

# An Area Efficient 64-bit Square Root Carry-select Adder for Low Power Applications

Yajuan He, Chip-Hong Chang and Jiangmin Gu

Centre for High Performance Embedded Systems, Nanyang Technological University,  
50 Nanyang Drive, Research Techno Plaza, 3rd Storey, Border X Block, Singapore 637553

**Abstract**—Carry-select method has deemed to be a good compromise between cost and performance in carry propagation adder design. However, conventional carry-select adder (CSL) is still area-consuming due to the dual ripple-carry adder structure. The excessive area overhead makes CSL relatively unattractive but this has been circumvented by the use of add-one circuit introduced recently. In this paper, an area efficient square root CSL scheme based on a new first zero detection logic is proposed. The proposed CSL witnesses a notable power-delay and area-delay performance improvement by virtue of proper exploitation of logic structure and circuit technique. For 64-bit addition, our proposed CSL requires 44% fewer transistors than the conventional one. Simulation results indicate that our proposed CSL can complete 64-bit addition in 1.50 ns and dissipates only 0.35mW at 1.8V in TSMC 0.18  $\mu$ m CMOS technology.

## I. INTRODUCTION

Addition is by far the most fundamental arithmetic operation. It has been ranked the most extensively used operation among a set of real-time digital signal processing benchmarks from application-specific DSP to general-purpose processors [1]. In particular, carry-propagation adder (CPA) is frequently part of the critical delay path limiting the overall system performance due to the inevitable carry propagation chain. For example, the delay of a fast CPA for converting the final carry-saved number to its two's complement form in a Wallace tree multiplier is typically 25% to 35% of the total multiplier delay [2].

Among the myriad of aggressive techniques, carry-select adder (CSL) has been an eminent technique in the space-time tug-of-war of CPA design. It exhibits the advantage of logarithmic gate depth as in any structure of the distant-carry adder family. Conventionally, CSL is implemented with dual ripple-carry adder (RCA) with the carry-in of 0 and 1, respectively. Depending on the configuration of block length, CSL is further classified as either linear or square root. The basic idea of CSL is anticipatory parallel computation. Although it can achieve high speed by not waiting for the carry-in from previous sub-block before computation can begin, they consume more power due to doubling the amount of circuitry needed to do the parallel addition of which half of the speculative

computations will be redundant. To obtain a lower transistor count, an add-one circuit was proposed by T.Y. Chang [3]. One group of RCA is replaced by an add-one circuit to achieve a 29.2% area reduction at the expense of 5.9% speed penalty for a 64-bit CSL over the conventional dual RCA design. The circuit was further modified by Y. Kim [4] to achieve even better performance. Unfortunately, an obscure flaw was found and the design as depicted in circuit architecture schematic of [4] has simulated to be functionally incorrect due to the missing of a multiplexer in the most significant bit position of the add-one block.

What has not been conceived in earlier designs of CSL is the power consumption. Due to the relentless drive for smaller and versatile mobile and portable electronics, power has now become a premier concern in DSP design. From power perspective, gate output load which is an aggregate of circuit fan-out and wire capacitance is as important as the gate depth. The significance of wire capacitance to gate delay and power consumption is particularly pronounced in today deep sub-micron regime. Therefore, it is imperative to combine logic structure with circuit technique to further reduce the transistor count of CSL so as to decrease the wire length and simplify the layout. Very often, area and power optimization are ensued from sensible reduction of transistor count. In this paper, a square root scheme with a new add-one circuit using one-inverter instead of two-inverter buffer has been proposed for the design of an area efficient 64-bit CSL. The proposed CSL outperforms the recently reported CSLs in both power-delay product and area-delay product.

