Exploration of Switching Activity Behavior of Addition Algorithms

Dursun Baran, Mustafa Aktan, Hossein Karimiyan and Vojin G. Oklobdzija
School of Electrical and Computer Engineering
The University of Texas at Dallas, Richardson, Texas
Email:{dxb075000,aktanmus, hossein.karimiyan,vojin}@utdallas.edu

***Abstract*—Total energy consumption of a micro-architecture directly depends on the switching factors of the internal nodes. Weinberger and Ling are the most widely used binary addition algorithms that are used in microprocessor adders. In this paper, the switching activity behaviors of those well known addition recurrence structures are explored. The formulae for estimation of switching activity rates of signals given the probabilities of input signals are derived. The results reveal that the preference of one addition algorithm to other one in terms of switching factors depends on the statistical properties of the input signals as well. The addition algorithms are examined on Kogge-Stone structure and it is possible to save energy up to 20% by selection of the proper addition algorithm in 64-bit adders.**

I. Introduction

In the recent years, the energy efficient design have gained more attention and for highly utilized functional units, especially for the adders, energy efficient design becomes the first concern. The energy consumption of a microprocessor adder depends on the circuit sizing, the addition algorithm, the recurrence structure and the wiring complexity. Each adder structure has different trade-offs and the prior selection of proper adder topology is an important design strategy. For a fair comparison of different designs, switching activities are required to be considered for each structure.

Energy-Delay estimation technique is a quick and accurate method which can be used to compare different adder topologies and sizing methodologies [2]. A model for a quick way of energy estimation is proposed in [2], [5]. This technique has been used to compare different adder designs. As a typical scenario in adders, a chain of gates is shown in Figure 1. The delay estimation is done by use of Logical Effort [1]. Similar to the delay estimation, the energy consumption of the driver gate can be approximated by eq. 1:

$$E = \{E_p W_G \ + E_g(W_{L1}\text{+...+}W_{LN}) \}\alpha_{0->1} + E_{lk} \qquad (1)$$

where Ep(fJ/μm) is the energy factor for internal parasitic capacitance of driver, $W_G(\mu)$ is the size of driver gate, Eg(fJ/μm) is the energy factor for external loads, W_{L1},...,$W_{LN}(\mu)$ are the sizes of loads and $\alpha_{0->1}$ is the switching activity rate at node X in Figure 1. E_{lk} is the leakage energy consumption at non-switched nodes.

For accurate energy estimation, sizes of logic gates(W_G,$W_L s$), energy factors(E_g,E_p) which can be obtained from technology characterization and switching factor of internal nodes are required. However, the requirement of switching factor of internal nodes makes the energy estimation difficult to apply. The switching behavior of a logic gate depends on the inputs and the function of the gate. Weinberger and Ling are two most widely used binary addition algorithms in high performance VLSI adders. More explanations about the addition algorithms are provided in Section II. Weinberger and Ling have different kind of logic gates on the carry recurrence and sum path. Also, the internal connections differ for each algorithm. Therefore, it is possible to get energy savings by preferring one of the addition algorithms which has less switching activity rate.

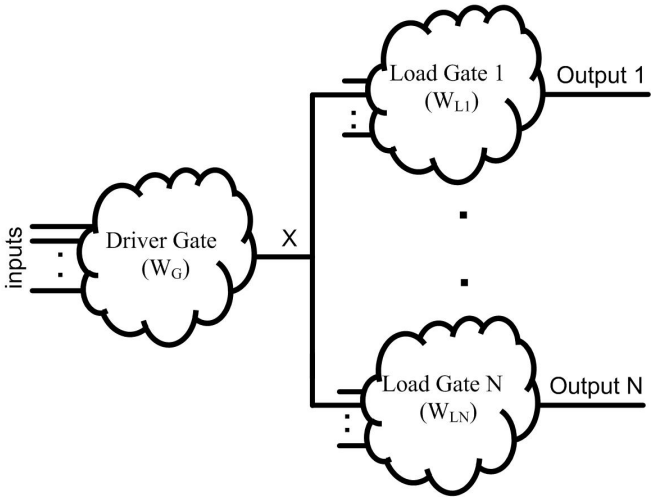

Fig. 1. Chain of Gates

The paper is organized as follows; section-II explores addition algorithms, section-III gives a method to calculate the switching activities of all nodes for static KS adders. The HSpice simulation results and energy comparison of addition algorithms are given in section-IV and section-V concludes the work.

