Switching Activity Calculation of VLSI Adders

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*—Using exact switching activity rates at all internal nodes when calculating energy of digital circuits is believed to result in improved accuracy over to the use of average switching activity. We compare the two approaches in the case of the Kogge-Stone adder implemented with Weinberger and Ling addition recurrences. The difference between the two is less than 4%. Further we examined the accuracy of the energy/delay estimation technique when using exact and average switching activities in 65nm, 45nm, 32nm and 22nm technology nodes. Even then the worse case error in estimating energy is under 15% at 22nm technology node for 64-bit Kogge-Stone adder. The error in delay estimation is less than 6% for all the nodes. Our finding is that using average switching activity does not yield large errors while simplifying the estimation process greatly.**

I. Introduction

Energy efficient design is important for highly utilized functional units, especially for the adders which are often found to be in the hot-spot of VLSI chip where energy efficient design becomes the first concern [1], [2]. 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 [7] and the selection of the best adder architecture at the beginning of the design reduces the design effort/time.

In order to compare different designs, accurate delay and energy estimations are required. Energy-Delay estimation technique is used to estimate the delay and energy [5], [8]. As a typical scenario in digital circuits, a block of logic is shown in Figure 1. The delay of driver gate can be estimated using logical effort [4] as in given in Eq. 1. Similar to the delay estimation, energy consumption of the driver gate is approximated using the formula as given in Eq. 2 [5].

$$D = \{g_{Driver}\left(\frac{W_{L1}+...+W_{LN}}{W_G}\right) + p_{Driver}\}\tau \qquad (1)$$

$$E = \{E_p W_G \ + E_g(W_{L1}+...+W_{LN})\ \}\alpha_{0->1} + E_{lk} \qquad (2)$$

where g_{Driver} is the logical effort of the driver gate, p_{Driver} is the parasitic of the driver gate, τ is the time constant of technology. E_p(fJ/μm) is the energy factor for internal parasitic capacitance of driver, $W_G(\mu)$ is the size of driver gate, E_g(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.

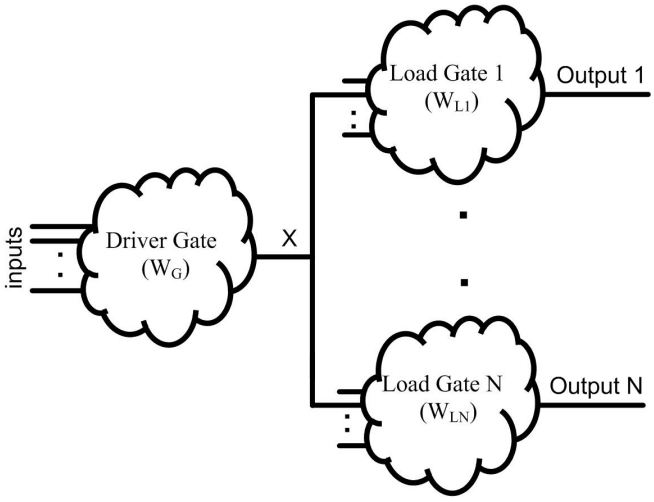

Fig. 1. Chain of Gates

Accurate estimation of delay and energy requires technology characterization. In this step, estimation parameters (g_{Drive}, p_{Drive}, E_g and E_p) are extracted using a proper characterization setup. By following that procedure, it is possible to reflect the changes introduced by the new technology. In addition, switching activity rates ($\alpha_{0->1}$) at every nodes of the circuit are required to estimate the energy as shown in Eq. 2. Dynamic and leakage energy consumption of digital circuits are directly proportional to the total size of logic gates. In addition, the same supply and threshold voltages are used for all designs in comparison.

The paper is organized as follows; section II gives formulas to calculate the switching activities of all nodes for static KS adders as well as switching activity measurement setup. Section III introduces the average switching activity technique for quick energy estimation. The results for energy-delay estimation and Spice are given in section IV and section V concludes the work.

II. Switching Activity Calculation & Measurement

A. Calculation

We calculate the switching activity rates of internal nodes for complex digital circuits under the assumption of temporally and spatially uncorrelated inputs. As a sample case, the internal activity rates of a Kogge-Stone [7] adder implemented

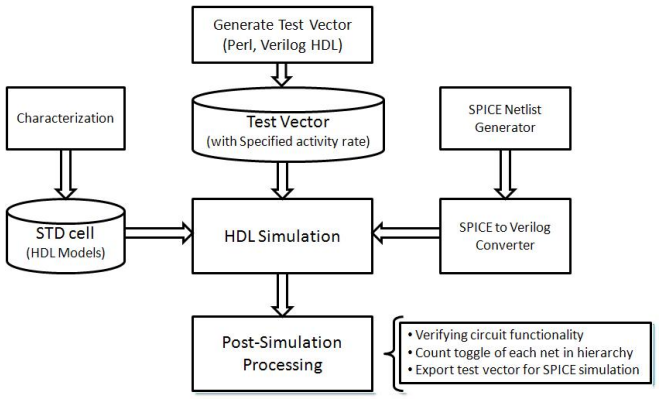

Fig. 2. Measurement Setup of Switching Activity Rates

with Weinberger and Ling addition algorithms are calculated in [3], [12].

