

A LOW POWER CARRY SELECT ADDER WITH REDUCED AREA

Youngjoon Kim and Lee-Sup Kim

Department of EECS, KAIST,
373-1 Kusong-dong, Yusong-gu, Taejeon,
Korea

ABSTRACT

A carry-select adder can be implemented by using single ripple carry adder and an add-one circuit [1] instead of using dual ripple-carry adders. This paper proposes a new add-one circuit using the first zero finding circuit and multiplexers to reduce the area and power with no speed penalty. For bit length $n = 64$, this new carry-select adder requires 38 percent fewer transistors than the dual ripple-carry carry-select adder and 29 percent fewer transistors than Chang's carry-select adder using single ripple carry adder [1]. This new 64b adder has 3.45ns delay time at 2.5 V power supply using a 0.25um CMOS technology.

1. INTRODUCTION

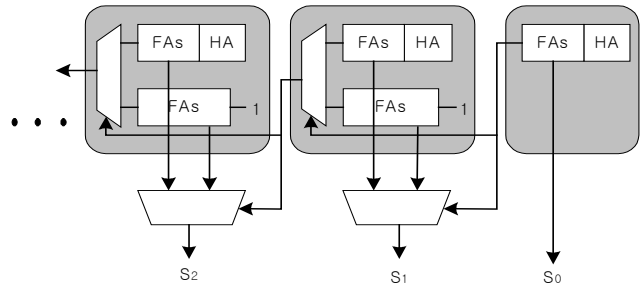
Due to the rapidly growing mobile industry, not only faster arithmetic units but also smaller and lower power arithmetic units are demanded. However, it has been difficult to do well both in speed and in area. In general, ripple-carry adder (RCA) provides a compact design but suffers from a long delay time. Carry look-ahead adder (CLA) gives a fast design but has a large area. Carry-select adder (CSA) is intermediate in regard to speed and area. Therefore, CSA is suitable in many applications that consider both speed and area. CSA is also used with CLA to improve the speed [4].

This paper proposes a new architecture to reduce area and power of CSA. Reduced area schemes are introduced in Section 2. In Section 3, proposed CSA architecture is discussed. In Section 4, SPICE simulated results and comparisons with other conventional CSAs are discussed. Finally, this paper ends with conclusion.

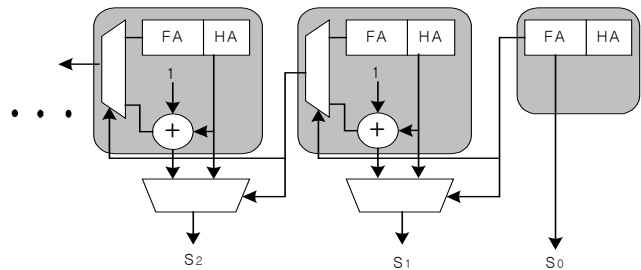
2. REDUCED AREA SCHEMES

2.1 An add-one circuit to replace one of RCAs

Adders with very large words sizes are constructed hierarchically by combining smaller "block" adders [3]. As shown in Fig. 1(a), the shaded parts are the blocks in the conventional carry-select adder consists two ripple carry adders, one for $C_{in} = 0$ and the other for $C_{in} = 1$. If the results for $C_{in} = 0$ is known, the result for $C_{in} = 1$ can be found by adding one to the result for $C_{in} = 0$. Thus, an add-one circuit can replace the ripple-carry adder for $C_{in} = 1$ in a block as shown in Fig. 1(b). With an efficient design of an add-one circuit, the area of CSA can be reduced. The add-one circuit architecture is discussed in the next section.



(a) Conventional CSA using dual RCAs.



(b) CSA with add-one circuit replacing the RCA for $C=1$.

Figure 1. A conventional and a modified carry-select adders.

2.2 Complement scheme for performing an add-one circuit

To design an efficient add-one circuit, the complement scheme by Chang is used [1]. This scheme is explained as follows. Assume $(S_{n-1}^0, S_{n-2}^0, \dots, S_0^0)$ and $(S_{n-1}^1, S_{n-2}^1, \dots, S_0^1)$ are the results of two CRAs with $C_{in} = 0$ and $C_{in} = 1$ respectively. Then, S_0^0 is always equal to the complement of S_0^1 and S_k^1 is equal to S_k^0 , if $\prod_{i=0}^{k-1} S_i^0 = 0$; otherwise, $S_k^1 = \text{Sbar}_k^0$ for $1 < k < n-1$, where Sbar is the complement of S [1]. In other words, adding one is just inverting each S^0 bit starting from the least significant bit until the first zero is found. Two examples for the scheme are shown in Fig. 2.

