35th Asilomar Conference on Signals, Systems and Computers
Pacific Grove, California, November 4-7, 2001

Application of Logical Effort Techniques for Speed Optimization and Analysis of Representative Adders*

Hoang Dao, Vojin G. Oklobdzija
ACSEL Lab
Department of Electrical and Computer Engineering
University of California, Davis 95616
(hqdao,vojin)@ece.ucdavis.edu

Abstract

This paper presents the transistor-level analysis of contemporary 64-bit adders. The logical effort technique was applied to provide more descriptive presentation of the delay and circuit architecture. It also enabled optimization of gate size for optimal performance. The selected adders were dynamic carry-lookahead adder (DCLA), static carry-select adder (SCSA), dynamic Kogge-Stone adder (DKSA) and Ling/conditional-sum adder (DLCNSA). The results matched well with simulation using 0.18μm, 1.8V CMOS. Adders with fewer levels in the critical path showed superior performance. In particular, for dynamic adders, a 0.6-FO4 per-gate delay improvement was observed.

1. Introduction

Adder delay is critical in the design of high-performance processor. Unfortunately, it is normally presented in terms of gate delays or simulation result. The former format is not efficient because delay is dependent on gate types and the number of inputs; the latter does not help to relate the result to the adder architecture and is difficult to compare. In our analysis, the logical effort method is used to express the delay.

The logical effort (LE) analysis [1] is efficient in delay estimation and connects delay to adder architecture. It models the gate delay using gate characteristics and its loading and compares the gate delay to τ, the delay of a parasitic-free fanout-1 (FO1) inverter. This delay is normally known by designers for a given technology. So, the delay estimation by logical effort can be fairly accurate. Furthermore, it also accounts for the effect of circuit architectures in delay, via branching and gate loading. For analysis four contemporary adders: dynamic carry-lookahead adder, modified static carry-select adder [2], dynamic Kogge-Stone adder [3] and dynamic Ling/ conditional-sum adder [4] were chosen.

2. Optimization conditions

All adders were optimized under the following conditions: maximum input size of 20μm, maximal allowable transistor size of 20μm and an identical load of 30μm-equivalent inverter. These conditions were set to get reasonable transistor sizes for layout and an acceptable load to the adder.

The adders were optimized according to the critical paths, estimated from the adder architecture. Delay effort in other paths was derived from the critical one. The optimization process was done recursively until all transistor sizes converged.

The wiring capacitance was included and estimated from the width of the most congested cell. It was measured from the preliminary layout of the cell. This width was kept minimal to reduce the impact of the wiring capacitance.

3. Logical effort analysis of adders

The SCSA had 9 gates, all static (Figs. 2-3). The group-2 generate/propagate structure was chosen. Unlike [2], the complementary propagates were used to avoid extra inverter delays in critical path. Both critical paths were to the sum MSB, via either the generate path from the operand LSB or via the propagate path from the operand 35^{th} LSB. An extra inverter was added in both paths during the optimization to reduce the gate size and enable less optimal delay effort. The per-gate effort delay was 3.7τ.

The DCLA had 14 gates: 7 dynamic and 7 high-skew (Figs. 4-5). (High-skew gates have faster pull-up transitions than the pull-down ones; reverse is true for low-skew gates.) The Group-4 propagate and generate

* Supported under SRC Contract No. 2001-HJ-931

scheme was used. There were two possible critical paths to the sum MSB: the generate path via the operand LSB and the propagate path via the operand 6th LSB. Logical effort analysis was done assuming both paths were equal. The resulting per-gate effort delay was 1.4τ.

The DKSA had 6 gates: 3 dynamic and 3 high-skew (Figs. 6-7). It used redundant scheme for Group-4 propagates/generates. The critical path was from the operand LSB via the generate path to the sum MSB. The per-gate effort delay was 2.3τ.

The DLCNSA had 9 gates: 4 dynamic, 4 high-skew and 1 low-skew (Figs. 8-9). It combined the Ling pseudo-carry/propagate to generate long carries and the conditional sum for local carries. The critical path was the path through the long carry to the sum MSB. The per-gate effort delay was 2.4τ.