II. ADDITION ALGORITHMS

As mentioned before, Weinberger and Ling are the most widely used addition algorithms in CMOS technology. Dorans transformations are not suitable for efficient CMOS realization [3], [6]–[8]. Doran shows that the recurrences they are more suitable in CMOS technology are Weinbeger and

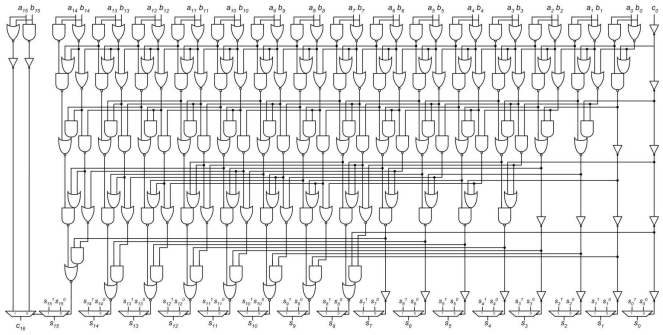

Fig. 2. Schematic of 16-bit Weinberger KS Adder

Ling [8]. As a brief explanation, Weinberger [6] introduced the concept of generate and propagate signals to allow parallel computation of carry signals to increase the speed of addition. Ling transformation simplifies the carry recurrence of Weinbeger at the cost of an increase in sum complexity as compared to Weinberger [3]. Also, Ling recurrence may give better performance than Weinberger [3]. For accurate energy comparison of those well-known recurrence algorithms, the switching activity of internal nodes needs to be considered.

Kogge-Stone (KS) is the one of the well known high performance and minimum depth parallel prefix adder structure, but it has high wiring complexity and energy consumption [4], [9]. This structure can be implemented by Weinberger and Ling addition algorithms and the schematic of 16-bit Kogge-Stone adder with Weinberger addition algorithm is shown in Figure 2 as an example.

The logic gates in carry path for both addition algorithms are the same except the first carry merge stage. Ling recurrence uses NAND gate in the first carry merge stage that is OAI gate in Weinberger. Also, the internal connections of gates are different from each other. In addition to them, sum generation blocks includes different logic gates and their complexities are different. The number of internal nodes is higher in Ling recurrence than Weinberger recurrence.

III. SWITCHING ACTIVITY CALCULATION

It is possible to explore both addition algorithms in all paralel prefix adder structures [4]. As a sample case, they are explored on Kogge-Stone structure. Under the assumption that all input signals of the circuit are spatially and temporally uncorrelated, switching activity of intermediate and output signals could be easily computed [10]. For a signal χ, let the probability of χ being logic '1'('0') be denoted by $p_1(\chi)$ ($p_0(\chi)$). The switching from logic 0 to 1 by the signal χ is referred as the switching event. The probability of this event is denoted as $p_{0->1}(\chi)$ and switching activity can be calculated from the signal probability as:

$$p_{0->1}(\chi) \ = p_0(\chi)\ p_1(\chi) \tag{2}$$

The carry signals in Weinberger recurrence can be calculated as:

$$C_i \ = g_{i-1} \ + {}^{\oplus}p_{i-1}C_{i-1} \tag{3}$$

In eq. 3, the terms of g_{i-1}, ${}^{\oplus}p_{i-1}$ and C_{i-1} are independent and the probability of carry signals can be calculated as:

$$p_1(C_i) \ = p_1(g_{i-1}) \ + p_1({}^{\oplus}p_{i-1})\ p_1(C_{i-1}) \tag{4}$$

The probabilities of $p_1(g_{i-1})$ and $p_1({}^{\oplus}p_{i-1})$ are easy to calculate. The probability of $p_1(C_{i-1})$ signal can be calculated from previous carry by use of eq. 4. Therefore, the probabilities of all carry signals can be calculated from previous carry signals iteratively. Other signals for Weinberger and Ling recurrences can be calculated in the same way as well.

