A New Parallel Prefix Adder Structure With Efficient Critical Delay Path And Graded Bits Efficiency In CMOS 90nm Technology

H. Moqadasi
Dept. Elect. Engineering
Shahed university
Tehran- IRAN
h.moqadasi AT shahed.ac.ir

M. B. Ghaznavi-Ghoushchi
Dept. Elect. Engineering
Shahed university
Tehran-IRAN
ghaznavi AT shahed.ac.ir

Abstract— In this work we have proposed an efficient parallel prefix adder (PPA) that is a variation of the popular Brent-Kung PPA. In this proposed adder, with a glance on the sklansky adder and by varying the graph topology, we have reduced the number of stages in the critical delay path in comparison with Brent-Kung PPA and so lowered the delay. This advantage is along with a little increase in power consumption and area due to increasing the cells number so that the consequent Power Delay Product (PDP) is improved. Also the performance of our proposed PPA increases by increasing the input bits number from 32 bits up to higher orders. The experimental results indicate that our proposed adder is faster than the Brent–Kung adder by 10.1% in 32 bits, 17.2% in 64 bits and 21.3% in 128 bits and the consequent PDP is reduced by 8.8% in 32 bits, 15.6% in 64 bits and 19.5% in 128 bits. Circuit level simulations were performed with SPICE and CMOS 90nm technology.

Keywords: Parallel Prefix Adder; Brent-Kung Adder; Sklansky Adder; Power-Delay Product; CMOS Technology;

I. INTRODUCTION

VLSI adders have a lot of applications in many fields such as arithmetic and logic units (ALU’s), microprocessors, floating point arithmetic units, memory addressing and program counting units and etc [1], [2], [3], [4], [5] so, much attention has been paid to the increasing of their performances. The prefix operation is an essential operation which has applications in the design of fast adders. Many fast adders were proposed based on the prefix computation [6]. Among all adders the parallel prefix adders (PPA’s) are in the spotlight recently [7]. That is because they use a tree network for reducing latency to log₂(n) through the carry path compared to O(n) in the RCA where ‘n’ is number of input bits[8], [9], [10] and also they have regular structure and simple cells [4]. Besides, they are appropriate for implementation of adders with wider word lengths [10]. Based on the prefix structure, various PPAs are presented up to now. Sklansky [11], Kogge-Stone [12] and Brent-Kung [13] adders are the classic and representative PPAs. The Sklansky adder has the minimum logic depth which is log₂(n) (‘n’ is number of input bits) and also has the minimum wiring tracks. These performances are combined with a drawback that is the large fan-out that increases linearly with logic depth. Therefore causes a big layout area and lowers the circuit speed [2]. Ladner-Fischer prefix network has higher logic depth in comparison with the Sklansky adder with a lower fan-out in the critical delay path nodes [8]. The Kogge-Stone adder is the fastest PPA and has the optimal logic depth as log₂(n)and low fan-out in its prefix network, instead the wire connections between its stages are complex, this results layout implementation problems. Also their computational cells are numerous which cause to more chip area and more power consumption [2], [4], [14]. The Brent-Kung adder is proposed to reduce the drawbacks of Kogge-Stone adder with less carry merge cells and less complexity. But in turn it has some disadvantages. For example it has higher logic depth and hence more delay [15]. Han-Carlson adder is the combination of the Brent Kung and Kogge Stone adders, which trades between the wiring tracks (similarly the prefix cells) and the logic depth [14]. Hybrid Han-Carlson adder is a variation of the Han-Carlson adder which is suited for large word sizes. In this adder delay increases slightly but the complexity, silicon area and power consumption are reduced considerably [16]. Knowles has presented a family of adders that trades between wiring tracks and the fan-out of intermediate nodes [17]. Harris has presented a three-dimensional taxonomy that describes the trade-offs among existing parallel prefix adders [18].

In this paper we have presented a parallel prefix adder based on the Brent-Kung adder so that by taking idea form the Sklansky adder the lateral fan-out of some computational nodes in the prefix tree is increased and the number of stages in its critical delay path is reduced. Consequently the speed and also power delay product is improved. Our simulations show this PPA lets us have an improved efficiency by increasing the inputs word sizes from 32 bits up to higher bit numbers.

