ACCURATE DATA PATH MODELS FOR RT-LEVEL POWER ESTIMATION[1]

S. Theoharis, G. Theodoridis, D. Soudris,* and C. Goutis

VLSI Design Lab., Dept. of Elect. and Comp. Eng.
University of Patras, Rio 26110, Greece,
theohari@ee.upatras.gr

VLSI Design and Testing Center, Dept. Elect. and Comp. Eng.,
Democritus University of Thrace, Xanthi, 67100, Greece

ABSTRACT.

Power models for RT-Level power estimation of common components, exploiting their functionality, regularity, symmetry, and separability are introduced. The accuracy of the proposed models is higher than the existing ones, as it is identical to that of a real delay gate level power estimator. The power model of *K*-bit Ripple Carry Adder is presented in detail manner. Comparison results prove that the computational complexity of the proposed models is similar or lower than that of the existing ones for the majority of the components.

1. INTRODUCTION.

Power consumption has become a challenging problem of the VLSI design in recent years [1, 2, 3]. Due to the limited battery life, the estimation and reduction of the consumed power is a critical point in the field of designing portable electronic devices. Additionally, issues such as reliability, electromigration, and packaging cost make the power dissipation an important parameter of the integrated circuit design [4]. Efficient solutions of the above problems require the development of new low-power design techniques and power estimation models at all design levels.

A Register-Transfer-Level (RTL) data path is characterised by the use of pre-designed components such as arithmetic components, registers, multiplexers, decoders, ALUs, etc., (the control units, buses, memories, and clock trees are excluded from this data path

[1] work is partially supported by the project "LPGD" 25256, ESPRIT IV.

category). Accurate estimation of the power dissipation of an RTL data path can be obtained by moving the RTL architecture description to a lower-description level, for instance, the gate level, where an accurate measurement can be performed. However, the computational complexity of this approach is enormous, particularly for large data paths. To overcome the above problem, power dissipation models for RTL components have been presented.

Recently, research work in RTL power estimation field have been reported [5, 6, 7]. In particular, a three dimensional Look Up Table (LUT), whose dimensions are the average input static probability, the average input transition density, and the average output transition density, is reported in [5]. The spatial correlation among the inputs is not taken into account. The reported RMS error is 6% approximately, while the maximum error is 20% compared with a gate level simulator. In [6], models, which take into account glitches, for data path and control circuits have been presented. These models use both analytical equations and LUTs and are based on the assumption that power model equations are separable. The reported errors for RTL benchmarks are 7%, approximately. However, the previous models are stochastic and the adopted assumptions, e.g. temporal and spatial correlation, making them inaccurate. Furthermore, randomly-generated input vectors were used in the presented experimental results, where the influence of the assumptions is not evident; input vectors are usually not random e.g. a biased input stream. Hence, the error of each of the above model is more increased than the reported ones. In [7], models for the power evaluation of each input vector pair (cycle by cycle) have been reported. Spatial correlations of the inputs up to three and first-order temporal correlations have been considered. The reported error is about of 10% and comes from the high-order spatiotemporal correlation.

In this paper, efficient power models of frequently-used RTL data path components, which evaluates the power consumption for each input vector pair (cycle by cycle), are presented. The architecture and functionality of each component are exploited, resulting into the development of an accurate and low computational complexity models. The accuracy of the proposed models is identical to that of a real delay gate level power estimator, while the computational complexity is smaller or similar to the majority of the components.

2. STATEMENT OF THE PROBLEM.

The problem of development power dissipation models for data path components at RTL can be stated as follows:

"*Given an N input component and its gate level description, find a low computational complexity model, such that the error of the evaluated power consumption to be minimum for any type of the applied input vector set*".

The published RT-Level models can be classified into two categories; the stochastic or pattern independent models [5, 6], and the non-stochastic or pattern dependent [7,8] ones. Using statistical properties, which describe the whole applied input vector set, e.g. the static and transition probabilities of the input vectors, the stochastic models provide the power consumption. Non-stochastic models provide the power dissipation by manipulating each vector separately. Stochastic models are characterised by low computational complexity but they are inaccurate due to the adopted assumptions e.g. the temporal and spatial correlation among the inputs. On the other hand, pattern dependent models provide higher accuracy, since each vector pair is examined separately without described by inaccurate global stochastic parameters. However, the complexity of the non-stochastic models is larger than the complexity of the stochastic ones.