## II. CARRY-SELECT ADDER AND ADD-ONE CIRCUIT

The carry-select adder partitions the adder into several groups, each of which performs two additions in parallel. Therefore, two copies of ripple-carry adder act as carry-evaluation block per select stage. One copy evaluates the carry chain assuming the block carry-in is zero, while the other assumes it to be one. Once the carry signals are finally computed, the correct sum and carry-out signals will be simply selected by a set of multiplexers. A typical block

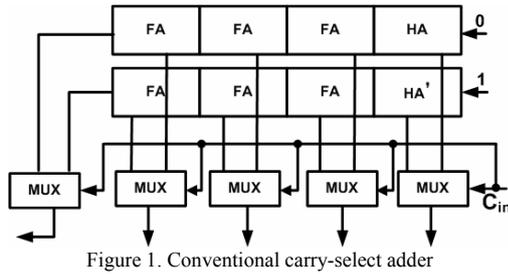


Figure 1. Conventional carry-select adder

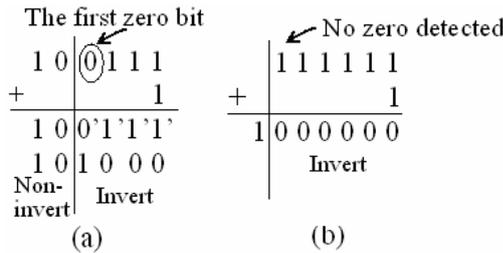


Figure 2. Examples for the first zero detection logic

of conventional CSL is shown in Fig. 1. FA and HA are abbreviations for full adder and half adder, respectively, and HA' is a full adder with a constant carry-in of logic 1.

The main drawback of the conventional CSL is the doubling of the area cost to duplicate another adder. Assume  $S^0 = (S_{n-1}^0, S_{n-2}^0, \dots, S_0^0)$  and  $S^1 = (S_{n-1}^1, S_{n-2}^1, \dots, S_0^1)$  are the sum outputs of these two copies of RCA with block carry-in  $c_{-1}^0 = 0$  and  $c_{-1}^1 = 1$ , respectively. The add-one circuit proposed by Chang [3] mitigates the resource overhead of CSL by replacing one copy of the RCA by

$$S^1 = S^0 + 1. \quad (1)$$

Let  $k$  denote the position that the first bit of "0" is detected in  $S_i^0$ , starting from the least significant bit (LSB). Then, from  $\prod_{i=0}^k S_i^0 = 0, k \in [0, n-1]$ , we have

$$S_t^1 = (S_t^0)' \quad t \in [0, k], \quad (2)$$

$$S_t^1 = S_t^0 \quad t \in [k+1, n-1] \quad (3)$$

where the single quotation mark on the superscript implies Boolean complement. From the above derivation, the add-one circuit is in essence, based on a "first" zero detection logic. It generates  $S^1$  by inverting each bit in  $S^0$  starting from the LSB until the first zero is encountered as shown in Fig. 2(a). However, if no zero is detected in  $S^0$  as illustrated in Fig. 2(b), i.e.,  $k \notin [0, n-1]$ ,  $S^1 = (1, (S_{n-1}^0)', ((S_{n-2}^0)', \dots, ((S_0^0)')$ . In other words, the carry-out signal for the add-one circuit is one if and only if all the sum outputs from the  $n$ -bit block are one. As all sums equal one, the first zero detection circuit generates one at the final node. For all the other cases, it generates a zero carry-out. As oppose to using dual RCAs in conventional CSL, the architecture of contemporary CSL adder comprises a single RCA, a first zero detection and selective complement add-one circuit, and a carry-select multiplexer circuit [3], as shown in Fig. 3.