Table 1. Logical effort delay

64-b Adder	# Gates	LE (tau)	# FO4
Dynamic Kogge	6 (3 dyn, 3 hi-sk)	30.1	***6.2***
Dynamic Ling	9 (4 dyn, 4 hi-sk, 1 lo-sk)	43.9	***9.0***
Dynamic CLA	14 (7 dyn, 7 hi-sk)	54.3	***11.1***
Static CSA	9 (all static)	60.7	***12.4***

Table 1 summarized the result of the logical effort analysis. The delays were in term of the parasitic-free FO1 delay τ and FO4 delay of an inverter. The adders with fewer stages yielded smaller delay. In addition, the calculated delays did not clearly show a linear relationship to the number of gates in the critical path. The static adder also appeared to be 1.5x worse in delay.

4. Results

The results were confirmed using HSPICE simulation for the 0.18μm, 1.8V CMOS technology at room temperature. The worst-case delay of the critical path was measured. The result was presented in Table 2.

Table 2. Comparison with the simulation result

64-b Adder	LE (ps)	LE (norm)	SPICE (ps)	SPICE (norm)	Err (%)
D-KSA	482	***1.00***	581	***1.00***	*17.0*
D-LCNSA	702	***1.46***	742	***1.28***	*5.4*
D-CLA	868	***1.80***	1064	***1.83***	*18.4*
S-CSA	971	***2.01***	1059	***1.82***	*8.3*

The HSPICE results were consistent with the logical effort analysis. In addition, as shown in Fig. 1, for dynamic adders, the delay difference was close to 0.10x per gate, normalized to the delay of the DKSA, which corresponded to 0.60 FO4 or 2.9τ.

5. Conclusion

The comparison of adder performance was presented. The circuit optimization was done using logical effort technique. The adder performance was dependent on, but not proportional to, the number of gates in the critical path.

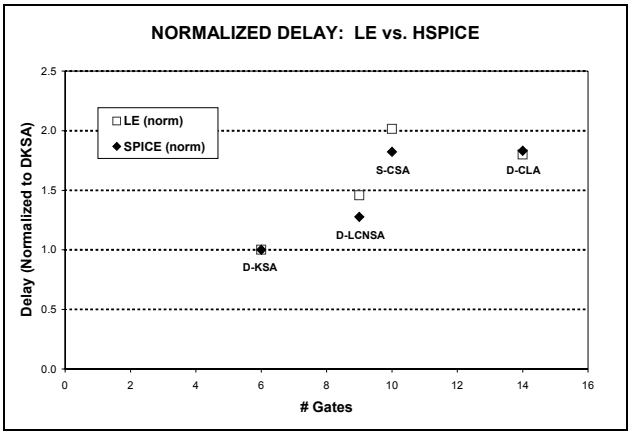

Fig. 1. Normalized delay

References:

[1] I. Sutherland, B. Sproull, D. Harris, "Logical Effort: Designing Fast CMOS Circuits," *Morgan Kaufmann Publisher*, c1999.

[2] A. Farooqui, V. G. Oklobdzija, F. Chehrazi, "Multiplexer-Based Adder for Media Signal Processing," *International Symposium on VLSI Technology, Systems, and Application*, pp.100-103, 1999.

[3] J. Park, H. C. Ngo, J. A. Silberman, S. H. Dhong, "470ps 64-Bit Parallel Binary Adder," *Symposium on VLSI Circuits Digest of Technical Papers*, pp. 192-193, 2000.

[4] S. Naffziger, "A Sub-Nanosecond 0.5μm 64b Adder Design," *ISSCC Digest of Technical Papers*, pp.362-363, 1996.

[5] A. Weinberger, J. Smith, "A Logic for High-Speed Addition," *National Bureau of Standard Circulars 591*, pp.3-12, 1958.

[6] B. D. Lee and V. G. Oklobdzija, "Improved CLA Scheme With Optimized Delay," *Journal of VLSI Signal Processing*, Vol. 3, No. 4, pp. 265-274, 1991.

[7] X. Xu, V. G. Oklobdzija, "Application of Logical Effort on Design of Arithmetic Blocks in VLSI CMOS Technology," *35th Asilomar Conference on Signals, Systems, and Computers*, November 2001.