In addition, both stochastic and non-stochastic models are inaccurate, since they are sensitive to the vector set that have been used during the characterisation phase. Characterisation is the procedure of tuning the model parameters, considering accurate power dissipation measurements of an input vector set, which is called training set. Characterisation is taken place once for each component and after the parameters tuning the same model is used for any input vector set. However, in many application experiments the applied input vector set is different than the training set, making the evaluated power consumption inaccurate. To solve the above problem, the evaluated power values are "*corrected*" using gate level power estimators and regression techniques. These RTL models are called adaptive models [9, 10].

Performing exhaustive simulation and storing the power dissipation values in a LUT, an accurate and low complexity power model can be achieved. The accuracy of the above model comes from the fact that it is based on simulation without any assumption as the stochastic models. In addition, the power values of the LUT correspond to any input

vector combination making the model accurate for any type of the applied input vector set; the model becomes insensitive to the training set. However, the size of the LUT is exponentially increased as the input number of the component increases resulting into inefficient model in terms of memory space. The size of the LUT is equal to 2^{2N} entries, where N denotes component inputs. Additionally, the effort of filling the entries of the LUT, despite that it takes place once for each component, is also increased prohibitively.

3. PROPOSED METHOD

The most frequently used RTL data paths components are arithmetic components (adders, substractors, multipliers), multiplexers, comparators, registers, Multiply Accumulate Units, and ALUs. The architecture of each component, includes a uniformly-repeated primitive cell, e.g. 1-bit full adder, flip-flops, and is characterised by regularity, symmetry, and frequently, by separability. Exploiting these properties, which have not properly been used in previously reported models, an accurate and low complexity power model for each component with N inputs and K primitive cells, can be developed. Also, it is assumed that the input vectors are applied on an RTL component in parallel fashion and the component is stabilised before the loading of the next input vector.

The proposed method performs RTL power estimation using a set of small LUTs. Instead of constructing one large LUT for the whole component, employing the concept of "divide and conquer", the memory size can be reduced. Two kinds of LUTs are: i) the *primary* LUT, and ii) the *secondary* LUT (or LUTs). In particular, the primary LUT corresponds to the power dissipation of a block of L $(1 \le L \le K)$ primitive cells. It contains the power dissipation values for any input vector pair as well as any additional useful information such as the transitions number of the interconnection signals among the blocks of the component, the steady values of the component's outputs that correspond to any block e.t.c. It must be stressed that the additional information of the primary LUT depends on the functionality and architecture of the considered component. The secondary LUT(s) is(are) used to reduce the computational complexity of the proposed power model of those RTL components, which have complex architecture and functionality. The power values of the LUTs are expressed in terms of switching capacitance, derived by a real delay gate level simulator. Thus, the power consumption of the component can be evaluated based on the LUTs with a small amount of functional simulation.

Performing a small amount of functional simulation any useful information is transferred from one block to another and the power consumption is evaluated by accessing either the primary LUT or the secondary LUTs.

Since the power values of the two LUTs are calculated by a real delay gate level power estimator and the glitches from one block to next one are captured by performing functional simulation, the accuracy of the proposed model is identical to the accuracy of a power estimator based on a gate level simulator. Moreover, the model is not sensitive to the training set, since the primary LUT contains the power dissipation values that correspond to all possible input combinations. Finally, there is a trade off between the computational time and the size of the LUTs depending on the length *L* of the used block.

To estimate the power consumption of a composite RTL data path component three steps are needed: i) the decomposition of the composite RTL data path component into its basic components, ii) the estimation of the power consumption of each basic component, and iii) the superposition (or addition) of the individual power values.

The power model of a *K*-bit Ripple Carry Adder (RCA) is presented below, while the power models for the remaining basic RTL data paths have been developed in [12].

4. THE PROPOSED MODEL OF *K*-BIT RIPPLE CARRY ADDER

The addition operation is performed in two phases: i) the *generate* phase, and ii) the *propagate* phase. During the *generate* phase the input operands are applied in parallel and the carries are generated. In the latter phase, the carries are propagated through the carry propagate chains, while the input operands are constant. Consequently, the power dissipation can be evaluated in two phases separately, applying the principle of superposition. In other words, the total power dissipation, P_{tot}, can be obtained by:

$$P_{tot} = P_{gen} + P_{prop} \quad (1)$$

where, P_{gen} and P_{prop} are the power dissipations derived from the propagate and generate phases, respectively.

