An Area-Efficient Carry Select Adder Design by Sharing the Common Boolean Logic Term

I-Chyn Wey, Cheng-Chen Ho, Yi-Sheng Lin, and Chien-Chang Peng

Abstract—**In this paper, we proposed an area-efficient carry select adder by sharing the common Boolean logic term. After logic simplification and sharing partial circuit, we only need one XOR gate and one inverter gate in each summation operation as well as one AND gate and one inverter gate in each carry-out operation. Through the multiplexer, we can select the correct output result according to the logic state of carry-in signal. In this way, the transistor count in a 32-bit carry select adder can be greatly reduced from 1947 to 960. Moreover, the power consumption can be reduced from 1.26mw to 0.37mw as well as power delay product reduced from 2.14mw*ns to 1.28mw*ns.**

Index Terms—**Carry Select Adder, Area-Efficient, Hardware-Sharing, Boolean Logic**

I. Introduction

The carry-ripple adder is composed of many cascaded single-bit full-adders. The circuit architecture is simple and area-efficient. However, the computation speed is slow because each full-adder can only start operation till the previous carry-out signal is ready. In the carry select adder, N bits adder is divided into M parts. Each part of adder is composed two carry ripple adders with cin_0 and cin_1, respectively. Through the multiplexer, we can select the correct output result according to the logic state of carry-in signal. The carry-select adder can compute faster because the current adder stage does not need to wait the previous stage's carry-out signal. The summation result is ready before the carry-in signal arrives; therefore, we can get the correct computation result by only waiting for one multiplexer delay in each single bit adder. In the carry select adder, the carry propagation delay can be reduced by M times as compared with the carry ripple adder. However, the duplicated adder in the carry select adder results in larger area and power consumption.

Manuscript received Dec. 25, 2011. This work was supported in part by NSC 99-2221-E-182-062-MY2 and was supported by National Chip Implementation Center, Taiwan, for the EDA tool support.

I-Chyn Wey is with Graduate Institute of Electrical Engineering, Electrical Engineering Department, Green Technology Research Center, and Healthy Aging Research Center, Chang-Gung University, Taiwan. (e-mail: icwey@mail.cgu.edu.tw).

Cheng-Chen Ho is with Electrical Engineering Department, Chang-Gung University, Taiwan.

Yi-Sheng Lin is with Electrical Engineering Department, Chang-Gung University, Taiwan.

Chien-Chang Peng is with Graduate Institute of Electrical Engineering, Chang-Gung University, Taiwan.

In this paper, we proposed an area-efficient carry select adder by sharing the common Boolean logic term. After Boolean simplification, we can remove the duplicated adder cells in the conventional carry select adder. Alternatively, we generate duplicate carry-out and sum signal in each single bit adder cell. By utilizing the multiplexer to select the correct output according to its previous carry-out signal, we can still preserve the original characteristics of the parallel architecture in the conventional carry select adder. In this way, the circuit area and transistor count can be greatly reduced and power delay product of the adder circuit can be also greatly lowered.

II. Area-Efficient Carry Select Adder

The carry ripple adder is constructed by cascading each single-bit full-adder [1]. In the carry ripple adder, each full-adder starts its computation till previous carry-out signal is ready. Therefore, the critical path delay in a carry ripple adder is determined by its carry-out propagation path. For an N-bit full-adder as illustrated in Fig. 1, the critical path is N-bit carry propagation path in the full-adders. As the bit number N increases, the delay time of carry ripple adder will increase accordingly in a linear way.

In order to improve the shortcoming of carry ripple adder to remove the linear dependency between computation delay time and input word length, carry select adder is presented [2]. The carry select adder divides the carry ripple adder into M parts, while each part consists of a duplicated (N/M)-bit carry ripple adder pair, as illustrated in Fig. 2 as M=16 and N=4. This duplicated carry ripple adder pair is to anticipate both possible carry input values, where one carry ripple adder is calculated as carry input value is logic "0" and another carry ripple adder is calculated as carry input value is logic "1". When the actual carry input is ready, either the result of carry "0" path or the result of carry "1" path is selected by the multiplexer according to its carry input value. An example of 5-bit carry select adder is illustrated in Fig. 3. To anticipate both possible carry input values in advance, the start of each M part carry ripple adder pair no longer need to wait for the coming of previous carry input. As a result, each M part carry ripple adder pair in the carry select adder can compute in