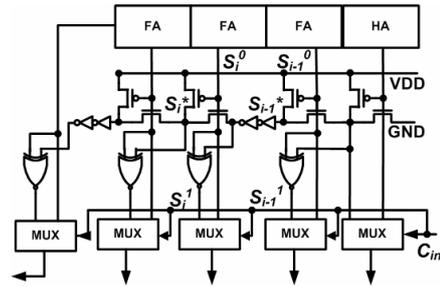


Figure 3. CSL adder with single ripple-carry adder and add-one circuit

### III. PROPOSED ADD-ONE CARRY-SELECT ADDER SCHEME

#### A. 64-bit square root carry-select adder design

Since the speed of a linear CSL is linearly proportional to the bit length  $n$ , thus, to optimize the worst-case delay, square root scheme will be used in this design of CSL with variable-sized blocks and ripple-carry addition in each block [5]. Conventionally, an  $n$ -bit square root carry-select adder can be divided into  $p$  stages with sizes  $s_1, s_2, \dots, s_p$ ,  $\sum_{i=0}^p s_i = n$ . In an ideal square root scheme, the block size is designed to optimally match the signal arrival time at the final multiplexer input to the delay time of carry-in signal. To determine the optimal variable block sizes, the latencies of primitive gates used in the conventional 64-bit CSL have been simulated for the same driving strength and standard output loading. The results are listed in TABLE I. HA and HA' are built with transmission gates to speed up the worst-case delay. The delay time of MUX (sel) refers to the delay of the multiplexer from the select signal to the output signal and MUX (thru) refers to the delay from the input signals to be selected to the output signal. FA (sum), HA (sum) and HA' (sum) refer to the delays from the input to the sum output. The delays from the input to the carry output are similarly annotated with "(Cout)". According to these basic gate latencies, it is evident that there will be mismatch of arrival time between the carry-select signal and the sum signals to the MUX in a square root CSL. The equalization of the delays through both paths can be achieved by progressively adding more bits to the subsequent stages of adder groups, so that more time is required for the generation of carry signals. Thus, the block sizes of our 64-bit CSL can be determined as indicated in TABLE II. Starting from two-bit RCA per group for the first two groups, the bits beyond the fifth bit are grouped in such a way that the number of bits in the group increases by one progressively. In this way the discrepancy in arrival time at the MUX nodes will be minimized. As the block delay of the conventional square root CSL is very similar to ours, the same configuration of CSL block sizes has been adopted in our proposed design. The worst case delay happens when the carry propagates from the LSB to MSB.

TABLE I. LATENCY OF BASIC GATES

Basic Gates	Delay Time (ps)
Inverter	33
NAND	54
XOR	86
MUX (sel)	94
MUX (thru)	42
FA (Sum)	291
FA (Cout)	212
HA (Sum)	91
HA (Cout)	114
HA' (Sum)	122
HA' (Cout)	143

TABLE II. BLOCK SIZES OF 64-BIT SQUARE ROOT CSL

Block no.	11	10	9	8	7	6	5	4	3	2	1
RCA, $n =$	8	10	9	8	7	6	5	4	3	2	2

Each CSL block contains one copy of RCA and an add-one circuit. The actual sum is selected using 2-to-1 MUX by the carry of previous block. This carry-out signal from the previous block also selects the actual carry-out for the next block. Due to the capacitive loading, the carry-out signal of the CSL block has to be buffered before it is applied to the next block. As to be elaborated in Section B, we will use NAND gates as buffers and this extra NAND gate delay incurred is less than that of a 1-bit full adder delay from TABLE I. Therefore, the carry-in of the CSL block would be ready slightly earlier than the output of its add-one circuit. Once the add-one circuit has completed its computation, the correct sum and carry-out of the block will be piped through a bank of MUX.

### B. New add-one scheme

For CSL with large operand, the longest RCA may contain a long carry chain. Therefore, a buffer should be inserted between every two pass transistors to restore the drive and logic level of the decaying signal strength along cascaded chain of pass transistors. To simply the layout and lower the transistor count for further interconnect and logic area reduction, we propose a new add-one scheme, which employs single inverter buffers and uses only MUX to substitute exclusive NOR gates along with MUX.

As shown in Fig. 3, the complement of the sum bit is generated from the internal nodes of PMOS-NMOS chain. Before the first zero is detected, each PMOS-NMOS pair functions as an inverter. Once the first zero occurs, it acts as a MUX and the correct sum is selected as described by

$$S_i^* = S_i^0 \cdot S_{i-1}^* + \overline{S_i^0} \quad (4)$$

$$S_i = S_i^1 \cdot C_{in} + S_i^0 \cdot \overline{C_{in}} = (S_i^0 \odot S_{i-1}^*) \cdot C_{in} + S_i^0 \cdot \overline{C_{in}} \quad (5)$$

Fig. 4 depicts our proposed add-one circuit using buffers with only one inverter. In what follows, we will prove that the add-one circuit with single inverter buffers performs exactly the same function as that shown in Fig. 3. With reference to Fig. 4, there is no change in the output

$$S_x = \overline{S_i^0 \cdot S_{i-1}^*} + \overline{S_i^0 \cdot 0} = \overline{S_i^0 \cdot S_{i-1}^*} \quad (6)$$

