Closing the Gap between Carry Select Adder and Ripple Carry Adder: A New Class of Low-power High-performance Adders

Behnam Amelifard	Farzan Fallah	Massoud Pedram
University of Southern California	Fujitsu Laboratories of America	Universality of Southern California
amelifar@usc.edu	farzan@fla.fujitsu.com	pedram@ceng.usc.edu

Abstract

Based on the idea of sharing two adders used in the Carry Select Adder (CSA), a new design of a low-power high-performance adder is presented. The new adder is faster than a Ripple Carry Adder (RCA), but slower than a CSA. On the other hand, its area and power dissipation are smaller than those of a CSA.

1. Introduction

The increase in the popularity of portable systems as well as the rapid growth of the power density in integrated circuits have made power dissipation one of the important design objectives, second only to performance. Because adders are one of the most widely used components in integrated circuits, designing efficient adders has been the goal of much research in VLSI design. While Ripple Carry Adders (RCAs) have the most compact design ($O(n)$ area) among all types of adders, they are the slowest types of adders ($O(n)$ time). On the other hand, Carry Look-ahead Adders (CLAs) are the fastest adders ($O(\log(n))$ time), but they are the worst from the area point of view ($O(n\log(n))$ area) [2]. Carry Select Adders (CSAs) have been considered as a compromise solution between RCAs and CLAs ($O(\sqrt{n})$ time and $O(2n)$ area) because they offer a good tradeoff between the compact area of RCAs and the short delay of CLAs. As a result, some effort has been done to improve the efficiency of this kind of adder [1-5]. In [1], for example, an area efficient adder has been proposed which uses an increment circuit instead of one of the two adder blocks which add high bits.

In this research, based on the idea of sharing the two adders that are typically used in the CSA, a new architecture is proposed which is more compact and power efficient than the CSA. Additionally it is shown that by using this idea iteratively, one can effectively trade area for delay. More specifically, the delay of the proposed adder is $O(\sqrt{2n})$ while its area is $O((1+\alpha)n)$, where $\alpha<1$.

The rest of the paper is organized as follows. In Section 2, CSA is introduced and a partitioning methodology for hierarchical design of CSA is presented. In Section 3 a new architectures is proposed to reduce area and power consumption of CSA. Section 4 demonstrates simulation results of the new architecture, while Section 5 concludes the paper.

2. Carry Select Adder

The conventional n-bit CSA consists of one $n/2$-bit adder for the lower half of the bits and two $n/2$-bit adders for the upper half of the bits. Of the two latter adders, one performs the addition with the assumption that $C_{in}=0$, whereas the other does this with the assumption that $C_{in}=1$. Using a multiplexer and the value of carry out that is propagated from the adder for the $n/2$ least significant bits, the correct value of the most significant part of the addition can be selected. Although this technique has the drawback of increasing the area, it speeds up the addition operation. The architecture of the adder is shown in Figure 1.

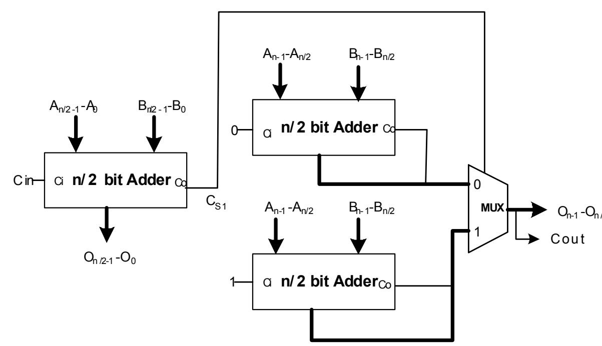

Figure 1: Architecture of a two-stage CSA

Using the idea of CSA iteratively, the delay of the adder can significantly be reduced. It can be shown that if the delay of multiplexers is negligible, the delay of the iterative CSA will grow with the square root of the number of bits. A more accurate analysis is required to find the delay of CSA if the delay of multiplexers is not negligible. In the following, such an analysis is presented and will be confirmed with simulation results.

Assume that for constructing an n-bit CSA, the bits are partitioned into m groups where group i contains P_i bits such that the bit width of the least significant part is P_1 and the bit width of the most significant part is P_m. In CSA, the adders of all parts except for P_1 should be duplicated. A schematic of a three-stage CSA (m=3) is shown in Figure 2.

In this figure, C_{s1} is the carry propagated from the first part to the second one, while C_{S2} is the carry propagated from the second part to the third one. Moreover, *Cout* is the carry-out of the n-bit addition operation.