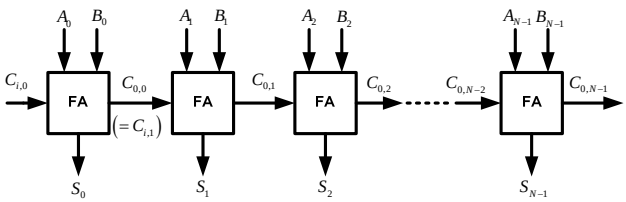

Fig. 1 The N-bit carry ripple adder constructed by N set single bit full-adder

ISBN: 978-988-19251-9-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

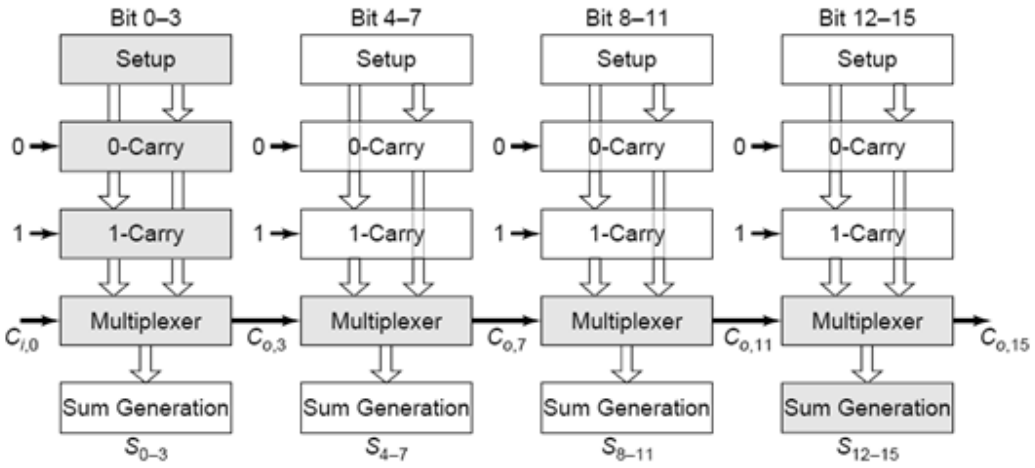

Fig. 2 The 16-bit carry select adder is divided the carry ripple adder into 4 parts, while each part consists of a duplicated 4-bit carry ripple adder pair.

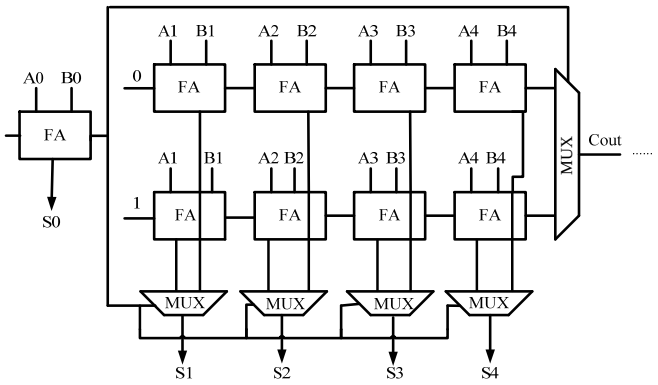

Fig. 3 5-bit carry select adder [1], [2].

Cin	A	B	S0	C0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fig. 4 The truth table of single-bit full-adder, where the upper half part is the case of Cin=0 and the lower half part is the case of Cin=1.

