

High-Performance Carry Select Adder Using Fast All-one Finding Logic

Yan Sun, Xin Zhang, Xi Jin

Institute of Microelectronics, University of Science and Technology of China

Hefei, AH 230026 P.R.China

E-mail: {ysun, xinz}@mail.ustc.edu.cn, jinxi@ustc.edu.cn

Abstract

A carry-select adder(CSA) can be implemented by using single ripple carry adder and an add-one circuit instead of using dual ripple-carry adders to reduce the area and power but with speed penalty. This paper proposes a new add-one circuits using the fast all-one finding circuit and low-delay multiplexers to reduce the area and accelerate the speed of CSA, and no restrictions are imposed on the design of the adder blocks. For bit length $n = 64$, this new carry-select adders requires approximate 38 percent fewer transistors and 16 percent shorter delay than the original dual ripple-carry carry-select adder.

1. Introduction

Due to the rapidly growing mobile industry, not only faster units but also smaller area and less power become major concerns for designing digital circuits. Adders are critical components of the ALU's (Arithmetic Logic Unit) or DSP (Digital Signal Processing) chips[1]. Among various adders, the carry-select adder (CSA) is intermediate regarding speed and area and widely used in mobile applications[2]. CSAs with very large sizes can be constructed hierarchically by combining smaller 'block' adders [3]. Fig. 1 shows the conventional CSA which consists of two ripple carry adders (RCAs) in each block(except block1), one for C_{in} (from lower block) = 0 and the other for C_{in} (from lower block) = 1. Instead of using dual RCAs, Chang proposed a 29.2% reduced area CSA with 5.9% speed penalty by replacing one RCA to an add-one circuit [4]. And many work has been done to reduce large area of CSA with speed penalty or reduce smaller area with negligible speed penalty[5-8]. In this paper we propose a further reduced area CSA with shorter delay.

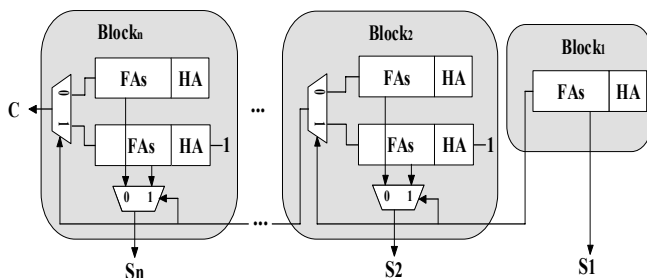


Fig.1. Conventional CSA using dual RCAs with n blocks

2. An add-one circuit to replace one of RCAs

The blocks in the conventional carry-select adder consists of 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 as S^0 , the result for $C_{in}=1(S^1)$ can be obtained by adding one to S^0 . Thus, an add-one circuit can replace the ripple-carry adder for $C_{in}=1$ to reduce the area in a block. To design an efficient add-one circuit, the first zero finding circuit is used in implement scheme[4]. Adding one to the result for $C_{in}=0 (S^0)$, if the S^0_k is the first zero count from the least significant bit, the S^1 is just inverting each bit of S^0 starting from the least significant bit until the S^0_k bit(included), and other bit(s) remain the same. The 4-bit add-one circuit architecture used by Chang is showed in the Fig.2.

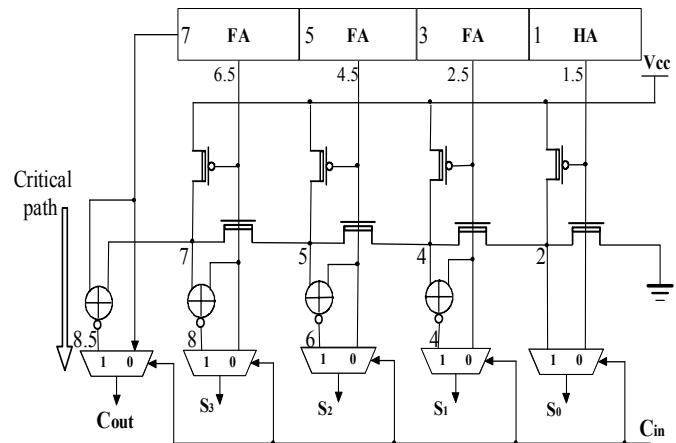


Fig. 2. Carry-select adder using add-one circuit

The full adder(FA) cell consists of a two-level NAND gate for carry output and two-level two-input exclusive-or gates with the critical delays. And the

delay in the unit of the two input NAND gate was illustrated in Fig.2. The carry-chain is the critical path in the CSA, so the critical path increase 1.5 unit in every block compared with the original RSA structure.

