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

Application of Logical Effort on Delay Analysis of 64-Bit Static Carry-Lookahead Adder*

Hoang Q. 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 the 64-bit static carry-lookahead adder (CLA). The carry blocks were implemented in two schemes: (A) 2-level and (B) multilevel. The logical effort technique was used to optimize the circuits for best performance. The analysis was verified with SPICE simulation, using 0.18μm, 1.8V CMOS technology, and confirmed with small error. In addition, scheme B showed 12% improvement due to faster gate in carry block and less loading in (P,G) ones.

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 delays are very dependent on gate types and the number of inputs; the latter does not help to relate the result to the adder architecture itself and is difficult to compare. In our analysis, the logical effort method is proposed 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 loading and compares the gate delay to τ, the delay of a parasitic-free fanout-1 (FO1) inverter. The latter 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.

The 64-bit static carry-lookahead adder was chosen to illustrate the advantages of this technique. Two schemes of carry blocks were used to show the effect of gate and loading on total delay.

The paper is presented as followed. Section 2 outlined the conditions for circuit optimization. The analysis of the adder using logical effort technique was discussed in section 3. The results of LE analysis and HSPICE simulation were shown in section 4. The conclusion was given in section 5.

2. Optimization conditions

The CLA adder 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.

3. Analysis using logical effort

The fixed group-4 propagate and generate CLA structure was used (Fig. 1). Two schemes were implemented using static gates with (A) 2-level carry block and (B) multi-level carry block.

In scheme A (Fig. 2), the critical paths were found to the 63rd-bit sum, through the propagate path from the 21st bit or via the generate path from the least significant bit. The optimal per-gate delay effort was 2.9τ and resulted a total delay of 71τ or 14.5FO4.

In scheme B, the carry block was modified to exploit the non-critical path that did not go through C_{in}, (Fig. 3). Thus, the critical path in carry blocks went through NAND2 gates, instead of NAND4s. Furthermore, the loading on propagate and generate signals was also

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

reduced. The critical paths were found to the 63rd-bit sum, via the propagate path from the 26th bit or via the generate path from the least significant bit. The final per-gate delay effort was 2.5τ and the total delay was 63τ or 12.9FO4.

4. Results

The two implementations were simulated in HSPICE using the 0.18μm, 1.8V CMOS technology at room temperature. The worst-case delay of the critical path was measured and presented in Table 1.

The results using logical effort technique matched very well to HSPICE, at 3-5% error. In addition, the 12% improvement in scheme A was achieved by using faster gates in the critical paths and by reducing loading due to the non-critical paths.

5. Conclusion

The analysis of two CLA implementations using logical effort technique was presented. The results matched well to HSPICE simulation and therefore partially proved the usefulness of the method. It was also illustrated that, besides using better architecture, improvement on delay could be achieved if non-critical paths were exploited.

References:

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

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

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

[4] 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.

Table 1. Result of logical effort analysis and HSPICE simulation

Scheme	**Logical Effort**			**HSPICE**	***Error***
	Delay (τ)	*# FO4*	*Delay (ns)*	*Delay (ns)*	*(%)*
A: 2-Level Carry	71.0	14.5	1.14	1.20	5.3
B: Multi-Level Carry	62.9	12.9	1.00	1.03	3.0