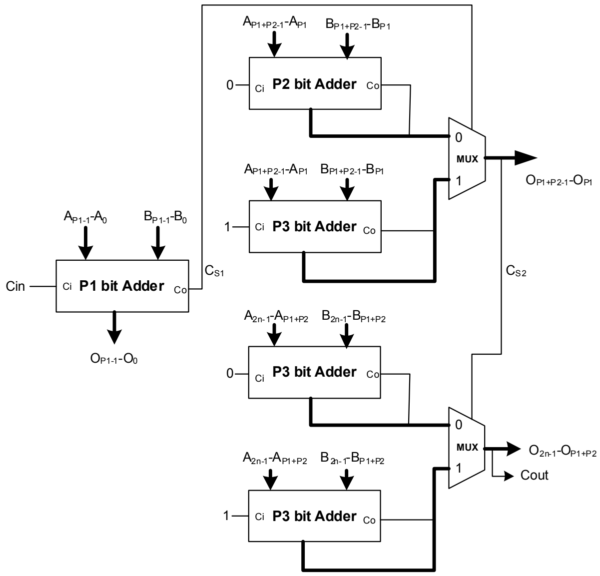

Figure 2: Three-stage CSA

The problem of designing the fastest CSA can be expressed as finding m and P_i's ($1 \le i \le m$) to minimize the delay of the circuit. The basic assumption of the following analysis is that the delay of an RCA is a linear function of the number of bits. Moreover, it is assumed that by using techniques such as buffer insertion, the capacitive load for each operational module (i.e., the adders and the multiplexers) is constant.

Assume the delay of a (one-bit) full adder is A and the delay of a multiplexer is M. The first assumption can be written as,

$$delay(k\ bit\ adder) = k \times delay(full\ \ adder) \tag{1}$$

To design the fastest CSA, we should balance all paths from the inputs of the adder to $Cout$. These delays may be written as:

$$\begin{aligned} D_1 &= P_1.A + (m-1)M \\ D_2 &= P_2.A + (m-1)M \\ &\cdots \\ D_i &= P_i.A + (m-i+1)M \\ &\cdots \\ D_m &= P_m.A + M \end{aligned} \tag{2}$$

where D_i is the delay from the i^{th} adder to $Cout$. By setting these delays equal to each other, it can be easily shown that,

$$P_i = P_1 + (i-2)\frac{M}{A},\ (2 \le i \le m) \tag{3}$$

Considering the fact that $\Sigma P_i = n$, the values of P_i's will be,

$$P_1 = P_2 = \frac{n}{m} - \frac{M}{2A}\frac{(m-2)(m-1)}{m} \tag{4}$$

$$P_i = \frac{n}{m} - \frac{M}{2A}\frac{(m-2)(m-1)}{m} + \frac{M}{A}(i-2),\ (3 \le i \le m) \tag{5}$$

So, the delay of the adder can be written as,

$$delay = P_1 A + M(m-1) = A.\frac{n}{m} + \frac{M}{2}\frac{(m-1)(m+2)}{m} \tag{6}$$

To minimize the delay:

$$\frac{\partial delay}{\partial m} = 0 \Rightarrow m^* = \sqrt{2\left(n\frac{A}{M} - 1\right)} \approx \sqrt{2n\frac{A}{M}} \tag{7}$$

Using this value for m, from (4), it follows that,

$$P_1^* = \frac{M}{2A}\left(3 - \sqrt{\frac{2M}{nA}}\right) \approx \frac{3}{2}\frac{M}{A} \tag{8}$$

Now, from (6), the minimum delay of a CSA can be obtained as:

$$delay^* = \sqrt{2nAM} + \frac{1}{2}M \tag{9}$$

Moreover, it is easily seen that the area of this adder can be expressed as:

$$area = (2n - P_1)S_A + (n + m - 1 - P_1)S_M \tag{10}$$

Where, S_A and S_M are the area of a full adder and one bit multiplexer, respectively. So, the area of the fastest adder of this architecture would be equal to:

$$area^* = (2n - \frac{3}{2}\frac{M}{A})S_A + (n + \sqrt{2n\frac{A}{M}} - \frac{3}{2}\frac{M}{A} - 1)S_M \tag{11}$$

3. Design of a New Adder

The proposed innovation for doing addition is that instead of using two separate adders in CSA, one for the case C_{S1}=1 and the other for the case C_{S1}=0 (C_{S1} is the carry propagated from the first partition to the second one), one adder will be used to reduce the area and power dissipation. In this scheme, each of the two additions is done in half of the clock cycle. To accomplish this sharing, some latches are required. This adder is called Carry Select Adder with Sharing (CSAS).