The signals generated in Weinberger and Ling addition algorithms are given in Table I. The signal probabilities for those addition algorithms are given in Table II. Formulae are derived assuming inputs a_i and b_i s have an equal probability of ν. The probability of the input carry signal is κ. Given the input signal probabilities first the bitwise generate (g_i)

TABLE I
SIGNALS IN WEINBERGER AND LING ADDITION ALGORITHMS(N-BIT)

	Weinberger		Ling	
Inputs	a_i	input a(i=0,1,...,N-1)	a_i	input a(i=0,1,...,N-1)
	b_i	input b(i=0,1,...,N-1)	b_i	input b(i=0,1,...,N-1)
	c_0	input carry	c_0	input carry
Pre-Processing	g_i=$a_i b_i$	bitwise generate	g_i=$a_i b_i$	bitwise generate
	${}^{+}p_i$=a_i+b_i	bitwise propagate(or)	t_i=a_i+b_i	bitwise propagate(or)
	${}^{\oplus}p_i$=$a_i \oplus b_i$	bitwise propagate(xor)	${}^{\oplus}p_i$=$a_i \oplus b_i$	bitwise propagate(xor)
Carry Block	c_i=g_i+$p_i c_{i-1}$	carry	h_i=g_i+$t_{i-1} h_{i-1}$	pseudo-carry
	$G_{i:j}$=$G_{i:k}$+$P_{i:k} G_{k-1:j}$	group generate	$H_{i:j}$=$H_{i:k}$+$T_{i-1:k-1} H_{k-1:j}$	group pseudo-carry
	$P_{i:j}$=$P_{i:k} P_{k-1:j}$	group propagate	$T_{i:j}$=$T_{i:k} T_{k-1:j}$	group propagate
Outputs	s_i=$a_i \oplus b_i \oplus c_i$	sum	s_i= { $a_i \oplus b_i$, h_{i-1}=0; $a_i \oplus b_i \oplus t_{i-1}$, h_{i-1}=1	sum
	c_N=g_{N-1} + $p_{N-1}\ c_{N-1}$	output carry	c_N= { g_{N-1}, h_{N-2}=0; g_{N-1} + $t_{N-1}\ t_{N-2}$, h_{N-2}=1	output carry

TABLE II
SIGNAL PROBABILITIES FOR WEINBERGER AND LING ADDITION ALGORITHMS. SIGNAL PROBABILITIES OF LING RECURRENCE THAT ARE NOT SHOWN ARE SAME AS FOR WEINBERGER.

Weinberger	Ling
$p_1(a_i) = \nu$	
$p_1(b_i) = \nu$	
$p_1(c_0) = \kappa$	
$p_1(g_i) = \nu^2$	
$p_1(^{+}p_i) = \nu(2\text{-}\nu) = \gamma$	$p_1(t_i) = \nu(2\text{-}\nu) = \gamma$
$p_1(^{\oplus}p_i) = 2\nu(1\text{-}\nu) = \beta$	
$p_1(c_i) = \nu^2(1\text{-}\beta^i)/(1\text{-}\beta) + \kappa\beta^i$	$p_1(h_i) = \nu^2 + (1\text{-}\nu^2)p_1(c_i)$
$p_1(G_{i:j}) = \nu^2(1\text{-}\beta^{i-j+1})/(1\text{-}\beta)$	$p_1(H_{i:j}) = \nu^2 + (1\text{-}\nu^2)p_1(G_{i-1:j})$
$p_1(P_{i:j}) = \gamma^{i-j+1}$	$p_1(T_{i:j}) = \gamma^{i-j+1}$
$p_1(s_i) = \beta + (1\text{-}2\beta)(\nu^2(1\text{-}\beta^i)/(1\text{-}\beta)+\kappa\beta^i)$	