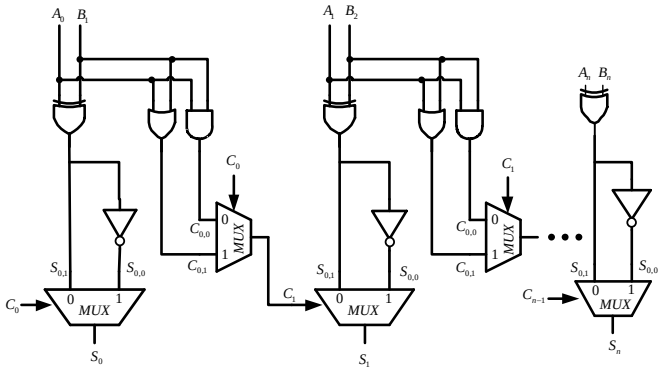

Fig. 5 The proposed area-efficient carry select adder is constructed by sharing the common Boolean logic term in summation generation.

parallel. In this way, the critical path of N bit adder can be greatly reduced. In the conventional N-bit carry ripple adder design, the critical path is N-bit carry propagation path plus one summation generation stage. Alternatively, the critical path is (N/M)-bit carry propagation path plus M stage multiplexer with one summation generation stage in the N-bit carry select adder. Since M is much smaller than N and delay in the multiplexer is smaller than that in the full adder, the computation delay in the carry select adder is much shorter than that in the carry ripple adder. However, implementing the adder with duplicated carry generation circuit costs almost twice hardware and twice power consumption as compared with the carry ripple adder. Therefore, in this paper, we proposed an area-efficient carry select adder by sharing the common Boolean logic term to remove the duplicated adder cells in the conventional carry select adder. In this way, we can save many transistor counts and achieve a lower PDP.

Through analyzing the truth table of a single-bit full-adder, we can find out that the output of summation signal as carry-in signal is logic “0” is the inverse signal of itself as carry-in signal is logic “1”. As illustrated as two red circles in the truth table of Fig. 4, S_0 is “0110” as C_{in} is logic “0” and S_0 is “1001” as C_{in} is logic “1”. By sharing the common Boolean logic term in summation generation, we illustrate our proposed area-efficient carry select adder design in Fig. 5. To share the common Boolean logic term, we only need to implement one XOR gate with one INV gate to generate the summation signal pair. As the carry-in signal is ready, we can select the correct summation output according to the logic state of carry-in signal. As for the carry propagation path, we construct one OR gate and one AND gate to anticipate possible carry input values in advance. Once the carry-in signal is ready, we can select the correct carry-out output according to the logic state of carry-in signal. In this way, we can keep both the summation generation circuit of XOR gate and INV gate and the carry-out generation circuit of OR gate and AND gate in parallel. Since we still retain part of parallel architecture of conventional carry select adder, we can still maintain some competitiveness in speed. On the other hand, we needn’t to prepare the duplicated adder cells in the conventional carry select adder, which can greatly reduce the transistor count and lower the power consumption.

In the proposed area-efficient carry select adder, we

ISBN: 978-988-19251-9-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

trade-off transistor count with speed to achieve a lower power-delay product. In the N-bit carry ripple adder, the delay time can be expressed as:

$$T_{CRA} = (N\text{-}1)T_{carry} + T_{sum} \quad (1)$$

In the N-bit carry select adder, the delay time is:

$$T_{CSA} = T_{setup} + (N/M)T_{carry} + MT_{mux} + T_{sum} \quad (2)$$

In our proposed N-bit area-efficient carry select adder, the delay time is:

$$T_{new} = T_{setup} + (N\text{-}1)T_{mux} + T_{sum} \quad (3)$$

As compared with the conventional carry select adder, our speed is a little slower since the parallel path in our design is shorter. However, we can achieve lower area, lower power consumption, and lower PDP. As compared with the carry ripple adder, our speed can be faster because some of the parallel architecture in the conventional carry select adder is retained. The delay time in our proposed adder design is also proportional to the bit number N; however, the delay time of multiplexer is shorter than that of full adder. Consequently, our area-efficient adder can perform with nearly the same transistor count, nearly the same power consumption, but with faster speed and lower PDP as compared with the carry ripple adder.

III. Simulation Comparison Results

We compare the circuit performance with three different architectures, 32-bit carry ripple adder, 32-bit carry select adder, and 32-bit area-efficient carry select adder that is proposed in this paper. As for the transistor count, the transistor count of our proposed area-efficient carry select adder could be reduced to be very close to that of carry ripple adder; however, the transistor count in the conventional carry select adder is nearly double as compared with the proposed design. This result shows that sharing common Boolean logic term could indeed achieve a superior performance in aspect of transistor count.

