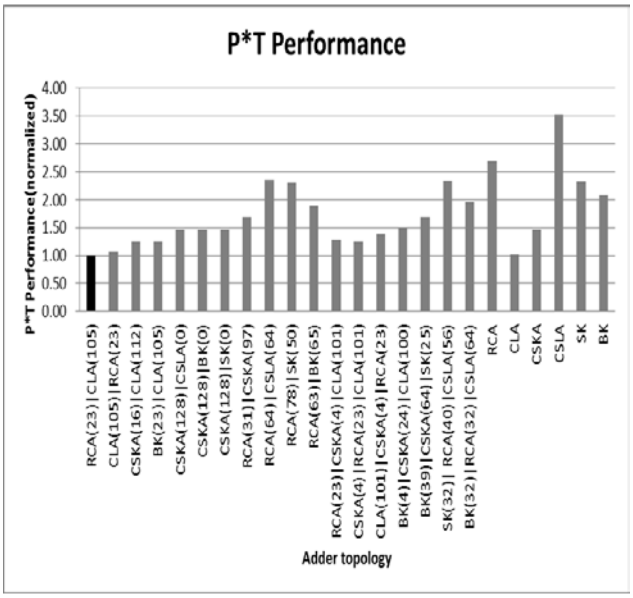


Fig 12. Power x delay (P x T) Performance comparison of hybrid and classical adders

Figure 13 shows the power x delay product (P x T) of the RCA(23)|CLA(105) hybrid adder. The obtained results shown in Table .4 and Fig. 13 illustrate that the RCA(23)|CLA(105) hybrid adder topology provides an improvement in (P x T) performance from (min 1.42 to max 71.6%) compared to the performance of classical adders. In addition, good candidates such as BK|CLA and CSKA|CLA are found for replacing classical BK and CSKA adders, respectively. The results illustrate clearly that the proposed design scheme successfully can generate the optimal topology of a hybrid adder. Besides, it indeed allows delay/power tradeoffs much better than the classical adder designs.

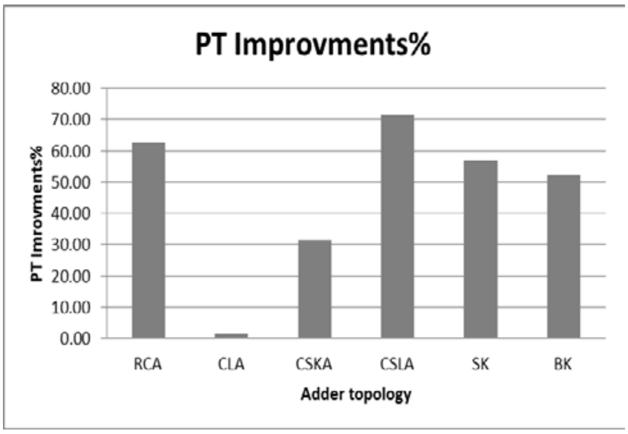


Fig 13. Power x delay (P x T) Improvements introduced by RCA|CLA hybrid adder in comparison with classical adders

5. CONCLUSION

This paper presents a mathematical modeling based technique for designing a hybrid adder through allocating the optimal bit-widths and types of classical adders constituting a hybrid adder. The proposed optimization models consider two aspects design delay and power. They are based on a mathematical modeling of the proposed hybrid adder architecture following the principle of Linear Programming. Two models, delay optimization under power constraint and power optimization under delay constraint, are introduced. All adders design are developed using Xilinx ISE 9.1 EDA tool. Optimization models of the proposed hybrid adder are coded using LING 13.0 package.