In this paper we have presented a parallel prefix adder based on the Brent-Kung adder so that by taking idea form the Sklansky adder the lateral fan-out of some computational nodes in the prefix tree is increased and the number of stages in its critical delay path is reduced. Consequently the speed and also power delay product is improved. Our simulations show this PPA lets us have an improved efficiency by increasing the inputs word sizes from 32 bits up to higher bit numbers.

This paper is organized as follows: Section II describes the theme and basics of parallel prefix addition. Section III reviews the Brent-Kung adder, section IV presents our proposed PPA and section V is dedicated to the simulations. Finally section VI is conclusions.
II. PARALLEL PREFIX ADDITION

Parallel prefix adders also known as carry tree adders [19] are based on the Carry Look Ahead (CLA) adder and similar to it perform addition in three steps shown in Fig. 1 [20]. In the first step the carry generate and carry propagate signals which are intermediate variables [18]. Are pre-computed from input bits (pre-computation). Second step calculates all the carry signals in parallel (prefix-network), finally in the third step, sum signals are computed from the propagate and carry signals generated in the previous stages (post-computation) [21].

The addition logic of three stages in a PPA adder follows (1) and (2) and (3) where $A_i$ and $B_i$ and $C_{in}$ are inputs, $S_i$ is the output sum and $G$ and $P$ are the generate and propagate prefix signals $(i$ and $k$ are indexes representing bit positions) [20].

Pre-Computation :  
\[
G_{ij} = A_i \cdot B_i \quad P_{00} = C_{in} \\
P_{ij} = A_i \oplus B_i \quad P_{00} = 0  \tag{1}
\]

Prefix-Network :  
\[
G_{i,j} = G_{i,k} + P_{i,k} \cdot G_{k,i,j} \\
P_{i,j} = P_{i,k} \cdot P_{k,i,j}  \tag{2}
\]

Post-Computation :  
\[
S_i = P_i \oplus G_{i,j}  \tag{3}
\]

In order to configure the prefix network, the dot operator "\(\cdot\)" defined as (4) and the semi-dot operator "\(\circ\)"defined as (5) are used based on [22] with a little modification for the semi-dot operation.

\[
(g_{i,k} \cdot p_{i,k}) \circ (g_{m,j}, p_{m,j}) = (g_{i,k} + p_{i,k} \cdot g_{m,j}, p_{i,k} \cdot p_{m,j}) = (G_{i,j}, P_{ij})  \tag{4}
\]

\[
(G_{i,j}, P_{ij}) \circ C_k = (g_{i,j} + p_{i,j} \cdot g_{m,j}) = C_i  \tag{5}
\]

In (4), the dot operator works with two pairs of bits $(g_{i,k}, p_{i,k})$ and $(g_{m,j}, p_{m,j})$ where $j < k \leq m < i$ [22]. A new pair of bits results from this operator named group-generate and group-propagate signals and can be again combined with another pairs of bits by another dot or semi-dot operator. Semi-dot operator works with the pair $(G_{i,j}, P_{ij})$ and $C_k$. This procedural use of dot and semi-dot operators configures a prefix network which finally generates the carry signals [14].

The prefix operator has two main properties included idempotency and associativity that lets evaluate the prefix operations according to a binary tree and have various prefix networks. But it does not have the commutativity property which means the order of two neighboring input operands must not be altered [23], [24]. Fig.2 (a), (b), (c), (d) shows the gate-level structure of dot cell, semi-dot cell, pre-processor cell and post-processor (sum) cell respectively. These cells are used in the circuit level structure of PPAs.

III. BRENT-KUNG ADDER

According to section II descriptions the associativity property of prefix operator lets us have various prefix tree networks for carry generation. Brent-Kung adder is one of the main prefix structures was proposed by Brent and Kung with a regular layout [13]. This adder has been proposed to solve the disadvantages of Kogge-Stone adder. Fig. 3 shows the Graph topology of a 32-bit Brent-Kung adder.

This adder has the least computational nodes among all PPAs [8] for a certain number of input bits which results the reduced area. Instead this structure has maximum logic depth [25] which yields the increase in the adder’s latency in comparison with other structures [8] so lowers the speed [26]. In fact this adder trades area for logic depth and is an excellent instance for the PPA with maximum logic depth and minimum area [26]. Furthermore Brent-Kung adder is an efficient choice for nowadays synthesis tools and is the state of the art [27].