As shown in Fig.3(a), the implementation of the multiplexer with control signal C_{in} requires three two-input NAND gates and an inverter for the complementary signal of C_{in} [4]. Thus, the delay from C_{in} to C_{out} in the unit of the two input NAND gate is 2.5.

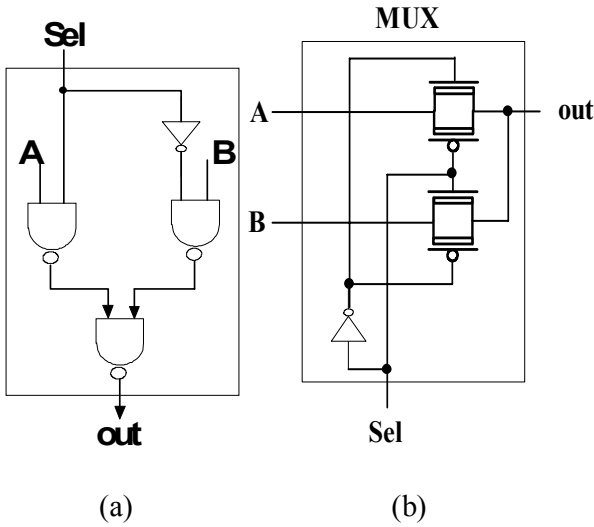


Fig. 3. the implementation of the multiplexer

3. Proposed CSA architecture

As mentioned in the previous section, the carry-chain between blocks is the critical path in the CSA, it includes the delay of multiplexers and the wire delay. When another structure of multiplexer showed in Fig.3(b) was used in proposed CAS, the delay time through the multiplexer becomes very small. Furthermore, in every block, compared with the original CSA structure, the circuit used to determine the C_{out} when $C_{in}=1$ increased the delay of the critical path. Instead, a new add-one circuit is proposed to eliminate this delay and further reduce the area as shown in Fig. 4.

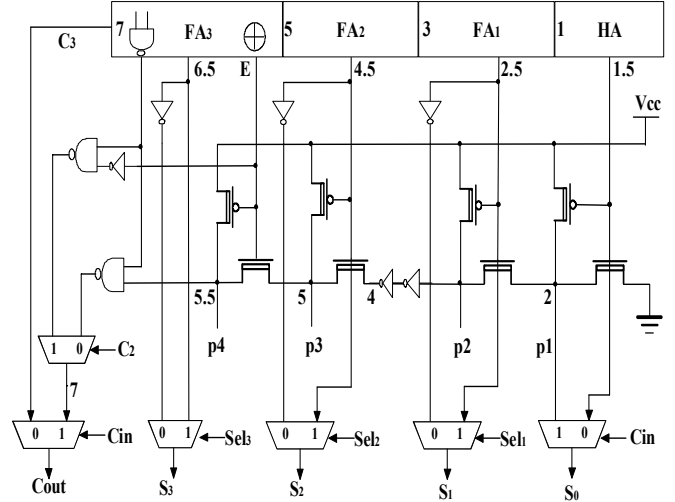


Fig. 4. Proposed CSA with fast all-one finding circuit

According to the previous complement scheme, the sum result S_k is either the S_k^0 or the S_k^1 (S_k^0 or inverted S_k^0), thus the S_k can be chosen from S_k^0 and inverted S_k^0 through a multiplexer shown in Fig.4. The select signals were shown in following equations, and the P signals was generated from the first zero finding circuit to decide whether the S_k^1 is S_k^0 or inverted S_k^0 .

$$\begin{cases} Sel_1 = C_{in} \cdot \overline{P_1} \\ Sel_2 = C_{in} \cdot \overline{P_2} \\ Sel_3 = C_{in} \cdot \overline{P_3} \end{cases}$$