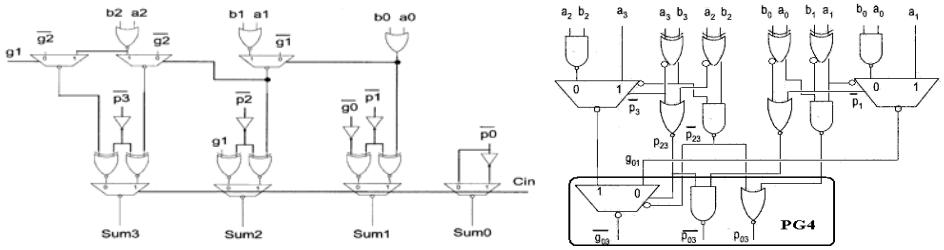

Fig. 2. Carry-select adder: circuit

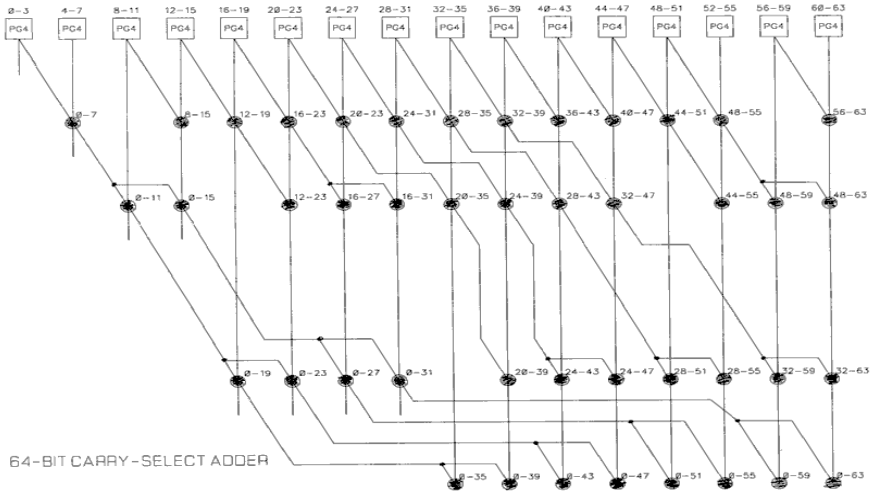

Fig. 3. Carry-select adder: diagram

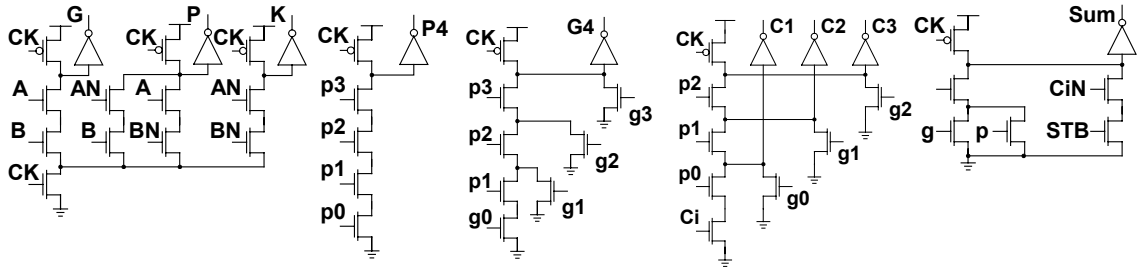

Fig. 4. Carry-lookahead adder: circuits

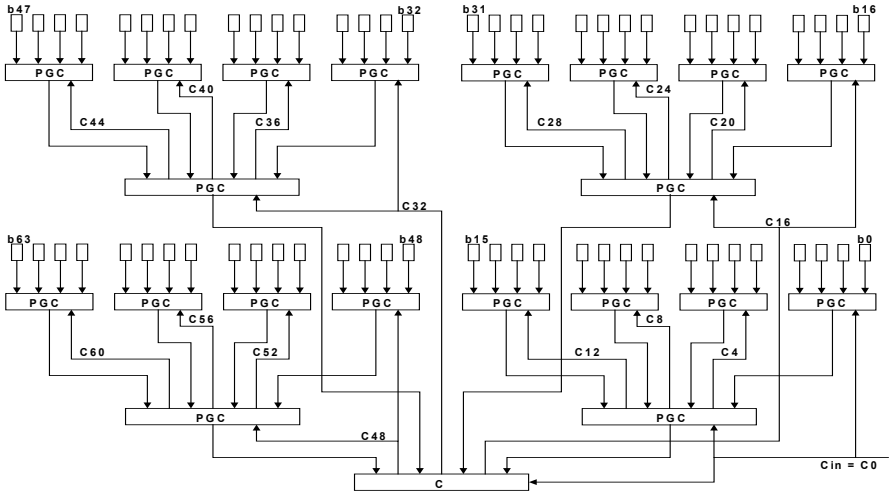

Fig. 5. Carry-lookahead adder: diagram