IV. PROPOSED ADDER

Fig. 4 shows the graph structure of our proposed PPA for 32 bits which is a variation of Brent-Kung adder, the critical delay path is shown by a solid red line. Our proposed adder is the same as Brent-Kung adder in most stages except in the stage 2 and hence the stage9, in fact the modification of stage 2 results in the modification of stage9. The objective of our modification was to reduce the PDP, based on variation in PPA graph topology and for this, with ideas from the sklansky adder and specially its second stage we have increased the lateral fan-out of dot cells in the first stage up to two, these cells are in bit positions (N) follow (6). That is like the manner of stage2 in the sklansky adder.

\[
N = 8k + 5 ; \quad k = 1, 2, 3, \ldots, 2^{L-3} - 1 ; \quad L = \log_2 n  \tag{6}
\]

This modification reduces the number of stages in the critical delay path from 8 in the 32-bit Brent-Kung adder to 7 in our proposed adder, by the way decreases the delay of critical path in comparison with the Brent-Kung adder. Of course a little increase in power consumption is another outcome of our modification which is due to an increase in the number of dot cells so that in overall, the product of power and delay improves as acceptable. In this paper we have presented our design for 32-bit inputs but this idea can be extended for
designing larger adders with larger word sizes and brings more efficiency. Section V describes the results and outcome performances with more details.

V. SIMULATION

In this section we have simulated the Brent-Kung adder and also our proposed design from 4 to 128 bits with HSPICE and in CMOS 90nm technology. Basic cells are implemented according to the logic structure shown in Fig. 2 so that the l of transistors are chosen as the lmin of used technology equal to 90nm, and PMOS transistors widths as two times of NMOS types. In all simulations supply voltage was considered as 1.2 volt. In order to show the preferences of our proposed adder we have examined the specifications of both adders and compared them in each case. As we mentioned in section IV the Brent-Kung adder and our proposed one are the same in 4 bits and also 8bits so the results become equal and we have no performance for these bit widths, but by increasing the word size from 16 bits, differences are appeared which result performances. Following, we examine and compare both adders in various aspects.

Table I shows the delay of both Brent-Kung and proposed PPAs by varying the word sizes of inputs from 4 to 128 Bits and represents that in the word size of 4 and 8 delay values are the same because of the equality of both adders, but in larger bit lengths, our proposed design exhibits better and has lower delay. Also, It seems in the word sizes larger than16, the higher the word size the better the delay performance which means by increasing the inputs word sizes the percentage of delay lowering, increases too. This is shown in Fig. 5 (a); furthermore this figure shows that the slope of delay increase is smoother in our work in comparison with the Brent-Kung PPA which means our proposed PPA performs better by grading the input word size from the aspect of speed.

Also our work has some drawbacks in the cost of delay lowering. Table II lists the number of total computation nodes for each of the Brent-Kung and proposed PPAs. In any PPA adder the number of semi-dot cells is equal to the length of inputs so the number of total cells is the number of dot cells plus the number of semi-dot cells (the word size length). Our
Table II shows, in word sizes larger than 16 bits, our proposed PPA caused the increase in the number of dot cells so increases the number of total cells, hence results a grown area in comparison with the Brent-Kung adder. In fact this is the drawback of our design in domain of parameter trading. The chart in Fig. 6 represents this disadvantage. Beside, increasing the cells number roles as a factor of increasing the average power consumption according to Table III. But this secondary drawback is not considerable. Fig. 5 (b) shows this drawback negligibility. The important parameter for us was not only the power, but also the delay, so we focused on the product of them (PDP) as the Figure Of Merit (FOM) in order to improve the Brent-Kung adder performance while keeping other parameters in acceptable level. Table IV and Fig. 5 (c) present the PDP variation of both PPAs versus word size. As shown

<table>
<thead>
<tr>
<th>Word Size</th>
<th>4bits</th>
<th>8bits</th>
<th>16bits</th>
<th>32bits</th>
<th>64bits</th>
<th>128bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brent-Kung</td>
<td>0%</td>
<td>0%</td>
<td>13.2%</td>
<td>10.1%</td>
<td>17.2%</td>
<td>21.3%</td>
</tr>
<tr>
<td>Proposed</td>
<td>0%</td>
<td>0%</td>
<td>3.5%</td>
<td>4.9%</td>
<td>6.2%</td>
<td>7.8%</td>
</tr>
</tbody>
</table>