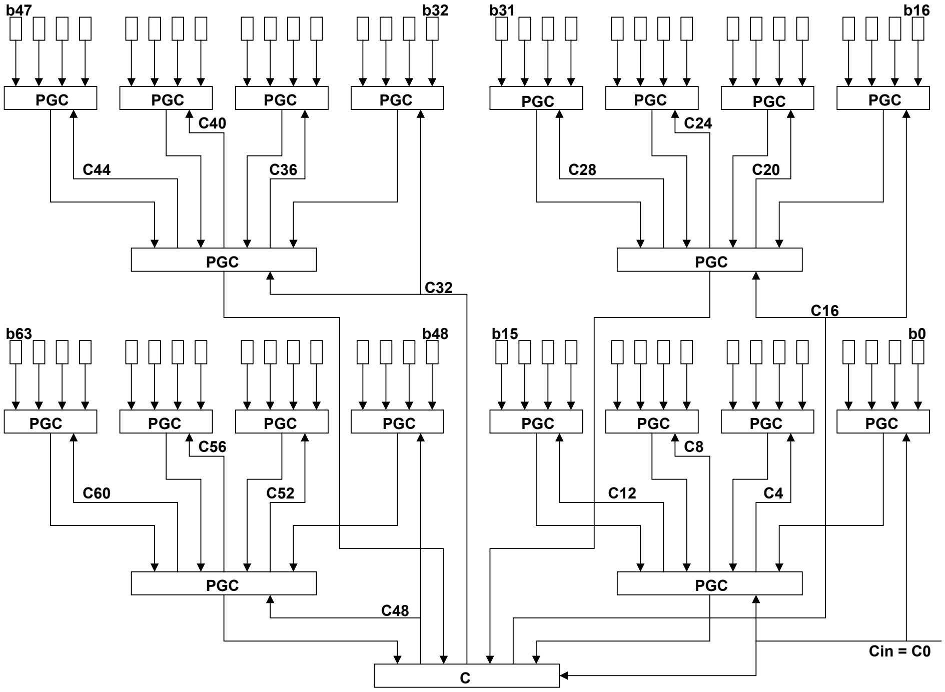

Fig. 1. Diagram of carry-lookahead adder

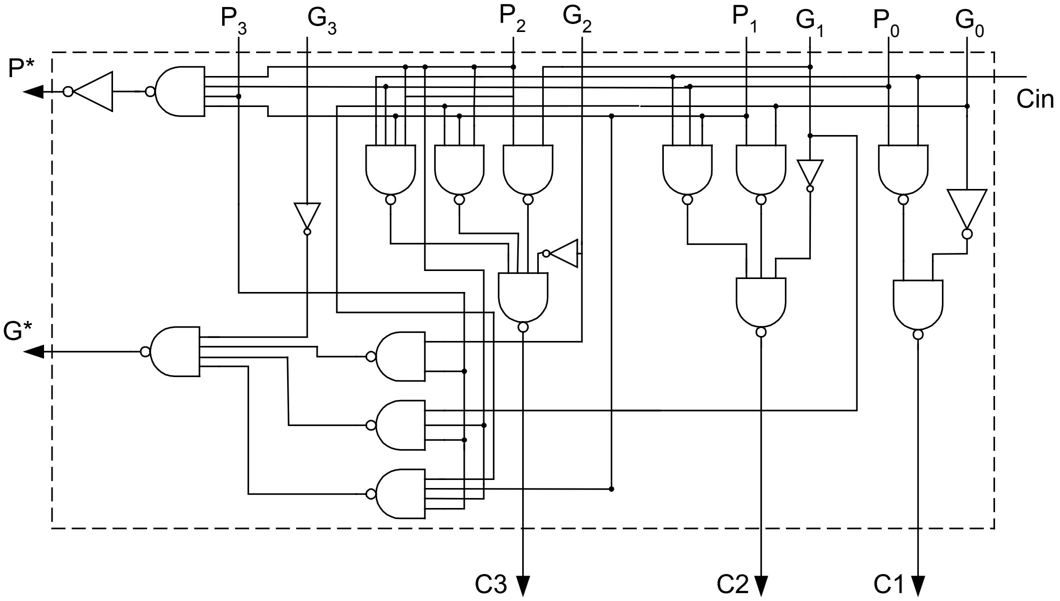

Fig. 2. Scheme A: (P, G, C) blocks

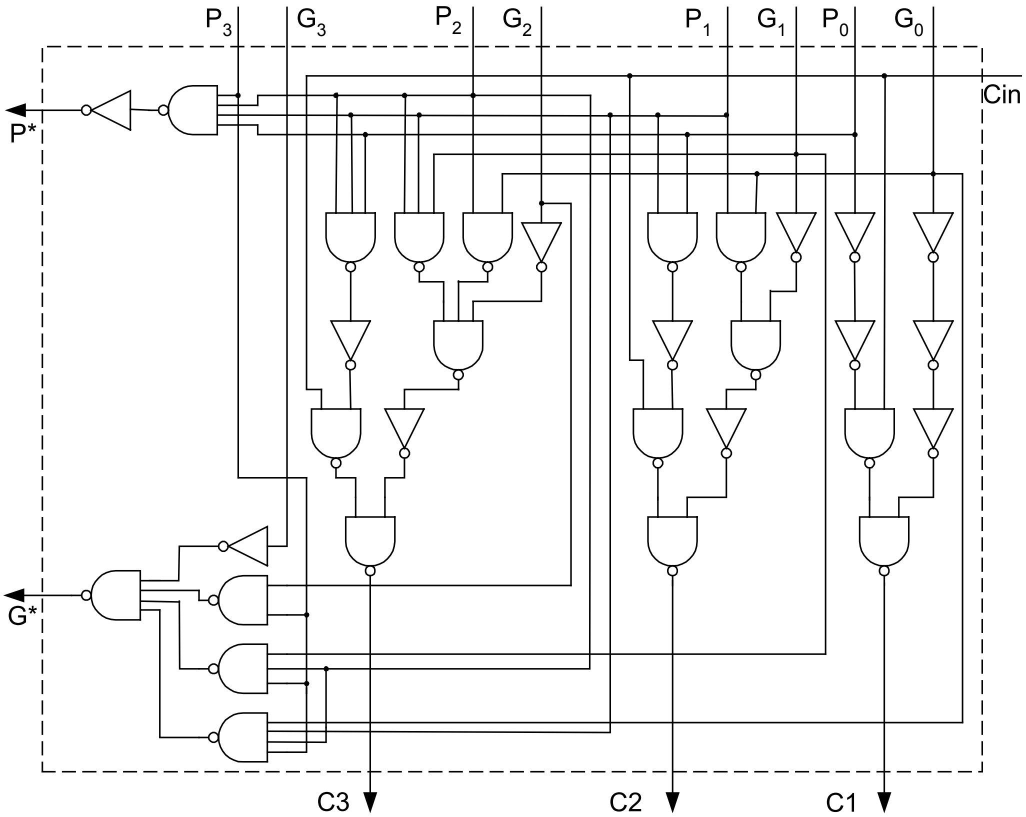

Fig. 3. Scheme B: (P, G, modified-C) blocks