Various experiments are presented to demonstrate the effectiveness of the proposed method. The results indicate that the proposed scheme successfully allocates simultaneously and in a high level systematic way the optimal bit-widths of sub-adders constituting a hybrid adder. In this regard, new design points are defined to a hybrid adder design for effective performance such as the RCA(23)|CLA(105) hybrid adder. Compared to the classical adder counterparts, the RCA/CLA hybrid adder provides an improvement in power x delay (P x T) performance from (min 1.42 to max 71.6%). The results also demonstrate that the proposed design scheme gains high flexibility in allowing design tradeoffs between delay and power compared to classical adder designs. Thereby, the efforts of this work will help support the overall goal to fabricate a tailored hybrid adder according to application demands.

References

[1] Hennessy, J.L & Patterson, D.A. (1990). Computer Architecture: A Quantitative Approach. Morgan Kaufmann.

[2] Franklin, M.A. & Pan, T. (1994). Performance Comparison of Asynchronous Adders. In: Symp. on Advanced Research in Asynchronous Circuits and Systems, pp. 117-125.

[3] Garside, J.D. (1993). CMOS VLSI Implementation of an Asynchronous ALU. In: Proceedings of the IFIP WG10.5 Working Conference on Asynchronous Design Methodologies, pp. 181-192.

[4] Koren, I. (2002). Computer Arithmetic Algorithms, (2nd ed.), A K Peters Ltd., Canada.

[5] Lynch, T. & Swartzlander, E. E. (1993). A spanning Tree Carry Look-ahead Adder. IEEE Trans.on Computers, 41 (August 1992), 931-939.

[6] Wang, Y., Pai, C., & Song, X. (2002). Design of Hybrid Carry Look-ahead/Carry–Select Adders. IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 49, no. 1.

[7] Li, J., Yu, J., & Huang, Y. (2005). A Design Methodology for Hybrid Carry Look-ahead/Carry-Select Adders with Re-configurability. In: Proc. 15th VLSI/CAD Symp., Pintung.

[8] Lakshmanan, Meaamar, A., & Othman, M. (2006). High-Speed Hybrid Parallel-Prefix Carry- Select Adder Using Ling's Algorithm. In: ICSE2006 Proc., Kuala Lumpur, Malaysia.

[9] Nithya, J. & Ramesh, S. R., (2019). Design of Delay Efficient Hybrid Adder for High Speed Applications. 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 374-378.

[10] Joel, A., & Manjith, R. (2019). Design of Low Power High Speed Hybrid Adder Using Gdi Technique. International Journal of Advance and Innovative ResearchVolume 6, Issue 1.

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

[12] Weinberger, & Smith, J. L. (1958). Logic for High- Speed Addition, National Bureau of

[13] Standards, Circulation 591, pp. 3-1. Kilburn, T., Edwards, D.B.G., & Aspinall, D. (1959). Parallel Addition in Digital Computers: A New Fast "Carry" Circuit. In: IEE, Vol.106, Pt.B. pp.464.

[14] Sklansky, J. (1960). Conditional Sum Addition Logic. IRE Trans. Electron. Comput., vol. EC-9, no. 6, pp. 226–231.

[15] Brent, R.P., & Kung, H.T. (1982). Regular Layout for Parallel Adders. In: IEEE Trans. Comput., vol.31, no. 3, pp. 260–264.

[16] Vanderbei, R.J. (2001). Linear Programming: Foundations and Extensions. (2nd ed.), USA.

Authors' Profiles

Mahmoud A. M. Alshewimy is currently working as Lecturer in Department of Computer and Automatic Control Engineering, Faculty of Engineering, Tanta University, Egypt. He received M.Sc. Computer Engineering from Tanta University in 2006 and Ph.D in Computer Engineering from Istanbul University in 2013. His research interests are in Object Recognition, Digital Systems Design using VHDL and IoT Applications.

How to cite this paper: Mahmoud A. M. Alshewimy, " Power-Time Efficient Hybrid Adder Design Based On Lp With Optimal Bit-Width Generation ", International Journal of Engineering and Manufacturing (IJEM), Vol.10, No.4, pp.1-12, 2020. DOI: 10.5815/ijem.2020.04.01