TABLE III
SIGNAL PROBABILITIES FOR THE CARRY-BLOCK OF A KOGGE-STONE ADDER

	Weinberger	Ling
Level 1	$p_1(G_{i:i-1}) = \nu^2(1+\beta)$	$p_1(H_{i:i-1}) = \nu^2(2\text{-}\nu^2)$
	$p_1(P_{i:i-1}) = \gamma^2$	$p_1(T_{i:i-1}) = \gamma^2$
Level 2	$p_1(G_{i:i-3}) = \nu^2(1\text{-}\beta^4)/(1\text{-}\beta)$	$p_1(H_{i:i-3}) = \nu^2+(1\text{-}\nu^2)\nu^2(1\text{-}\beta^3)/(1\text{-}\beta)$
	$p_1(P_{i:i-3}) = \gamma^4$	$p_1(T_{i:i-3}) = \gamma^4$
.	.	.
.	.	.
.	.	.
Level L	$p_1(G_{i:i-2^L+1}) = \nu^2(1\text{-}\beta^{2^L})/(1\text{-}\beta)$	$p_1(H_{i:i-2^L+1}) = \nu^2+(1\text{-}\nu^2)\nu^2(1\text{-}\beta^{2^L-1})/(1\text{-}\beta)$
	$p_1(P_{i:i-2^L+1}) = \gamma^{2^L}$	$p_1(T_{i:i-2^L+1}) = \gamma^{2^L}$
.	.	.
.	.	.
.	.	.

and propagate (p_i, t_i) are calculated. Using these probability (ν^2, β, γ) values, the more complex carry (c_i), pseudo-carry (h_i), group generate ($G_{i,j}$), group propagate ($P_{i,j}$, $T_{i,j}$), group pseudo-carry ($H_{i,j}$), and sum (s_i) signals are calculated. The switching activity of the signals is calculated using eq. 2.

Using the recurrence schemes of Weinberger and Ling, many adders have been proposed in literature. They use different combinations of the group generate and propagate signals to create carry/pseudo-carry signals. The probability formulae given in Table II can be used to find the signal probabilities of any adder using Weinberger or Ling recurrence. Table III shows the probabilities of the signals in the carry block of a Kogge-Stone adder with Weinberger and Ling recurrence.

Note that, generate signals that are at the same level, either Weinberger or Ling, are all equal. This also holds for propagate signals. Note also that, group-generate and group pseudo-carry signals are not equal. That is, one recurrence scheme may be preferable to the other because of having less overall switching activity. Preference is expected to change depending on the input probability.

IV. RESULTS

The switching activities are calculated by use of the formulae given in Table II and III. The average switching activities for each addition algorithms over different input probabilities are given in Figure 3. The results are calculated by averaging switching activities at all nodes. It shows that, Weinberger algorithm has lower average switching activity in region-I and Ling has lower switching factor at region-II. Therefore, it is expected to get lower energy consumption from Weinberger algorithm in region-I and from Ling algorithm in region-II under the same operating conditions. In region-I, the inputs are dominated with logic low and in region-II, they are dominated with logic high.

64-bit Kogge-Stone adder with Weinberger and Ling addition algorithms are implemented in 45nm technology operating at 1V under typical process conditions at a temperature of 25^oC. Both circuits are sized for optimum delay in MATLAB using Logical effort theory [1]. Both implementations have the same input/output loading constraints. To estimate the wire capacitance, a bit-pitch of 2 microns is used.

The HSpice results for energy consumption of each implementation are given in Figure-4. Energy consumption of each design is measured by applying input vectors with fixed input probabilities and then averaging them to obtain the average energy consumption of the design with given input probability. This process is repeated for different input probabilities. As seen in Figure-4, Weinberger saves 16% energy when the

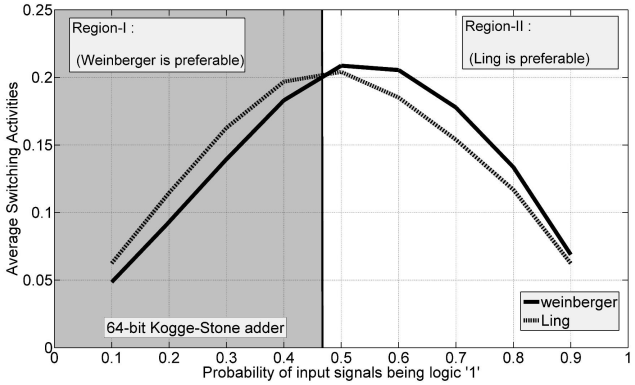

Fig. 3. Average switching activities

input signals probability of being logic high is 0.3(region-I) as compared to Ling algorithm. Similarly, in region-II, Ling saves 3.5% energy when the input signals probability is 0.7. Those results are consistent with the expected results from the average switching activities. This is not an exact match and the differences come from the fact that Ling addition algorithm has more complex sum path and total number of internal nodes for Ling algorithm is higher than Weinberger. The delays of the both implementations are equal to 207ps.