Figure 3 shows the implementation of the idea. In this architecture, (transparent) latches are used to save the result of addition when C_{in} of the 10-bit (MSB) RCA is one. When the clock transitions to low, the 10-bit adder calculates the result of 10-bit addition for the case C_{in} is zero. Therefore, at the end of the clock cycle, the result of the addition of the 10 MSBs is available for both cases of the carry-in signal. Next, based on the actual value of the carry calculated by the LSB adder, a MUX selects the appropriate value, either from the output of the latch or the output of the adder. This value will be the result of the 32-bit addition.

Referring back to Figure 3, notice that the inputs of the adder are partitioned into two parts, one adding the first 22 bits (the LSB adder), while the other adds the last 10 bits (the MSB adder.) Note that the LSB adder has nearly twice as many bits as the MSB adder. Therefore, the MSB adder can calculate the sum of 10 high bits twice when the LSB adder calculates the addition of the lower 22 bits.

This reduces the delay of adding 32 bits by a factor of 1/3 compared to RCA.

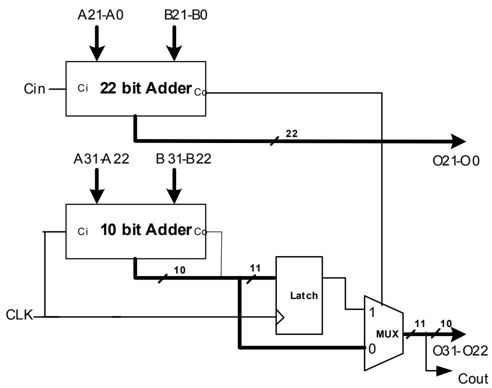

Figure 3: The architecture of CSAS

Like the CSA, this idea may be applied iteratively to achieve faster adders. Assume the bits of an *n*-bit adder are partitioned into *m* groups, where *m*=1 corresponds to the group containing the lease significant bits. P_i denotes the number of bits in group *i*. Figure 4 shows the three stage adder (m =3). Note that P_1=16, P_2=8, and P_3=8. To design the fastest possible adder by this technique, we should find the value of *m* and P_i's ($1\leq i\leq m$) to minimize the delay of the circuit.

Using the assumptions made in the previous section, we first try to balance all critical paths from the inputs of the adders to *Cout*. Note that the first adder performs addition only once, while all other adders do it twice. So,

$$\begin{aligned} D_1 &= P_1.A+(m-1)M \\ D_2 &= 2P_2.A+L+(m-1)M \\ &\cdots \\ D_i &= 2P_i.A+L+(m-i+1)M \\ &\cdots \\ D_n &= 2P_n.A+L+M \end{aligned} \qquad (12)$$

where *L* is the delay of the latch. By setting these delays equal to each other and exploiting the fact that $\Sigma P_i=n$, it can be shown that,

$$P_1 = 2P_2 + \frac{L}{A} \qquad (13)$$

$$P_i = P_2 + \frac{M}{2A}(i-2), \qquad (2 \leq i \leq m) \qquad (14)$$

So,

$$P_2 = \frac{n}{m+1} - \frac{M}{4A}\frac{(m-2)(m-1)}{m+1} - \frac{L}{A(m+1)} \qquad (15)$$

and the delay of the adder is calculated by,

$$delay = \frac{2n}{m+1}A + \frac{M}{2}\frac{(m-1)(m+4)}{(m+1)} - \frac{2L}{(m+1)} + L \qquad (16)$$

The optimum value of *m* to minimize delay is,

$$m^* = \sqrt{4n\frac{A}{M} - 4\frac{L}{M} - 6} - 1 \qquad (17)$$

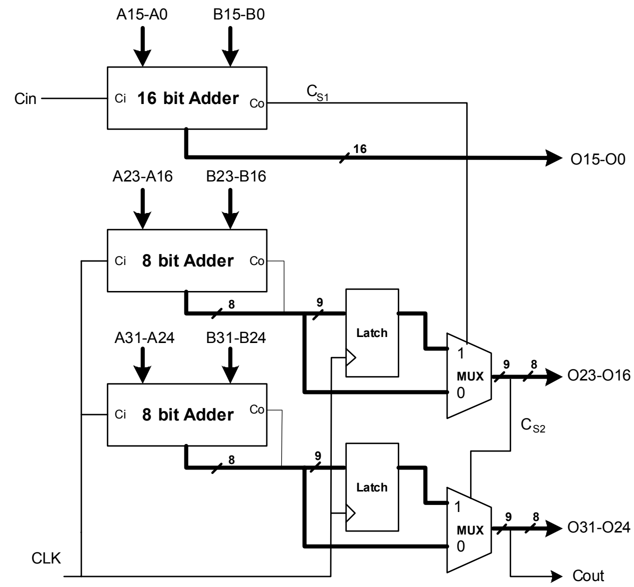

Figure 4: The new implementation of CSAS (m=3)