**Table I: Delay (Pico second).**

<table>
<thead>
<tr>
<th>Word Size</th>
<th>4bits</th>
<th>8bits</th>
<th>16bits</th>
<th>32bits</th>
<th>64bits</th>
<th>128bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brent-Kung</td>
<td>478.83</td>
<td>621.33</td>
<td>889.15</td>
<td>1167.43</td>
<td>1455.88</td>
<td>1755.67</td>
</tr>
<tr>
<td>Proposed</td>
<td>478.83</td>
<td>621.33</td>
<td>771.29</td>
<td>1049.40</td>
<td>1204.58</td>
<td>1380.18</td>
</tr>
</tbody>
</table>

**Table II: The number of computation nodes.**

<table>
<thead>
<tr>
<th>Word Size</th>
<th>4bits</th>
<th>8bits</th>
<th>16bits</th>
<th>32bits</th>
<th>64bits</th>
<th>128bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brent-Kung</td>
<td>5</td>
<td>12</td>
<td>27</td>
<td>58</td>
<td>121</td>
<td>248</td>
</tr>
<tr>
<td>Proposed</td>
<td>5</td>
<td>12</td>
<td>28</td>
<td>61</td>
<td>129</td>
<td>269</td>
</tr>
</tbody>
</table>

Fig. 4: 32-Bit Proposed PPA.

Fig. 5: The variation of (a) Delay, (b) Average power, (c) PDP of Brent-Kung PPA and proposed PPA versus word size.
the PDP of our proposed design is lower than Brent-Kung adder for word sizes larger than 8 bits and from 32 bits this performance increases by growing the word size, in other words the growth slope is smoother than Brent-Kung adder.

The important property of our proposed design is reducing the number of stages in the critical delay in comparison with the Brent-Kung PPA which seems it was the main reason of delay and hence PDP reduction in our design. Considering the solid red line in Fig. 3 and Fig. 4 that shows the critical delay path we can see this stage reduction in the 32-bit Brent-Kung and proposed adders. Also Table V presents it by varying the word size and also shows the percentage of stage reduction in each case. The maximum fan-out of the critical delay path was another attractive parameter in our design which defined as the fan-out of the computational node with maximum fan-out existed within the critical delay path. As Table VI shows, in our design this parameter is kept the same as Brent-Kung adder in 4, 8, 16, 32 bit lengths but in the 64 and 128 bits case we have an increase in the fan-out but in overall did not destroyed our design performance. Since we dealt with multiple parameters include "N "as the number of computational nodes, "D "as delay, "P "as the average power, "PDP "as the power delay product, "L "as the number of stages within the critical delay path and "F "as maximum fan-out of the critical delay path, and by considering the results presented in Table I to Table VI we have sketched the Design space of a 64-bit Brent-Kung and proposed PPA shown in Fig. 7. We can see the contribution and trade-offs between various parameters. This figure obviously shows the performances and drawbacks of our design versus Brent-Kung adder. As shown our design has lowered the stages (L), and increased the fan-out (F) but trading these two parameters resulted in reducing the delay. From the other hand in our proposed PPA the number of cells (N) are increased somewhat, consequently augmented the average power consumption (P) negligibly. Overall, the total PDP is reduced.

An important point is that these improvements are only for a single adder as a logical building block and since in most logic applications such as multicore processors we use thousands of these blocks, this improvement becomes considerable.

VI. CONCLUSIONS

This research presents an efficient parallel prefix adder based on the Brent-Kung adder where by taking idea from the Sklansky adder and varying the graph topology, the number of stages in its critical delay path is reduced. Consequently the speed and also power delay product is improved. Simulation results confirmed that the proposed adder is faster than the Brent–Kung adder by 10.1% in 32 bits, 17.2% in 64 bits and 21.3% in 128 bits and the consequent PDP is reduced by 8.8% in 32 bits, 15.6% in 64 bits and 19.5% in 128 bits. About the proposed adders it seems that the higher the input bits number, the higher the improvement percentage of performance. The
ACKNOWLEDGMENT
The authors wish to thank the following persons for their helps: Prof. David Money Harris from Harvey S. Mudd university, CA, USA. Dr. Naser Mohammadzadeh, Dr. Ali Asghar Bagheri Soulia from Department of Electrical and Computer Engineering Shahed university, Tehran, Iran. Dr. Mohammad Sharifkhani from Sharif university, Tehran, Iran. Miss Bahare Qorbany and Miss Hamideh Amir. The first author specially wishes to thank her beloved parents for their supports and encouragements.

REFERENCES