The carry out of a block can be chosen between the carry out of the RCA and the carry out of the add-one circuit. So the fast all-one finding scheme depends on two points in a block, one is that the two carry out will be different if and only if all sums from the RCA and the C_{in} are equal to one, another is that if all sum bits from the RCA in a block are equal to one, the inputs of every FA or HA in the block will be different. So we can use an exclusive-or in the highest bit to determine whether all sums from the RCA are equal to one. in Fig.4, if P_4 is 0, the four-bit sum from the RCA and C_2 (the carry of FA_2) should be '1111' and 0 respectively. The delay of P_4 is reduced without extra transistors because the value of exclusive-or is contained in the FA_3 . Furthermore, if $P_4 = 0$ or $a_3 \cdot b_3 = 1$, C_{out} should be 1. So, the output of the

$$\text{carry } C_{\text{out}} = C_3 + C_{\text{in}} \cdot (\overline{P_3 + a_3 \cdot b_3} + (a_3 \oplus b_3) \cdot C_2)$$

. A SPICE simulation of the 4bit carry-select adder of the proposed scheme was performed to verify the proposed design, and the delay of the Cout is on longer than the original dual ripple-carry adders.

The implementation of the 64 bit dual ripple-carry adder with 10 blocks [4] and proposed carry-select adder with 9 blocks is listed in Table 1 and Table 2 respectively. The first (second) row shows the block number (bit length of the CRA in each block), while the third (fourth) row lists the transistor count (delay time in units of delay due to the two-input NAND gate) in each block.

Table 1. Original 64-bit carry-select adder with 10 blocks

Block	1	2	3	4	5	6
CRA n=	2	2	3	4	6	7
Tr #	42	130	204	278	426	500
Delay	3	5.5	8	10.5	13	15.5
Block	7	8	9	10	Total	
CRA n=	8	9	11	12	64	
Tr #	574	648	796	870	4468	
Delay	18	20.5	23	25.5	25.5	

Table 2. Proposed 64-bit carry-select adder with 9 blocks

Block	1	2	3	4	5
CRA n=	4	4	5	6	7
Tr #	102	172	220	264	302
Delay	7	7.5	9.5	11.5	13.5
Block	6	7	8	9	Total
CRA n=	8	9	10	11	64
Tr #	356	404	448	496	2764
Delay	15.5	17.5	19.5	21.5	21.5

In the proposed scheme, the transistor count needed in each block (except block 1 which is a carry-ripple adder) with adder bit length n is $44n - 8 + 4[(n - 1)/2]$. As listed in Table 1 and Table 2, the transistor count is reduced to 61.86% in the proposed scheme, the delay is reduced to 84.31%. The 64-bit proposed adder shown in Table 2 has 1.98ns delay time at 1.2 V power supply using SMIC 0.13um CMOS technology.

4. Conclusions

In this paper we have presented a carry-select adder that replaces the RCA for C= 1 by the proposed add-one circuit with the fast all-one finding scheme and new multiplexers to reduces the number of transistors of the CSA with accelerated speed. No restrictions are imposed on the design of the adder blocks. The transistor count can be reduced by 38.14% and the delay can be reduced by 15.69% for n = 64. Compared to Chang's CSA, the proposed adder required 34.25%([4204-2764]/4204) fewer transistors when with no speed penalty.

5. References

- [1] J. Woopyo, and R. Kaushik, "Robust High-Performance Low-Power Carry Select Adder", *Design Automation Conference*, pp. 503 - 506, Jan. 2003.
- [2] K. Rawat, T. Darwish, and M. Bayoumi, "A low power and reduced area carry select adder", *The 45th Midwest Symposium on Circuits and Systems.*, vol.1, pp. 467-470, 2002.
- [3] N. Weste and K. Eshragian, "Principle Of CMOS VLSI designs: a system perspective", *2nd ed.*, Addison-Wesley, 1993.
- [4] T. Y. Chang and M. J. Hsiao, "Carry-select adder using ripple-carry adder", *Electron. Lett.*, vol. 34, pp. 2101-2103, 1998.
- [5] B. Amelifard, F. Fallah and M. Pedram, "Closing the gap between carry select adder and ripple carry adder: a new class of low-power high-performance adders", *Sixth International Symposium on Quality of Electronic Design*, pp. 148 - 152, 2005.
- [6] Y. He, C.H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for low power applications", *IEEE International Symposium on Circuits and Systems*, Vol. 4, pp. 4082 - 4085, 2005.
- [7] Y. Chen, H. Li. Roy and K. C. Kok, "Cascaded carry-select adder: a new structure for low-power CSA design", *Proceedings of the 2005 International Symposium on Low Power Electronics and Design*, pp. 115 – 118, 2005.
- [8] A. Neve, H. Schettler, and T. Ludwig, "Power-delay product minimization in high-performance 64-bit carry-select adders", *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 12 , Iss. 3, pp. 235 – 244, March 2004.