The architecture of an RCA is characterized by regularity, symmetry, and separability, since the 1-bit full adder cell is repeated-uniformlly. [11]. Considering the above properties, two LUTs are needed to evaluate the power dissipation of one *K*-bit

RCA. The K-bit RCA is decomposed to $B = \left\lceil \frac{K}{L} \right\rceil$ $(1 \le L \le K)$ blocks, where each of them has $2L$ input operands, one input carry, L sum signals, and one output carry.

The primary LUT corresponds to a L-bit adder and contains: i) the power dissipation values for any input vector pair, ii) the length M $(0 \le M \le L)$ of the carry propagate chain, whose first element is the first cell of the L-bit RCA, iii) the number of the output carry transitions of the L-bit RCA, and iv) the steady state values of the output carry and sum signals.

The secondary LUT calculates the power dissipation during the propagate phase. Specifically, the carry propagates through an 1-bit adder cell when the input operands are "10"or "01" and ceases the propagation when the input operands are either "11" or "00". Thus, it contains two power values, $P_{prop\_\sec}$, and $P_{hold\_\sec}$, which are the power dissipations when a carry is propagated or blocked at the 1-bit adder cell, respectively.

The power dissipation of the *generate* phase, P_{gen}, can be evaluated by accessing the primary LUT B times, taking the applied input vector and the previous state of the carry signals into account. The power consumption of the i-th, i=1,…, B, block during at the *propagate* phase, P_{prop_i}, is equal to:

$$P_{prop_i} = \begin{cases} M_i \times T_{i\_in} \times P_{prop\_\sec} & if \;\; M = L \\ M_i \times T_{i\_in} \times P_{prop\_\sec} + T_{i\_in} \times P_{hold\_\sec} & if \; M < L \end{cases} \qquad (2)$$

where, M_i is the length of the propagate chain of the block and T_{i_in} is the number of the block input carry transitions. Thus, the total power consumption is:

$$P_{prop} = \sum_{i=1}^{B} P_{prop_i} \qquad (3)$$

Carry propagates from one block to the next one when the first primitive cell of the next block allows the carry propagation. Consequently, among all possible propagate chains of each block, only that whose the first cell is the first cell of the block is considered to evaluate the power dissipation during the propagate phase. That is, it corresponds to a chain of length M_i.

The number of input carry transitions, T_{i_in}, of i-th block is equal to:

$$T_{i\_in} = T_{i-1\_out} \quad (4)$$

where, T_{i-1_out}, are the transitions of the (i-1)-th block. The output transitions of i-th block is given by:

$$T_{i\_out} = \begin{cases} T_{i\_prim} & if\ M_i < L \\ T_{i\_prim} + T_{i\_in} & if\ M_i = L \end{cases} \quad (5)$$

where, T_{i_prim} are the output carry transitions due to the input vector application, which are provided by the primary LUT.

Applying the proposed method, the number, I_1, of computer instructions required for the evaluation of the power consumption of a K bit adder equals to, $I_1 \cong 17 \times B \times V$, where V is the number of the input vectors [12]. The size of the primary LUT is $2^{2(2L)+1} \times 4$ bytes, while the size of the secondary LUT is 2×4 bytes [12].

5. COMPARISONS

Applying the proposed method, power models for the adopted set of common components have been developed. Using the real delay gate level simulator QUICKSIM II of Mentor Graphics, the accuracy of the model has been experimentally verified [12]. In [5], the required computer instructions, I_5, for the power evaluation of a component having N inputs and M outputs are $I_5 \cong (10N + 7M) \times V$. In [6], the word dependent components require $I_6 \cong 20 \times V$ instructions, while the bit-sliced components (MUX, register) are addressed with different approach. For example, the 2-to-1 K-bit multiplexer requires $8 \times K \times V$ instructions while the K-bit register $(4 \times K + 4) \times V$ instructions. In [7], the required instructions, I_7, are $I_7 \cong (7 \times C + 19 \times D + 55 \times E) \times V$, where, C, D, and E are the macro-model parameters [7].