Since usually *n* is large, $A\approx M$ and $L\approx M$, equation (17) may be approximated as:

$$m^* \approx 2\sqrt{n\frac{A}{M}} - 1 \qquad (18)$$

By replacing (18) into (16), the minimum delay of this type of adder can be estimated by,

$$delay^* \approx 2\sqrt{nAM} + \frac{M}{2} + L \qquad (19)$$

Comparing equations (9) and (19), one can see the fastest adder designed using the new idea is $\sqrt{2}$ times slower than the fastest CSA. However, the area of CSA is larger than the area of the CSAS.
On the other hand, it can be easily verified that the area of CSAS with *m* stages can be written as,

$$area = nS_A + (n+m-1-P_1)(S_M + S_L) \qquad (20)$$

where S_L is the area of a latch. Therefore, the area of the fastest adder of this architecture can be expressed by:

$$area^* = nS_A + (n + 2\sqrt{n\frac{A}{M}} - \frac{5}{2}\frac{M}{A} - \frac{L}{A} - 2)(S_M + S_L) \qquad (21)$$

Table 1 shows the comparison among the delay and area of RCA, the fastest adder of CSA, and the fastest adder of CSAS.

4. Simulation Results

The proposed ideas in Section 3 has been applied to 32 and 64-bit adders. The *Synopsys Design Analyzer* tool was used to synthesize the circuits for minimum power dissipation and area. To estimate the power dissipation of the resulting circuits, *Synopsys DesignPower* was used.

Table 1: Comparison of the area and delay of the fastest CSA and CSAS

	Delay	Area
RCA	nA	nS_A
CSA	$\sqrt{2nAM}+\frac{1}{2}M$	$(2n-\frac{3}{2}\frac{M}{A})S_A+(n+\sqrt{2n\frac{A}{M}}-\frac{3}{2}\frac{M}{A}-1)S_M$
CSAS	$2\sqrt{2nAM}+M+2L$	$nS_A+(n+2\sqrt{n\frac{A}{M}}-\frac{5}{2}\frac{M}{A}-\frac{L}{A}-2)(S_M+S_L)$

For simulations, the inputs of the adders were set to 0 or 1 with equal probability. In these experiments a 0.35um library with the power supply of 3.3V has been used.

Table 2 shows the delay and area of full adders, multiplexers and latches for this library.

Table 3 shows the results for 32-bit adders, while Table 4 shows the delay of the adders calculated using the formulas developed in Section 2 and 3. Here, no flip flop has been inserted at the output of the adders. The numbers in parentheses show the values of P_i's; for example, (9-8-15) means P_3=9, P_2=8, and P_1=15. It is noteworthy that for the library used in this experiment, the optimum value of m for minimizing the delays of CSA, and CSAS are 6, 9, and 5, respectively.

Table 2: Delay and area of full adders, multiplexers, and latches

	Delay (nS)	Area (μm^2)
Full Adder	0.6	70
MUX	0.7	17
Latch	0.8	42

Tables 5 and 6 show the same data as those in Tables 3 and 4, but for a 64-bit adder. The optimum value of m for minimizing the delay for CSA, and CASS are 10 and 9, respectively. For the sake of brevity, only three versions have been shown for each adder. Comparing the specifications of CSA and CSAS with different number of stages shows the proposed idea is superior to CSA. For example, comparing two-stage CSA with four-stage CSAS for 64 bits (Table 5) shows that although the power consumption of the CSAS is 1.5% more than CSA, but its area and delay are respectively 4.4% and 3.4% less than CSA.

5. Conclusion

Based on the idea of sharing two MSB adders used in CSA, a new technique for designing high-performance and low power adders is proposed. In the technique, one adder and some latches are used for adding the MSB's. This innovation results in adders that are faster than RCA, but slower than CSA. On the other hand, the area overhead of CSAS is smaller than that of CSA. Therefore, in designs where some reduction in the delay of the critical path is desired, but large area overhead is intolerable, CSAS can be used to replace RCAs. Alternatively, in designs where smaller area or lower power consumption is desired while some increase in the delay of the longest path in the circuit is allowed, CSAS can be used to replace CLAs.