The area-efficient carry select adder can also achieve an outstanding performance in power consumption. Power consumption can be greatly saved in our proposed area-efficient carry select adder because we only need one XOR gate and one INV gate in each summation operation as well as one AND gate and one OR gate in each carry-out operation after logic simplification and sharing partial circuit. Because of hardware sharing, we can also significantly reduce the occurring chance of glitch. Besides, the improvement of power consumption can be more obvious as the input bit number increases. We simulated the power consumption in the proposed area-efficient adder and the conventional carry select adder with 4, 8, 16, and 32-bit respectively in tsmc 0.18um CMOS technology. Fig. 6 shows the simulation results of the proposed area-efficient carry select adder and the conventional carry select adder. The power consumption difference between these two designs is small in the case of 4-bit input word length. Since the conventional carry select adder consists of the duplicated adder cells to prepare both the possible output values for the corresponding carry input values in advance. It not only needs larger hardware area, but also generates more glitch signals because of propagation path difference. Therefore, as the input bit number increases, the slope of power consumption increase in the conventional carry select adder would be larger than that in our proposed design. As the input bit number of the conventional carry select adder increases to 32-bit, the power consumption in the conventional carry select adder will be 3.3 times larger than that in our proposed area-efficient carry select adder.

The conventional carry select adder performs better in terms of speed. The delay of our proposed design increases slightly because of logic circuit sharing sacrifices the length of parallel path. However, the proposed area-efficient carry select adder retains partial parallel computation architecture as the conventional carry select adder design; the delay increment of the proposed design is similar to that in the conventional design as the input bit number increases. We also simulated the delay performance in the proposed area-efficient adder and conventional carry select adder with 4, 8, 16, and 32-bit respectively. Fig. 7 shows the simulation results of the proposed area-efficient carry select adder and conventional carry select adder. The delay difference existing between these two designs is mainly come from the length difference in their parallel paths. In the conventional carry select adder, it divided N bits into M blocks; however, our proposed design divided every single bit as individual block. In other words, we still retain N blocks in the N bits adder. Such arrangement will lead to some speed sacrifice.

We further analyze the Power-Delay-Product as shown in Fig. 8. The simulation was performed in the proposed area-efficient carry select adder and conventional carry select adder with 4, 8, 16, and 32-bit respectively. We can find out that the PDP of our proposed design is smaller as compare with the conventional carry select adder and carry ripple adder design. The difference of PDP between these three designs is small in the case of the smaller input bit number. However, as the input bits increases, the slope of power consumption increment in the conventional carry select adder would be larger than that of the proposed design. Our proposed design can compute the addition function more efficiently by means of logic circuit sharing and partial parallel computation architecture retaining; therefore, the power saving ratio in our design would be much higher than the ratio of speed sacrifice.

Simplifying the carry select adder through logic simplification and partial logic circuit sharing can make the

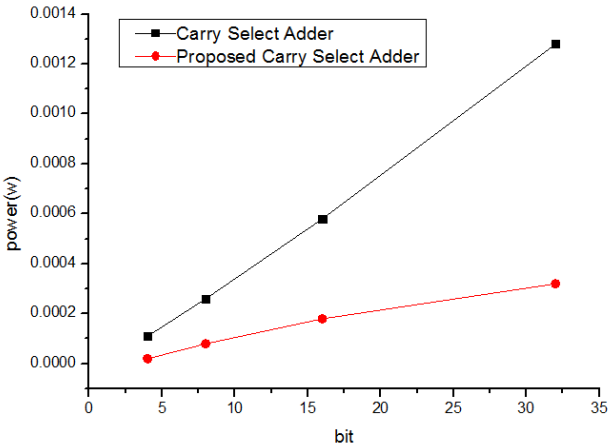

Fig. 6 The simulation results of the power consumption comparison in the proposed area-efficient carry select adder and the conventional carry select adder.

