International Journal of Computer and Communication Technology

Volume 7 | Issue 2 Article 8

April 2016

An Area Efficient 32-bit Carry-select Adder for Low Power Applications

N. Ravikumar
Department of ECE, Anurag Engineering College, Kodad, nvn_ravikumar@yahoo.com

M. Vishwanath
VLSI, MITS College, Kodad, mvisu87@gmail.com

B.Durga Malleswara Reddy
Department of ECE, Anurag Engineering College, Kodad, bdmreddy@hotmail.com

Recommended Citation

Ravikumar, N.; Vishwanath, M.; and Reddy, B.Durga Malleswara (2016) "An Area Efficient 32-bit Carry-select Adder for Low Power Applications," *International Journal of Computer and Communication Technology*: Vol. 7 : Iss. 2 , Article 8.
DOI: 10.47893/IJCCT.2016.1349
Available at: https://www.interscience.in/ijcct/vol7/iss2/8

An Area Efficient 32-bit Carry-select Adder for Low Power Applications

N. Ravikumar[1], M.Vishwanath[2] & B.Durga Malleswara Reddy[3]

[1&3]Department of ECE, Anurag Engineering College, Kodad
[2] VLSI, MITS College, Kodad
E-mail : nvn_ravikumar@yahoo.com[1], mvisu87@gmail.com[2], bdmreddy@hotmail.com[3]

Abstract – CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum [1]. However, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input Cin=0 and cin=1, then the final sum and carry are selected by the multiplexers (mux). The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position.

Keywords - *Application-specific integrated circuit (ASIC), area-efficient, CSLA, low power, MUX and BEC.*

I. INTRODUCTION

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].

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].

Among the myriad of aggressive techniques, carryselect adder (CSL) has been an eminent technique in thespace-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 32-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 32-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

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 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 $S0 = (S_{n-1}^0, S_{n-2}^0, \ldots, S_0^0)$ and

$$S1 = (S_{n-1}^1, S_{n-2}^1, \ldots, S_0^1)$$

are the sum outputs of these two copies of RCA withblock carry-in $c_{-1}^0 = 0$ and $c_{-1}^1 = 1$, respectively. The addone circuit proposed by Chang [3] mitigates the resource overhead of CSL by replacing one copy of the RCA by

$$S^1 = S^0 + 1 \qquad (1)$$

From the above derivation, the addone circuit is in essence, based on a "first" zero detection logic. It generates $S1$ by inverting each bit in $S0$ starting from the LSB until the first zero is encountered as shown in Fig. 2(a). However, if no zero is detected in $S0$ as illustrated in Fig. 2(b), i.e. $\notin[0,n-1]$, $S1 = (1, (Sn-1\ 0)', ((Sn-2\ 0)', \ldots, ((S0\ 0)')$. 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.

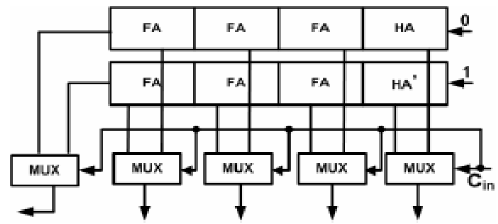

Figure 1. Conventional carry-select adder

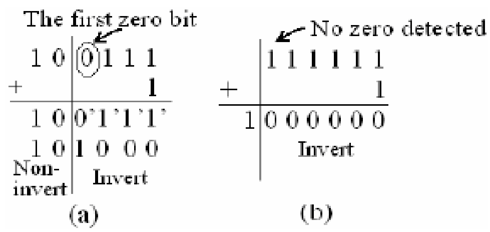

Figure 2. Examples for the first zero detection logic

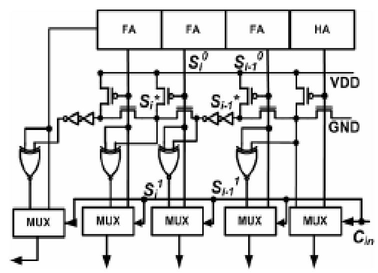

Figure 3. CSL adder with single ripple-carry adder and add-one circuit

III. PROPOSED ARCHITECTURE

A. 32-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 $s_1, s_2 ..$

$$S_p \quad \sum_{i=0}^{p} S_i = n$$

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 32-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 32-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.

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 neither 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

$$S_i^* = S_i^0 \cdot S_{i-1}^* + S_i^0$$

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

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

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

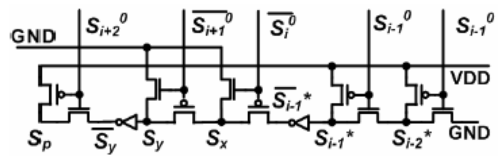

Figure 4. Modified add-one scheme

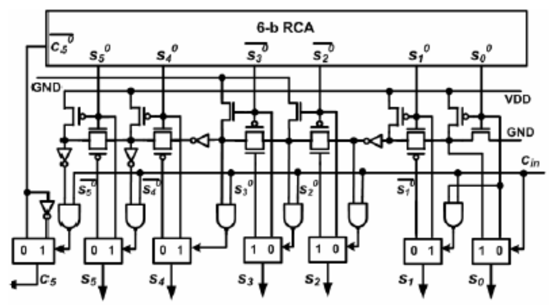

Figure 5. A 6-bit CSL with the proposed add-one circuit

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.

IV. SIMULATION RESULTS

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 0.18μm 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 AT2) are also provided to evaluate the performances for different application criteria.

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

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^2 (%)	70.3	100	98.3	79.4
No. of Trans.	2535	3644	3166	2841

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 *Cin*=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.

V. ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their comments which were very helpful in improving the quality and presentation of this paper.

REFERENCES:

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] R.K. Kolagotla, H. R. Srinivas and G. F. Burns, "VLSI implementation of a 200MHz 16×16 left-to-right carry-free multiplier in 0.35μm CMOS technology for next-generation DSPs," in Proc. IEEE Custom Integrated Circuits Conf., pp. 469-472, May 1997.

AUTHORS PROFILE:

Dr. N. RAVIKUMAR
nvn_ravikumar@yahoo.com
AP, India
Completed his Ph.D and presently working as Vice-Principal of Anurag Engineering College, Kodad with 16 years of Teaching Experience

M.VISHWANATH
mvisu87@gmail.com
AP, India
Pursuing Master Degree in VLSI in MITS college, KODAD and also has 2 years of Teaching Experience

B.Durga Malleswara Reddy
bdmreddy@hotmail.com
AP, India
Completed his Master Degree and working as Assistant Professor in ECE Department of Anurag Engineering College, Kodad with 5 years of Teaching Experience

❑❑❑