The signals generated in Weinberger and Ling recurrence algorithms [6], [9], [10] are given in Table I. The formulas to calculate the probability of being logic '1' of all signals for Weinberger and Ling recurrences are given in Table II. Table III shows the probabilities of the signals in the carry block of a Kogge-Stone adder with both Weinberger and Ling recurrences. For a signal x, let the probability of x being logic '1' ('0') be denoted by $p_1(x)$ ($p_0(x)$). With switching activity of signal x we refer to the event signal x switching from 0 to 1 for which the probability is denoted as $p_{0->1}(x)$. Switching activity can be calculated from the signal probability as:

$$p_{0->1}(\chi) \ = p_0(\chi)\ p_1(\chi) \tag{3}$$

B. HDL Simulations

In order to measure exact value of switching activity at each node, we used gate level HDL simulation. The setup for switching activity measurement is given in Figure 2. Required cell parameters are extracted in cell characterization phase. Activity rate is related to circuit topology and input pattern. Circuit sizing affect delay of each cell which in turn affects the glitch probability. Random test vectors are generated in MATLAB using the required transition probability at each input bit. Total number of random vectors for each case is 64k sample. After simulation we perform post processing step including averaging. Table 4 shows the average switching activity rates from HDL simulations and from calculations. The results are consistent and error is less than 1% for both addition algorithms.

III. AVERAGE SWITCHING ACTIVITY TECHNIQUE

For accurate energy and delay estimation, the switching activity rates of all nodes are required. In secton-2, the methods for calculation/measurement of switching activities of all nodes for Kogge Stone structure are given. However, calculation/measurement of all switching factors is cumbersome as it introduces complex calculation in the comparison method. Thus, there is a need to approximate the switching activity rates of a given circuit in order to estimate its energy consumption in a quick way. To simplify this step we use the same average switching activity rate at all nodes in energy estimation. In order to estimate the error made by using Average Switching Activity (ASA) we compare the two approaches.

The switching activities are calculated by use of the formula 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. Figure 3. shows that, the use of Weinberger recurrence results in lower average switching activity in the region-I, while the use of Ling recurrence results in lower switching factor in the region-II. Therefore, to the input switching probability allow us to choose the recurrence resulting in less switching and lower energy consumption. In region-I, logic low condition is dominant at the inputs, while in region-II, logic high dominates.

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

TABLE IV
COMPARISON OF AVERAGE SWITCHING ACTIVITIES OF STATIC 64-BIT KS(W) AND KS(L) ADDER.

	Weinberger		Ling	
$^1p_1(x)$	Calculation	HDL	Calculation	HDL
0.1	4.84	4.83	6.25	6.24
0.2	9.29	9.27	11.45	11.43
0.3	13.93	13.92	16.25	16.24
0.4	18.30	18.26	19.66	19.64
0.5	20.90	20.84	20.37	20.36
0.6	20.55	20.53	18.46	18.48
0.7	17.77	17.77	15.32	15.37
0.8	13.36	13.37	11.62	11.67
0.9	6.88	6.90	6.20	6.24

$^1p_1(x)$ is the input probability of being logic '1'.
The number of input vectors applied in HDL simulation is 64K for each case.

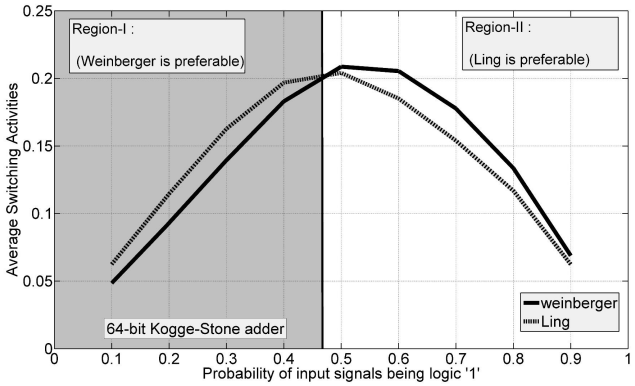

Fig. 3. Average switching activities

When the input low/high probability is 50%, the average switching activity rates reaches their peak value for both addition recurrences and the energy consumption is highest. In order to compare two designs, the average expected input switching activity should be considered (approximately 20% for static CMOS) as 50% input probability leads to the highest energy consumption, as shown in Figure 3.

IV. RESULTS

The results obtained using energy-delay estimation method [5], [6], [8] and HSPICE simulation are compared for 65nm, 45nm, 32nm and 22nm PTM technologies under typical process conditions. In Table V, a summary of technology dependent parameters is given. The optimal sizing of the

circuits is achieved as an iterative process performed by MATLAB. Each data point corresponds to a unique optimal sizing for each adder structures. To estimate the wire length, a bit-pitch given in Table V is used for each technology.

The worst case delay and the average energy consumption are reported for HSpice results. For the energy measurement, random input vectors with 50% probability of being logic high are generated. Those vectors are applied to the adder and the energy results are averaged to calculate the average energy consumption of the adder. Sufficient number of input vectors is applied to reach an average value for each case.