ISBN: 978-988-19251-9-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

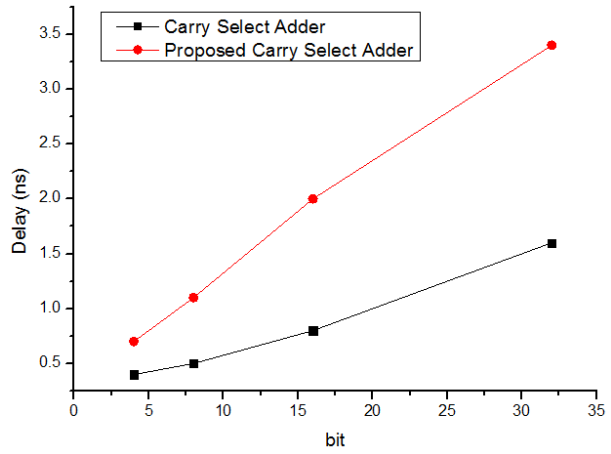

Fig. 7 The simulation results of the computation speed comparison in the proposed area-efficient carry select adder and the conventional carry select adder.

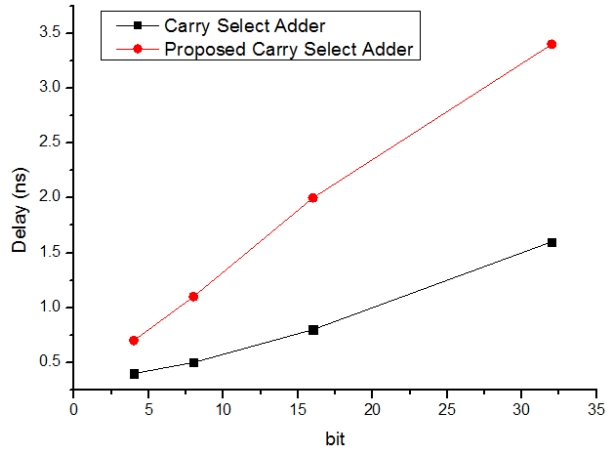

Fig. 8 The simulation results of the power delay product comparison in the proposed area-efficient carry select adder and the conventional carry select adder.

carry select adder more area-efficient and more power-efficient. The performance index of transistor count, power, delay, and PDP are summarized in Table 1. As compared with the carry ripple adder, operation speed in our proposed carry select adder can be much faster; however, transistor count and power consumption only increase slightly. In the case of a 32-bit adder, the transistor count in the carry ripple adder is 896. The transistor count in our proposed area-efficient carry select adder is 960, which only increases 7%. However, the transistor count in the conventional carry select adder is 1974, which increases more than twice. In terms of power consumption, we can save much power through removal of redundant logic and redundant signal switching by means of sharing common Boolean logic term. As compared with the conventional carry select adder, we can save 70% power. Relative to the carry ripple adder, we only increase 2% power. As a result, our proposed area-efficient carry select adder can perform the lowest PDP, which is only 60% of conventional carry select adder and 66% of carry ripple adder, respectively.

IV. Conclusion

In this paper, an area-efficient carry select adder is proposed. By sharing the common Boolean logic term, we can remove the duplicated adder cells in the conventional carry select adder. In this way, the transistor count in a 32-bit carry select adder can be greatly reduced from 1947 to 960. Moreover, the power consumption can be reduced from 1.26mw to 0.37mw as well as power delay product reduced from 2.14mw*ns to 1.28mw*ns. By retaining part of parallel architecture of conventional carry select adder, we can still maintain some competitiveness in speed. In this way, our area-efficient adder can perform with nearly the same transistor count, nearly the same power consumption, but with faster speed and lower PDP as compared with the carry ripple adder.

Table 1 Performance summary for the proposed area-efficient carry select adder, the conventional carry select adder, and the conventional carry ripple adder.

Adder Type	CRA	CSA	Proposed CSA
Transistor	896	1974	960
Power (W)	$3.72*10^{-3}$	$1.26*10^{-3}$	$3.77*10^{-4}$
Delay (S)	$5.25*10^{-9}$	$1.70*10^{-9}$	$3.40*10^{-9}$
PDP (W*S)	$1.95*10^{-12}$	$2.14*10^{-12}$	$1.28*10^{-12}$

References

[1] J. M. Rabaey, Digital Integrated Circuits," IEEE Trans. on VLSI Systems, 2003.
[2] O. Bedrij, "Carry Select Adder," IRE Trans. on Electronic Computers, Vol. EC-11, pp. 340-346, 1962.

ISBN: 978-988-19251-9-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)