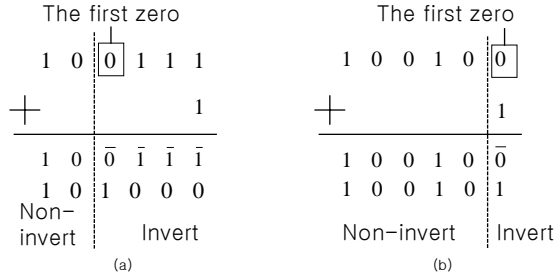


Figure 2. Examples for the complement scheme. The first zero decides whether the bit is needed to be inverted or not.

3. PROPOSED CSA ARCHITECTURE

3.1 Inverter elimination in carry path of RCA

CSA is composed of many small RCA blocks. Thus, reducing the delay of RCAs is important for designing a CSA. In order to optimize the RCA delay, all RCAs in this paper use the mirror adders [4] and the inverter elimination scheme in carry path [3]. The inverter elimination scheme uses two properties of the mirror adder. The first property is inverting all inputs on the full adder results in inverted values for all outputs. The second one is the mirror adder generates the complement of carryout first and inverts it to generate the carryout. Therefore, by putting even and odd cells as shown in Fig. 3, the number of the inverting stages in the carry path is reduced [1]. This reduces $N \times$ inverter delay in the carry pass where N is the block size. There is no transistor penalty for this scheme. In fact, one less transistor is used than the conventional full adder with 28 transistors.

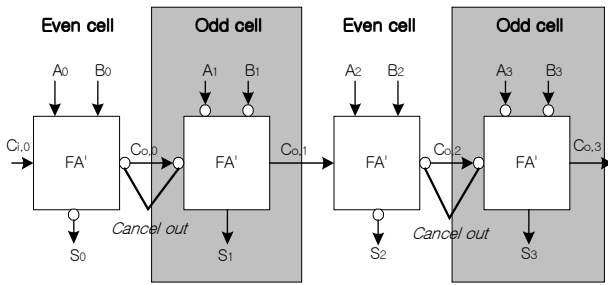


Figure 3. Inverter elimination in carry path. FA' stands for the mirror full adder without the inverters in carry and sum paths. The FA' contains 24 transistors.

3.2 Proposed add-one circuit

As mentioned in the previous section, the complement scheme is used for performing an add-one circuit. Chang used half adders, inverters, and multiplexers to perform the add-one circuit [1]. Instead, a multiplexer-based add-one circuit is proposed as shown in Fig. 4. According to the previous complement scheme, a S_k^1 is either the S_k^0 or the complement of S_k^0 where S_k^0 represents a sum of k th bit for $C=0$. Since the full adder generates both sum and the complement of sum, no extra inverter

is needed to get the complement. To generate the add-one circuit, a multiplexer is needed for each bit to choose either S_k or the complement of S_k . The control signal of the multiplexer is from the first zero finding circuit. The first zero finding circuit is NMOS and PMOS chains as shown in the top middle of Fig. 4. This circuit generates 0 at the k th node if no zero is founded until k th bit from the least significant bit; otherwise, it generates 1. If the control signal is 0, the multiplexer chooses S_k ; otherwise, it chooses the inverted S_k . The least significant bit does not need a multiplexer since S_0^1 is always the opposite of S_0^0 . This saves a few transistors for each block.

The carry out for a block can be chosen between the carry out for the RCA or the carry out for the add-one circuit. The carry out for the add-one circuit is one if and only if all sums from the RCA are equal to one. When all sums are equal to one, the first finding circuit generates zero at the final node. All other cases it generates one. Therefore, the inverted final node can be used as the carryout for $C=1$.

Finally, the multiplexers is placed in the bottom to choose between the results for $C=0$ and the results for $C=1$.