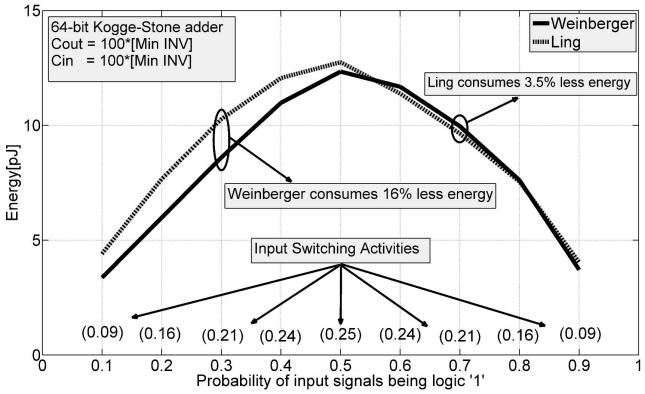

Fig. 4. Energy consumption of Kogge-Stone adders with Weinberger and Ling addition algorithms

V. CONCLUSION

The energy consumption of VLSI adders directly depends on the switching factors of internal nodes. The exact calculation of switching activities is provided under the assumption of spatially and temporally uncorrelated inputs. The selection of energy efficient addition algorithm depends on the statistical nature of the input signals. By looking at the average switching activities of addition algorithms, it is possible to prefer one to other and the accuracy of this selection method is proved by HSpice simulations. Weinberger algorithm can provide up to %20 energy saving as compared Ling in region-I (Input signals are dominated with logic low). Similarly, Ling algorithm can save energy up to %4 in region-II (Input signals are dominated with logic high). The number of nodes for Ling algorithm is higher than Weinberger and this decreases the possible power savings of Ling algorithms in terms of average switching activity.

VI. ACKNOWLEDGEMENTS

This work has been supported by SRC Research Grant No. 1799.001, Intel Corp. and Texas Instruments Corp.

REFERENCES

[1] R. F. Sproull and I. E. Sutherland, *Logical Effort: Designing for speed on the Back of an Envelope*, IEEE Adv. Research in VLSI, C.Sequin(editor),MIT Press, 1991.

[2] V.G.Oklobdzija, B. R. Zeydel, H. Q. Dao, S. Mathew and R. Krishnamurthy, *Energy-delay estimation technique for high-performance microprocessor VLSI adders*, in Proc. 16th Int. Symp. Computer Arithmetic, Santiago de Compostela, Spain, Jun. 2003, pp. 272-279.

[3] B. R. Zeydel, T. Kluter, V. G. Oklobdzija, *Efficient Mapping of Addition Recurrence Algorithms in CMOS*, 17th IEEE Symposium on Computer Arithmetic, Cape Cod, Mass., USA, June 2005.

[4] S. Knowles, *A Family of Adders*, 14th IEEE Symposium on Computer Arithmetic, Adelaide, Australia, April 14th-16th, 1999.

[5] V. G. Oklobdzija, B. R. Zeydel, H. Q. Dao, S. Mathew, R. Krishnamurthy, *Comparison of High-Performance VLSI Adders in Energy-Delay Space*, IEEE Transaction on VLSI Systems, Vol. 13, No. 6, June 2005, pp. 754-758.

[6] A. Weinberger, J.L. Smith *A Logic for High-Speed Addition*, Nat. Bur. Stand. Circ., 591:3-12, 1958.

[7] H. Ling, *High-Speed Binary Adder*, IBM Journal of Research and Development, vol. 25, no.3, pp. 156-166, May 1981.

[8] R. W. Doran, *Variants of an Improved Carry Look-Ahead Adder*, IEEE Transactions on Computers, Vol. 37, No.9, Sept. 1988.

[9] P.M. Kogge and H.S. Stone, *A parallel algorithm for the efficient solution of a general class of recurrence equations*, IEEE Trans. Computers Vol. C-22, No. 8, Aug. 1973, pp.786-793.

[10] A. Ghosh, S. Devadas, K. Keutzer, and J. White, *Estimation of average switching activity in combinational and sequential circuits*, in Proc. ACM/IEEE Design Automation Conf., June 1992, pp. 253259.