References

[1] K. Rawwat, T. Darwish, and M. Bayoumi, "A low power carry select adder with reduces area," *Proc. of Midwest Symposium on Circuits and Systems*, pp. 218-221, 2001.

[2] A. Tyagi, "A reduced area scheme for carry-select adders," *IEEE Trans. on Computer*, vol. 42, pp. 1163-1170, 1993

[3] W. Jeong and K. Roy, "Robust high-performance low-power adder," *Proc. of the Asia and South Pacific Design Automation Conference*, pp. 503-506, 2003

[4] Y. Kim and L-S Kim, "64-bit carry-select adder with reduced area," *Electronics Letters*, vol. 37, pp. 614-615, May 2001.

[5] O. Kwon, E. Swartzlander, and K. Nowka, "A fast hybrid carry-lookahead/carry-select adder design," *Proc. of the 11th Great Lakes symposium on VLSI*, pp.149-152, March 2001.

Table 3: The specifications of 32-bit RCA, CSA, and CSAS

Adder	Area μm^2	Delay Ns	Power mW	Area Ratio%	Delay Ratio%	Power Ratio %
RCA	2181	19	73.7	-	-	-
CSA: 2 stages (16-16)	3537	10.7	121.5	162.2	56.3	164.9
CSA: 3 stages (12-10-10)	4036	8.2	137.3	185.1	43.2	186.3
CSA: 4 stages (10-8-7-7)	4293	7.1	144.1	196.8	37.4	195.5
CSA: 5 stages (9-7-6-5-5)	4440	7.0	148	203.6	36.8	200.8
CSAS: 2 stages (10-22)	2769	15.6	99.3	127.0	82.1	134.7
CSAS: 3 stages (8-8-16)	3152	13.8	115.1	144.5	72.6	156.2
CSAS: 4 stages (7-6-6-13)	3373	12.1	123.7	154.7	63.7	167.8
CSAS: 5 stages (6-6-5-5-10)	3456	11.3	135.3	158.5	59.5	183.6

Table 4: Comparison between analytical delay and simulate delay for 32-bit RCA, CSA, and CSAS

Adder	Analytical Delay (ns)	Simulated Delay(ns)	Error %
RCA	19.0	19	0.0
CSA: 2 stages (16-16)	10.3	10.7	-3.7
CSA: 3 stages (12-10-10)	7.6	8.2	-7.3
CSA: 4 stages (10-8-7-7)	6.4	7.1	-9.9
CSA: 5 stages (9-7-6-5-5)	5.8	7.0	-17.1
CSAS: 2 stages (10-22)	15.9	15.6	1.9
CSAS: 3 stages (9-6-17)	12.8	13.8	-7.2
CSAS: 4 stages (9-6-3-14)	11.1	12.1	-8.3
CSAS: 5 stages (9-6-3-1-13)	10.1	11.3	-10.6

Table 5: The specifications of 64-bit RCA, CSA, and CSAS

Adder	Area μm^2	Delay Ns	Power mW	Area Ratio%	Delay Ratio%	Power Ratio %
RCA	4447	37.42	150	-	-	-
CSA: 2 stages (32-32)	7069	19.81	244.6	159.0	52.9	163.1
CSA: 3 stages (22-21-21)	7985	14.39	275.7	179.6	38.5	183.8
CSA: 4 stages (10-8-7-7)	8484	11.84	291.9	190.8	31.6	194.6
CSAS: 2 stages (21-43)	5677	26.37	203.6	127.7	70.5	135.7
CSAS: 3 stages (17-14-33)	6368	21.35	232.3	143.2	57.1	154.9
CSAS: 4 stages (15-12-10-27)	6759	18.8	248.3	152.0	50.2	165.5

Table 6: Comparison between analytical delay and simulate delay for 64-bit RCA, CSA, and CSAS

Adder	Analytical Delay(ns)	Simulated Delay(ns)	Error %
RCA	38.4	37.42	2.6
CSA: 2 stages (32-32)	19.72	19.81	-0.5
CSA: 3 stages (22-21-21)	13.73	14.39	-4.6
CSA: 4 stages (10-8-7-7)	10.91	11.84	-7.9
CSAS: 2 stages (21-43)	30.8	27.76	11.0
CSAS: 3 stages (17-14-33)	24.0	23.46	2.3
CSAS: 4 stages (15-12-10-27)	20.1	21.45	-6.3