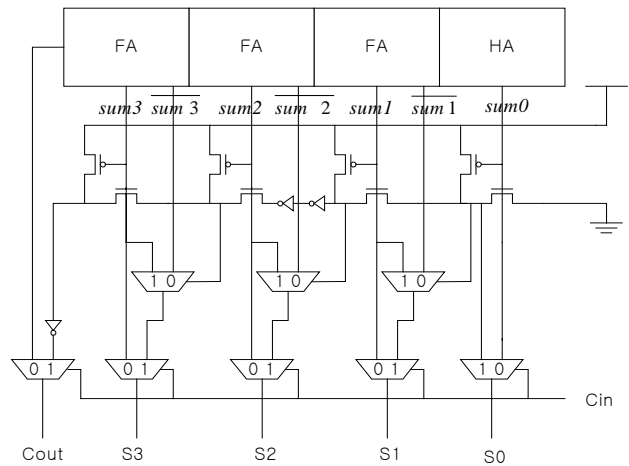


Figure 4. Proposed multiplexer-based add-one circuit.

One multiplexer and a NAND can replace the two multiplexers in Fig. 4. The Fig. 5 shows that two circuits are equal. Then, $2 \times (N-1)$ can be reduced where N is the number of bits in a block.

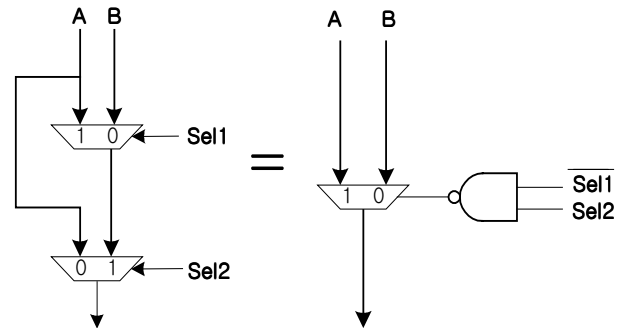


Figure 5. Replacing two multiplexers by one multiplexer with a NAND.

However, to use this module, the control signal 1 should be inverted as shown in Fig. 5. Therefore, by switching the VDD and the GND on the first zero finding circuit, the inverted control 1 signals are generated. This scheme also eliminates an inverter delay for the carry out.

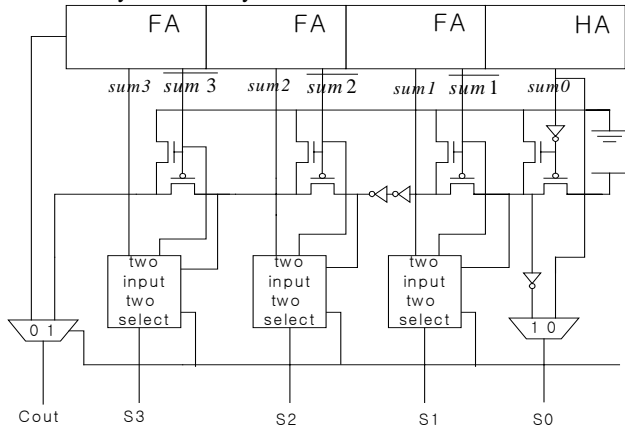


Figure 6. Final proposed circuit.

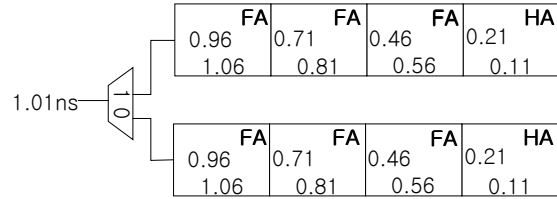
3.3 Designing square-root CSA

In order to optimize the worst-case delay, the square-root scheme is used [1]. The square-root scheme is matching the block size according to the arrival time for the carry-in signal. To determine the each block size, the delay for each basic gate is needed. Table 1 shows the SPICE simulated delay time for each basic gate on our 0.25um CMOS technology.

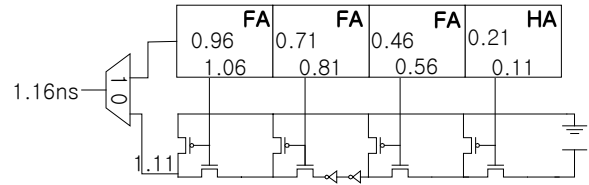
Table 1. The simulated actual delay time and normalized delay of basic gates.