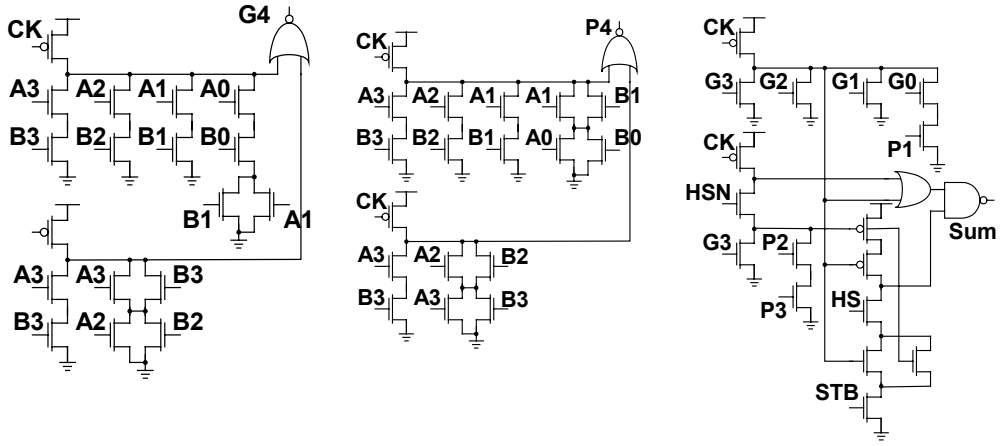

Fig. 6. Kogge-Stone adder: circuits

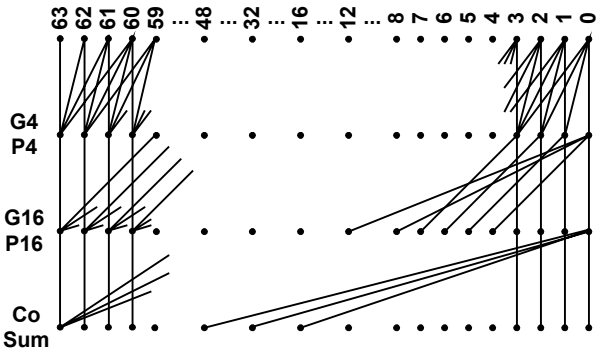

Fig. 7. Kogge-Stone adder: diagram

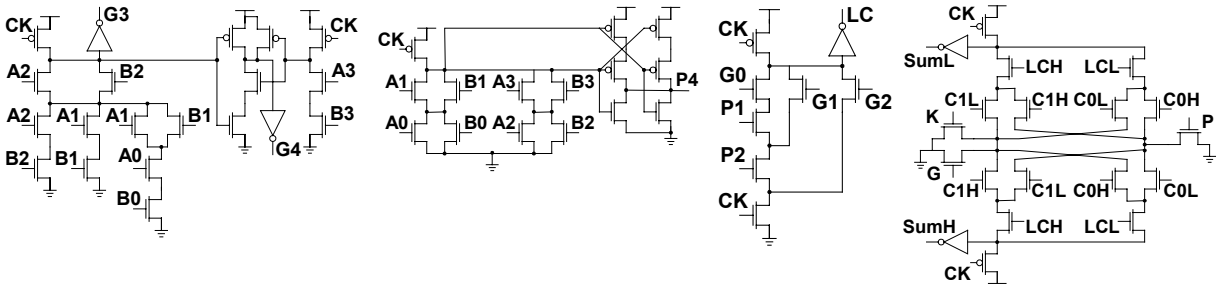

Fig. 8. Ling/conditional-sum adder: circuits

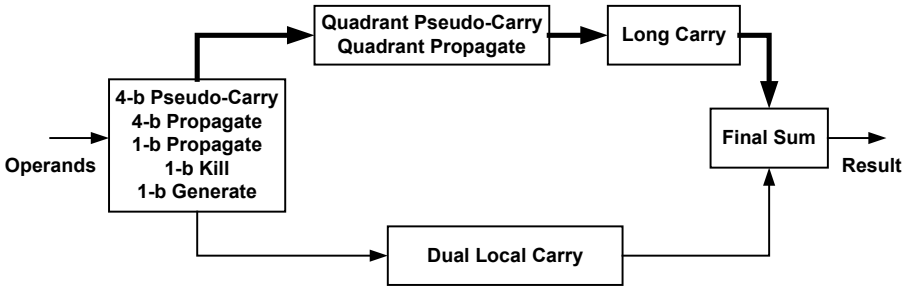

Fig. 9. Ling/conditional-sum adder: diagram