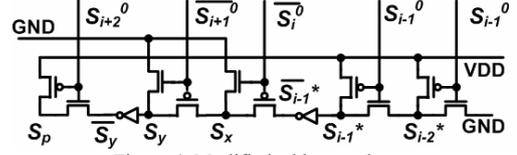


Figure 4. Modified add-one scheme

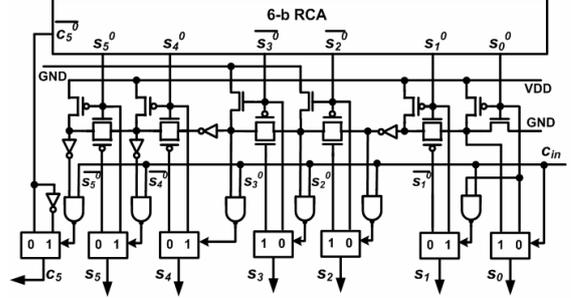


Figure 5. A 6-bit CSL with the proposed add-one circuit

According to (4),

$$\overline{S_i^*} = \overline{S_i^0 \cdot S_{i-1}^* + \overline{S_i^0}} = \overline{S_i^0 \cdot S_{i-1}^*} \cdot S_i^0 = \overline{S_{i-1}^*} \cdot S_i^0 \quad (7)$$

Therefore,

$$S_x = \overline{S_i^*} \quad (8)$$

Similarly,

$$S_y = S_{i+1}^0 \cdot S_x = S_{i+1}^0 \cdot \overline{S_i^*} = \overline{S_{i+1}^*}, \quad \overline{S_y} = S_{i+1}^* \quad (9)$$

$$S_p = S_{i+2}^0 \cdot S_{i+1}^* + \overline{S_{i+2}^0} = S_{i+2}^* \quad (10)$$

The above derivation verifies that our modified add-one circuit is functionally equivalent to that in Fig. 3, while the total number of inverters has been reduced by half. There is no speed penalty and in fact, it is envisaged that with the shorter chain and potential reduction of internal signal toggling, power dissipation will be lowered.

From (5), we have:

$$\begin{aligned} S_i &= S_i^0 \cdot S_{i-1}^* \cdot C_{in} + \overline{S_i^0 \cdot S_{i-1}^*} \cdot C_{in} + S_i^0 \cdot \overline{C_{in}} \\ &= S_i^0 \cdot (\overline{S_{i-1}^*} \cdot C_{in}) + \overline{S_i^0} \cdot (S_{i-1}^* \cdot C_{in}) \end{aligned} \quad (11)$$

This implies that the final sum and carry-out signals can be generated by using only NAND gates and MUX. The output,  $S_i$  is selected from  $S_i^0$  and its complement. The select signal  $\overline{S_{i-1}^*} \cdot C_{in}$  is generated by an NAND gate from the carry-in,  $C_{in}$  and the inverse of  $S_{i-1}^*$ .

A 6-bit CSL with the new add-one circuit is shown in Fig. 5. In our design, the RCAs are built with CMOS mirror topologies since this is the most interesting implementation in terms of its trade-off between power and delay performances [6]. Transmission gates are used in the first zero detection circuit to avoid the threshold voltage drop problem of pass transistor. At the bottom, the add-one circuit is connected to a group of MUX. These MUX are required for each output bit to choose from either sum or the complement of sum according to the control signal. The control signals are the outputs from the NAND gates which also function as buffers to improve the driving capability.

TABLE III. COMPARISON OF 64-BIT SQUARE ROOT CARRY-SELECT ADDERS @ 100MHz ON 1.8V SUPPLY

64-bit CSL	This work	Conv. CSL	Chang's CSL [2]	Kim's CSL [3]
Delay (ns)	1.501	1.493	1.588	1.507
Power (mW)	0.350	0.651	0.487	0.399
PDP (pJ)	0.53	0.97	0.77	0.60
AT (%)	69.9	100	92.4	78.7
AT <sup>2</sup> (%)	70.3	100	98.3	79.4
No. of Trans.	2535	3644	3166	2841