Basic gates	Delay
Inverter	0.08ns
NAND	0.13ns
Multiplexer(sel)	0.11ns
Multiplexer(thru)	0.05ns
XOR	0.11ns
Sum(Half Adder)	0.11ns
Cout(Half Adder)	0.21ns
Sum(Full Adder)	0.35ns
Cout(Full Adder)	0.25ns

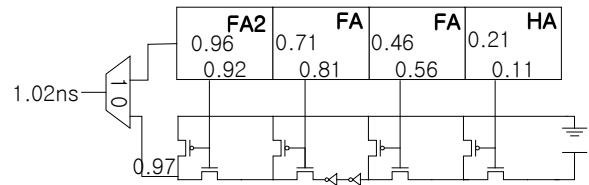
Base on the Table 1, the block delay can be estimated as shown in Fig. 7. Since the sum of the most significant bit for C=0 is used to get the carryout of a block, the proposed CSA delay is longer than the CSA using dual RCAs. Therefore, by replacing the last FA by two-level XORs to get the sum faster, the delay time can be reduced. As shown in Fig. 7 (a) and 7 (b), the estimated delay time for a block now becomes approximately same as the conventional CSA. The SPICE simulated results shown in Fig. 8 verified the previous statement. Using the estimated delay times, block sizes can be determined. Since the conventional and the proposed block delay are similar, the conventional CSA block size can be adapted in our proposed design.



(a) Conventional CSA using dual RCA.

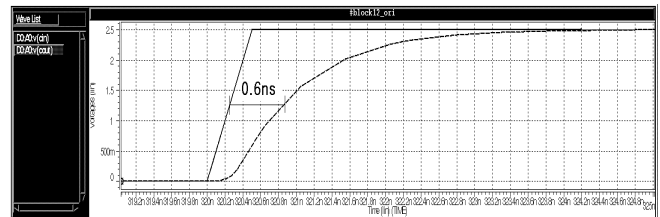


(b) CSA using the mirror FA for all bits.

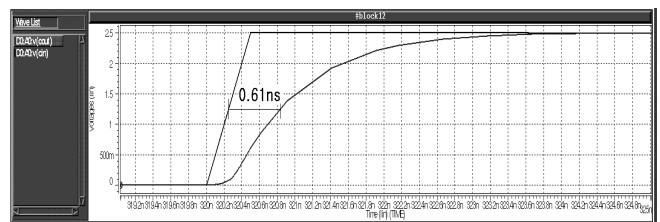


(c) Proposed CSA using FA2 with two-level XORs and two-level NANDs at the most significant bits. FA blocks are the mirror FA.

Figure 7. Estimated delay times for various 4 bits' adder block.



(a) The 12 bits conventional CSA block.



(b) The 12 bit proposed CSA block.

Figure 8. The 12 bits block delays for the conventional and the proposed carry-select adder.

Table 2,3,4 shows the number of transistors, block sizes, and the estimated delay time for various CSA types. The proposed adder has 38 percent fewer transistors than the conventional adder and

29 percent fewer transistors than the Chang's CSA. The estimated delay time for the proposed adder is very close to the conventional CSA.

Table 2. Original 64 bits square root carry-select adder with 10 blocks.

Block	Total	10	9	8	7	6	5	4	3	2	1
RCA n=	64	12	11	9	8	7	6	4	3	2	2
TR#	3660	708	650	530	470	410	348	228	170	108	38
Delay	3.11	3.11	2.76	2.26	2.01	1.76	1.51	1.01	0.76	0.51	0.46

Table 3. Chang's carry-select adder.

Block	Total	9	8	7	6	5	4	3	2	1
RCA n=	64	12	11	9	8	7	6	4	3	4
TR#	3166	616	566	462	408	358	304	200	150	102

Table 4. Proposed carry select adder.