TABLE V
SUMMARY OF TECHNOLOGY PARAMETERS

	65nm	45nm	32nm	22nm
Vdd(V)	1.1	1.0	0.9	0.8
Wmin(μm)	0.13	0.11	0.096	0.066
FO4(pS)	20.85	16.88	14.69	12.04
Bit Pitch(μm)	2	1.69	1.47	1.01
Temp.(^oC)	25	25	25	25

The energy estimation results utilizing the exact switching activity at each specific node and the average switching activity at all internal nodes for 64-bit static KS (W) and KS (L) adders are given in Table VI. The difference between the energy values obtained using exact switching activity and the averaged switching activity is less than 4%. Delay estimation remains the same, because, the estimated delays are the minimum delays which can be obtained from each adder structure under a range of operating conditions (input and output loading).

The energy-delay space of a Kogge-Stone adder implemented with Weinberger recurrence is generated using energy-delay estimation method and HSpice at 65nm, 45nm, 32nm and 22nm technology nodes. The results are given in Figure 4.

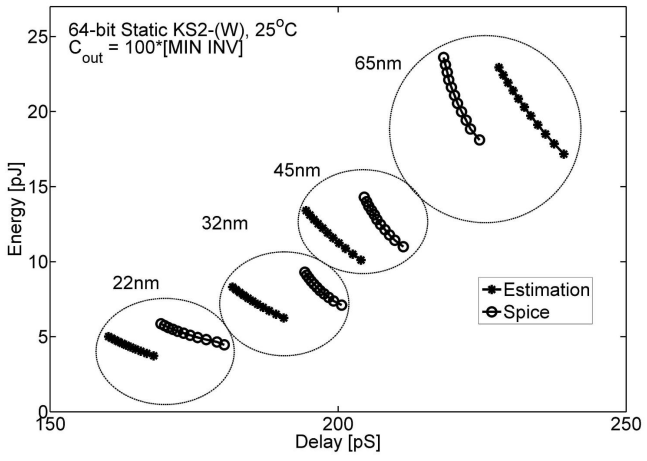

Fig. 4. Estimation versus Spice Simulation

V. CONCLUSION

Using correct switching activity while calculating energy is essential for energy estimation of VLSI adders. The exact calculation/measurement of switching activities of each internal node for static adders is possible for spatially and temporally uncorrelated input vectors. The use of averaged switching activity (ASA) at all internal nodes results in a quick and simple method to estimate internal switching activities. The difference in the energy estimations for exact switching activity case and the averaged switching activity case is less than 4% when the input switching probability is 50%. Our results show that the use of the averaged switching activity (ASA) at all internal nodes yields sufficient accuracy for the purpose of comparing different designs. H-Spice simulations for 64-bit static KS(W) adder and the energy estimation results are sufficiently close to each other and the worst case error between result obtained using H-Spice and energy estimation is less than 6% for delay and less than 15% for energy across all technology nodes.

TABLE VI
ENERGY ESTIMATION WITH EXACT AND AVERAGE SWITCHING ACTIVITY

	KS2-(W)			KS2-(L)		
[1]C_{in}	E(pJ)	E(pJ)		E(pJ)	E(pJ)	
	[2]HDL	20.9%	Err(%)	[2]HDL	20.36%	Err(%)
100	23.11	23.90	3.43	24.44	24.22	0.91
95	22.61	23.38	3.42	23.95	23.71	1.01
90	22.09	22.85	3.45	23.44	23.19	1.07
85	21.56	22.30	3.43	22.92	22.65	1.18
80	21.02	21.74	3.41	22.39	22.11	1.25
75	20.45	21.15	3.44	21.84	21.54	1.37
70	19.87	20.55	3.41	21.18	20.88	1.42
65	19.27	19.92	3.39	20.50	20.19	1.53
60	18.64	19.28	3.42	19.79	19.47	1.60
55	17.99	18.60	3.39	19.04	18.72	1.68
50	17.30	17.89	3.42	18.26	17.94	1.77

[1]Input Capacitance of Minimum Sized Inverter.
[2]HDL Provides Exact Switching Activities.

VI. ACKNOWLEDGEMENTS

This work has been supported by SRC Research Grant No. 2008-HJ-1799, Intel Corp. and Texas Instruments Corp.

REFERENCES

[1] V. G. Oklobdzija, R. K. Krishnamurthy, "High-Performance Energy-Efficient Microprocessor Design", Springer, July 2006.
[2] V. G. Oklobdzija, "High-Performance System Design: Circuits and Logic", Wiley-IEEE Press, July 1999.
[3] 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.
[4] 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.
[5] 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.
[6] 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.
[7] S. Knowles, *A Family of Adders*, 14th IEEE Symposium on Computer Arithmetic, Adelaide, Australia, April 14th-16th, 1999.

[8] 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.

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

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

[11] 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.

[12] D. Baran, M. Aktan, H. Karimiyan, V. G. Oklobdzija, *Exploration of Switching Activity Behavior of Addition Algorithms*, accepted for publication at MWSCAS 2009, to be presented in Cancun, Mexico, 2-5 August 2009.