#### IV. SIMULATION RESULTS

We analyze and compare the performance of our proposed square root CSL against two recently proposed competitive ones [3][4]. We replicate the contender circuits as reported in the literature to obtain the transistor count and simulate the worst case delay and average power consumption of their circuits. In the process, it is found that the schematic given in [4] contains an error and the circuit has been rectified to ensure its correct functionality before simulations. A simulation environment realistic to the actual circuit operational conditions has been set up, where the cell has both driving and driven circuit. All the 128 bit inputs are loaded from the input buffers before they are fed into the 64-bit CSL circuit and the 65 bit outputs are also loaded to the buffers after they are exported [7]. All the circuits are simulated using HSPICE based on the TSMC 0.18 $\mu$ m CMOS process model. The threshold voltages of the PMOS and NMOS transistors used are around 0.46V and 0.48V, respectively. The transistors are sized using a consistent optimization strategy. For each simulation, HSPICE will generate an average power consumption value. As the dynamic power dissipation increases linearly with frequency and quadratic with supply voltage, the power dissipation is simulated at 100MHz and 1.8V with 1024 randomly generated input data. Comparison of the three carry-select adders in terms power dissipation are listed in TABLE III. The power- delay product (PDP) and area-delay products (AT and AT<sup>2</sup>) are also provided to evaluate the performances for different application criteria.

From TABLE III, our proposed CSL has a comparable delay to the conventional one, slower by a negligible 8ps. This could probably be due to the results of the add-one circuit is derived from the block with carry-in 0. Thus, the delay time of the sum of the MSB for  $C_{in}=1$  in the add-one circuit will be slightly longer than that in the dual RCA structure. The proposed adder is faster than the other two contenders, and consumes the least power among all. Its power consumption has been reduced significantly by 46%, 28% and 12% in comparison with the conventional CSL, Chang's CSL [3] and Kim's CSL [4], respectively. It requires only 70% of transistors of the conventional CSL, and 80% and 89% of those of Chang's and Kim's CSL. Despite no prominent speed advantage over that of the conventional and Kim's circuits, our proposed adder has

distinct superiority in both power-delay product and area-delay products over its contenders.

#### V. CONCLUSION

In this paper, we have presented a 64-bit square root carry-select adder with only one carry evaluation block and one modified add-one circuit instead of a dual ripple-carry adder structure. Progressively expanding RCA block sizes has been optimized for just in time computation using gate delay model. A new add-one circuit with less transistor count is proposed. The complete circuit exhibits low logic complexity and reduced power dissipation with no degradation to speed. A fair comparison was made among all competitive add-one based CSL and conventional CSL in terms of power and delay metrics. The number of transistors used in each circuit is also accounted. All in all, the simulation result indicates that our proposed square root carry-select adder outperforms all the other CSLs with minimal number of transistors and the least power-delay and area-delay products.

#### ACKNOWLEDGEMENT

The authors would like to thank Panasonic Singapore Laboratory for the PSL-NTU Joint R&D Collaboration (No. M48900001).

#### REFERENCES

- [1] D. C. Chen, L. M. Guerra, E. H. Ng, M. Potkonjak, D. P. Schultz and J. M. Rabaey, "An integrated system for rapid prototyping of high performance algorithm specific data paths," in *Proc. Application Specific Array Processors*, pp. 134-148, Aug. 1992.
- [2] R.K. Kolagotla, H. R. Srinivas and G. F. Burns, "VLSI implementation of a 200MHz 16 $\times$ 16 left-to-right carry-free multiplier in 0.35 $\mu$ m CMOS technology for next-generation DSPs," in *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 469-472, May 1997.
- [3] T. Y. Chang and M. J. Hsiao, "Carry-select adder using single ripple-carry adder," *Electronics Letters*, vol. 34, no. 22, pp. 2101 – 2103, Oct. 1998.
- [4] Y. Kim and L. S. Kim, "64-bit carry-select adder with reduced area," *Electronics Letters*, vol. 37, no. 10, pp. 614 – 615, May 2001.
- [5] J. M. Rabaey, *Digital Integrated Circuits - A Design Perspective*. Prentice Hall Press, 2001.
- [6] M. Alioto, G. Palumbo, "Analysis and comparison on full adder block in submicron technology," *IEEE Trans. VLSI Systems*, vol. 10, pp. 806–823, Dec. 2002.
- [7] A. Shams, T. Darwish and M. Byoumi, "Performance analysis of low-power 1-bit CMOS full adder cells," *IEEE Trans. VLSI Systems*, vol. 10, pp. 20-29, Feb. 2002.