Table 1 shows the required number of computer instructions for the power consumption evaluation for the adopted common RT Level components, applying the proposed method. For the adder/substructor (ADD/SUB), comparator (COMP), register (REG), Wallace multiplier (MUL) [11], Multiply Accumulate Unit (MAC) and Arithmetic Logic Unit (ALU) the term W expresses the word length of each input. The multiplexer (MUX) is a 2–to-1 multiplexer and the term W expresses the word length of the two

multiplexed words, while in the decoder/demultiplexer (DEC/DEMUX) the term W expresses the number of the 1-bit outputs.

The complexity of the proposed models is always smaller than the ones of [5, 6, 7], in case of MUX, DEC/DEMUX, and REG, as it is shown in Table 1 (it is assumed L=3). Also, the complexity of ADD/SUB, COMP, and ALU components is smaller than the ones of [5, 7], while the complexity of a MAC is little greater compared to [5,7]. The complexity of ADD/SUB, COMP, ALU, and MAC is a little higher than [6], when $W \leq 32$. In the case of MUL, the complexity of the proposed model is similar to those of [5, 7] but it is higher than [6], when $W \geq 8$. The values of C=8, D=8, and E=2, which were used in the experimental results of [7], are also used here for comparisons.

The required memory for all the presented components is 600 Kbytes, approximately. By increasing the block length L the computational complexity is reduced while the memory size increases. Table 2 presents the required number of computer instructions for each of the adopted components when L=4. In this case, the required memory for all the presented components is 5 Mbytes, approximately. Also, comparisons of the computational complexity of each component for L=3 and L=4 are presented in Table 2.

However, due to the regularity and symmetry of each component a lot of power values of its primary LUT are same. Therefore, the primary LUT can be compressed reducing its size allowing blocks with length up to 10 to be considered. Therefore, the computational complexity is reduced drastically. However, an efficient addressing scheme has to be found to access the primary LUT. It is expected that the complexity will be reduced to 1.5 to 2 times approximately. We are studying this issue.

6 CONCLUSIONS.

An accurate method for RTL power estimation using a set of LUTs was proposed. The power model of a K-bit RCA was described in detail manner. Generally, the proposed model is based on the exploitation of any information coming from the architecture and functionality of a RTL data path, which is considered as a white open box. In contrast, the models of [5, 6, 7] studied each component as a black box. Moreover, the proposed model is accurate since it has simulative nature, without any dependence on the characterisation training set while its computational complexity is smaller or similar to models of [5, 6], and [7] for the majority of the components.

Component		Word Length (W)				
		4	**8**	**16**	**32**	**64**
ADD / SUB	**P_3**	34	51	102	187	374
	[5]	3,18	4,24	4,24	4,62	4,62
	[6]	-1,70	-2,55	-5,10	-9,35	-18,70
	[7]	11,12	7,41	4,29	2,98	2,13
COMP	**P_3**	24	32	72	128	272
	[5]	3,92	5,44	4,64	5,11	4,76
	[6]	-1,20	-1,60	-3,60	-6,40	-13,60
	[7]	15,75	11,81	6,08	4,36	2,93
MUX	**P_3**	16	24	48	88	176
	[5]	7,38	9,42	9,21	9,93	9,88
	[6]	2,00	2,67	2,67	2,91	2,91
	[7]	23,63	15,75	9,13	6,34	4,53
DEC / DEMUX	**P_3**	8	8	16	16	16
	[5]	7,25	12,00	10,13	17,75	32,38
	[6]	2,50	2,50	1,25	1,25	1,25
	[7]	47,25	47,25	27,38	34,88	49,88
ALU	**P_3**	90	139	286	531	1078
	[5]	2,10	2,14	1,79	1,78	1,68
	[6]	-3,21	-4,96	-10,21	-18,96	-38,50
	[7]	4,20	2,72	1,53	1,05	-1,35
MUL	**P_3**	77	218	693	2410	8917
	[5]	1,77	1,25	-1,27	-2,22	-4,10
	[6]	-3,85	-10,90	-34,65	-120,50	-445,85
	[7]	4,91	1,73	-1,58	-4,32	-11,17
MAC	**P_3**	306	500	1082	3012	9946
	[5]	-1,49	-1,46	-1,76	-2,60	-4,43
	[6]	-3,06	-4,31	-7,31	-14,21	-29,25
	[7]	1,24	-1,32	-2,47	-5,40	-12,46
REG	**P_3**	16	24	48	88	176
	[5]	4,25	5,67	5,67	6,18	6,18
	[6]	1,25	1,50	1,42	1,50	1,48
	[7]	23,63	15,75	9,13	6,34	4,53

Table 1. Comparison results for L = 3

P_3 : Number of instructions of the proposed model per vector pair.
+/- : Number of times the proposed model is faster/slower than that of [5], [6], and [7].