Block	Total	10	9	8	7	6	5	4	3	2	1
RCA n=	64	12	11	9	8	7	6	4	3	2	2
TR#	2268	486	442	358	318	282	226	150	96	66	38
Delay	3.17	3.17	2.84	2.33	2.07	1.81	1.55	1.04	0.77	0.52	0.46

4. SIMULATED RESULTS AND COMPARISONS

The SPICE simulated delays for the conventional CSA and the proposed CSA are shown in Fig. 9. The results show that the proposed adder is faster than the conventional adder. The reason for is shown in Fig. 10. The worst-case delay happens when the carry propagated from the LSB to MSB. In that case, the inputs for each adder are either $(a_n=1, b_n=0)$ or $(a_n=0, b_n=1)$ besides the LSB where both a_0 and b_0 should be 1. As shown in Fig.10, the Cout for Cin=1 in a block propagates and cause a long delay time for Cout. Thus, if the arrival time of Cin is faster than the time for generating Cout for Cin=1, the total delay time become slower than the proposed adder where no carry propagation occurred. The arrival times for the Cin and the Cout for Cin=1 should be adjusted to be the same. Then, the original CSA is faster than the proposed as estimated previously. However, it would be quite complicated since FA delay is not equal to a MUX delay. Therefore, no carry RCA propagation in the critical path is preferred for a CSA design.

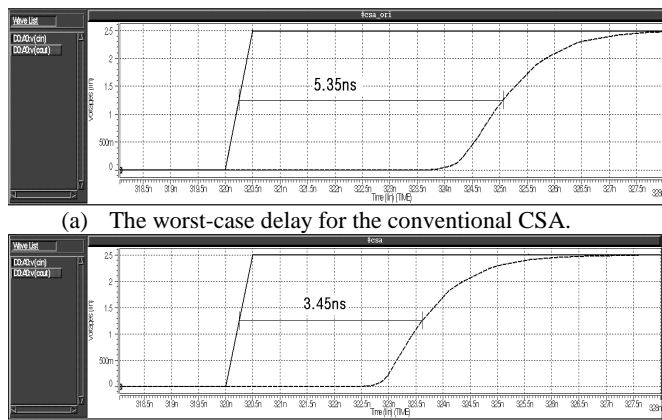
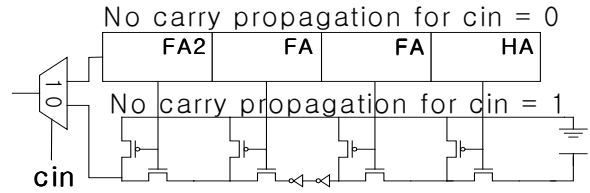
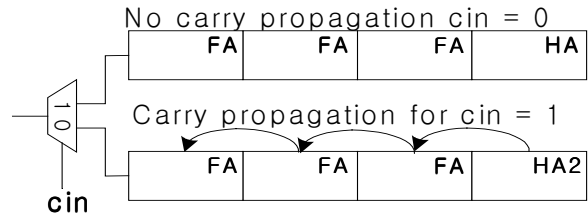


Figure 9. Critical path SPICE waveforms for the conventional and the proposed CSAs



(a) The proposed CSA. No carry propagation at the critical path in the block.



(b) The conventional CSA. Carry propagation of RCA for cin = 1 at the critical path

Figure 10. Showing carry propagation of a block in case of critical path at the worst case

5. CONCLUSIONS

Replacing the RCA for C=1 by the proposed add-one circuit with the complement scheme reduces the number of transistors of the CSA with ignorable speed penalty. Compared to the conventional and Chang's CSA, the proposed adder required 38% and 29% fewer transistors, respectively. Fewer transistors results less area and less power. The power consumption of proposed CSA is estimated to be only 75% of the conventional CSA from the SPICE simulation. The proposed 64b adder has 3.45ns delay time at 2.5 V power supply using a 0.25um CMOS technology.

6. ACKNOWLEDGMENT

This work was supported by KOSEF through the MICROS at KAIST, Korea.

7. REFERENCES

- [1] Chang, T. Y. and Hsiao, M. J., "Carry-select adder using single ripple-carry adder". *Electronics Letters*, vol. 34, No. 22, Oct 1998, pages 2101-2103.
- [2] Rabaey, J. M., *Digital Integrated Circuits: A Design perspective*. New Jersey, Prentice-Hall, 1996.
- [3] N. Weste and K. Eshragian, *Principles of CMOS VLSI Designs: A System Perspective*, 2nd ed., Addison-Wesley, 1985-1993.
- [4] Morinaka, H., Makino, H., Nakase, Y. et. al, "A 64 bit Carry Look-ahead CMOS adder using Modified Carry Select". *Custom Integrated Circuit Conference*, 1995, pages 585-588.