REFERENCES

[1] W. Nebel, and J. MERMET, "*Low Power Design in Deep Submicron Electronics*", Kluwer Academic Publishers, NATO ASI Series, Netherlands, 1997.

[2] J. Rabey, and M. Pedram, "*Low Power Design Methodologies*", Kluwer Academic Publishers, 1996.

[3] A. Chandrakasan, and W. R. Brodersen, "*Low Power Digital CMOS Design*", Kluwer Academic Publishers, 1995.
[4] H.B. Bakoglu, "*Circuits, Interconnections, and Packaging for VLSI*", Addison Wesley, Menlo Park, CA, 1990.
[5] S. Gupta, and F. Najm, "*Power Macromodeling for High Level Power Estimation*", in Proc. of Design Automation Conf., 1997, pp. 365-370.
[6] A. Raghuanathan, S. Dey, and N. Jha, "*Register-Transfer Level Estimation for Switching Activity and Power Consumption*", in Proc. of ICCAD, 1996, pp.158-165.
[7] Q. Qiu, Q. Wu, M. Pedram, and C-S. Ding, *"Cycle-Accurate Macro-Models for RT-level Power Analysis*", Proc. of Int. Sym. of Low Power Design, 1997, pp. 125-130.
[8] H. Mehta, R.M. Owens, and M.J. Irwin, "*Energy characterisation based on clustering*", in Proc. of Design Automation Conference, 1996, pp. 702-707.
[9] C-T. Hsieh, C-S. Ding, Q. Wu, and M. Pedram "*Statistical sampling and regression estimation in power macro-modelling*", in Proc. of ICCAD, 1996, pp. 583-588.
[10] A. Bigliolo, L. Benini, and G. De Micheli, "*Adaptive least mean square behavioural power model*", in Proc. of Europ. Design and Test Conf., 1997.
[11] Koren I., "*Computer Arithmetic Algorithms*", Prentice Hall, 1993.
[12] S. Theoharis, G. Theodoridis, D. Soudris, and C. Goutis, "*Accurate RTL Power stimation Approach*", Technical report, UP/EE/M3/98, University of Patras, 1998.

Component		**Word length (*W*)**				
		4	8	16	32	64
ADD / SUB	$\mathbf{P_4}$	17	34	68	136	272
	$\mathbf{R_{4/3}}$	2,00	1,50	1,50	1,38	1,38
COMP	$\mathbf{P_4}$	8	24	40	88	168
	$\mathbf{R_{4/3}}$	3,00	1,33	1,80	1,45	1,62
MUX	$\mathbf{P_4}$	8	16	32	64	128
	$\mathbf{R_{4/3}}$	2,00	1,50	1,50	1,38	1,38
DEC / DEMUX	$\mathbf{P_4}$	8	8	8	16	16
	$\mathbf{R_{4/3}}$	1,00	1,00	2,00	1,00	1,00
ALU	$\mathbf{P_4}$	49	98	188	376	744
	$\mathbf{R_{4/3}}$	1,84	1,42	1,52	1,41	1,45
MUL	$\mathbf{P_4}$	66	196	648	2320	8736
	$\mathbf{R_{4/3}}$	1,17	1,11	1,07	1,04	1,02
MAC	$\mathbf{P_4}$	247	425	973	2837	9637
	$\mathbf{R_{4/3}}$	1,24	1,18	1,11	1,06	1,03
REG	$\mathbf{P_4}$	8	16	32	64	128
	$\mathbf{R_{4/3}}$	2,00	1,50	1,50	1,38	1,38

Table 2. Comparison results for $L = 4$

P_4 : Number of instructions of the proposed model per vector pair.
$R_{4/3}$: Number of times the proposed model with $L = 4$ is faster than the corresponding one with $L = 3$