Power reduction and technology mapping of digital circuits using AND-Inverter Graphs

R. Mehrotra*, E. Popovici*†, K. L. Man† and M. Schellekens†

Abstract— **The minimisation of power consumption is an important design goal for digital circuits. The switching activity of a CMOS digital circuit significantly contributes to overall power dissipation. By approximating the switching activity of circuit nodes as internal switching probabilities using *AND Inverter Graphs* (AIGs), it is possible to estimate the dynamic power dissipation characteristic of circuits. A technique for minimising the overall sum of switching probabilities is presented. The method is based on local reordering and swapping of nodes in AIG representing the functionality of the circuit to be realised.**

This paper also focuses on the problem of mapping a technology independent circuit to a specific technology, using specific gates and cells (AND cells, inverters and buffers) from a given library, with power as the optimisation metric. The resulting circuit that is obtained by mapping the AIG after switching probability optimisation has shown in simulation reduced power dissipation characteristic without an area penalty.

I. Introduction

In recent years, power dissipation is being given increased weightage in design metrics. Average power dissipation in digital CMOS circuits can be expressed as the sum of three main components $P_{short-circuit}$, $P_{leakage}$ and $P_{dynamic}$. $P_{short-circuit}$ is the power from stacked P and N devices in a CMOS logic gate that are in the ON state simultaneously. $P_{leakage}$ is the power dissipation due to spurious currents in the non-conducting state of the transistor. $P_{dynamic}$ is the dynamic power dissipation, also called the switching power which is the dominant source of power consumption in most of the digital circuits. It is generally represented by the following approximation:

$$P_{dynamic} = \alpha C_l V_{dd}^2 f_{clk} \qquad (1)$$

where α is the switching activity factor (also called transition probability), C_l is the overall capacitance to be charged and discharged in a reference clock cycle, V_{dd} is the supply voltage and f_{clk} is the clock frequency. Here we are dealing with switching activity of circuit nodes approximated as internal switching probabilities using AIGs [1]. Various data structures such as BDDs [2] have been used for switching activity estimation and manipulation. We use AIGs because of two reasons. Firstly they have much compact representation of the same circuit than BDDs. Secondly it is difficult to incorporate delay parameter in BDDs which restricts them to power optimisation only. However AIGs allow to map delay parameter along with the switching probability. Hence power and timing analysis both is possible using such data structures [3].

The switching probability estimation on each of the AIG node is based on the technique used in [2]. For this estimation, we have assumed circuit input signals to be statistically uncorrelated and completely independent. We also assume a zero delay model, that is all gate output evaluations occur instantaneously. In AIGs, each AND node is represented as shown in Fig. 1. Here probability of a node c denoted as $P(c)$ is the probability that c has a value of '1' at some arbitrary time of observation. Consider a node c having the probability $P(c)$ and probabilities $P(a)$ and $P(b)$ for the corresponding inputs a and b such that $c = a.b$. We seek the switching probability $P_{sw}(c)$ of c. Switching occurs if and only if the value of c changes from '0' to '1' or '1' to '0'. Probability that c is '1' valued is when both inputs are '1'.

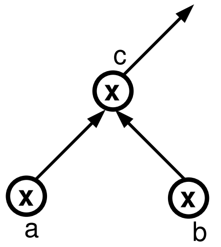

Fig. 1. Switching probability on AIG node.

$$P(c) = P(a)P(b)$$

Probability that c is '0' valued is when either one or both inputs are '0'.

$$\begin{aligned} P(c') &= (1 - P(a))P(b) + P(a)(1 - P(b)) \\ &\quad + (1 - P(a))(1 - P(b)) \end{aligned}$$

R. Mehrotra* is with the Department of Microelectronic Engineering, University College Cork, Cork, Ireland, E-mail: r.mehrotra@student.ucc.ie

E. Popovici*† is with the Department of Microelectronic Engineering and Centre for Efficiency-Oriented Languages, University College Cork, Cork, Ireland, E-mail: e.popovici@ucc.ie

K. L. Man† is with the Centre for Efficiency-Oriented Languages, University College Cork, Cork, Ireland, E-mail: klm1@cs.ucc.ie

M. Schellekens† is with the Centre for Efficiency-Oriented Languages, University College Cork, Cork, Ireland, E-mail: m.schellekens@cs.ucc.ie

978-1-4244-7201-7/10/$26.00 © 2010 IEEE

Based on the technique mentioned in [2], consider the value of c at two different observation times c^{t1} and c^{t2}. If the value of c at those instants is not same, c is considered to switch. Hence the switching probability P_{sw}(c) of c can be estimated as

$$\begin{aligned} P_{sw}(c) &= P((c^{t1}=0) \cap (c^{t2}=1)) \\ &\quad +P((c^{t1}=1) \cap (c^{t2}=0)) \end{aligned}$$

$$P_{sw}(c) = 2P(a)P(b)(1-P(a)P(b)) \tag{2}$$

Using eq (2). we can calculate switching probability on each of the AIG node if we know the probabilities of the primary inputs of the AIG. Next, we applied some local reordering and swapping rules on AIG nodes so as to reduce the sum of the switching probabilities of all the AIG nodes. We build a tool called RESWITCH for this task, implemented as a sub-package in ABC [1]. However most of the time it is difficult to predict if any of the reduction seen at the technology independent level will hold up after final technology mapping step. The technology mapping of an AIG network to an exact netlist with exact mapping of switching probability is presented here using a tool AIG2NET which we implemented as a sub-package in ABC.

A. Organisation

The remainder of the paper is organised as follows. Section 2 explains the rewriting rules and the background behind the RESWITCH tool. Section 3 presents some results on MCNC Benchmark circuits exhibiting the reduction in switching probability on the application of the rewriting rules. Section 4 presents the tool AIG2Net which is used for the technology mapping of digital circuits here using AIGs. Finally concluding remarks and proposal for future work are given in Section 5.

II. Background behind the RESWITCH tool

The RESWITCH tool is based on three rewriting rules R1,R2 and R3 which help us reduce the over all switching probability.

The rule R1 is explained in Fig. 2. According to the Fig. 2, consider two consecutive nodes named d and e such that d is the fanin of e. There shall be no complemented edge between d and e and node d shall not have any other fanouts except e. At this stage if we swap node c with either of the nodes a or b, the functionality of node e will not change and since node d does not have any other fanouts, the swapping will not effect the over all AIG network. Due to the swapping of nodes, the switching probability at node d might decrease or increase. For this, we have made cases within the tool to direct the swapping of a particular node only when the switching probability is getting reduced. Consider the switching probability at node d before swapping is $P_{sw/present}$(d), switching probability at node d on swapping a and c is $P_{sw/ac}$(d) and switching probability at node d on swapping b and c is $P_{sw/bc}$(d).

If ($P_{sw/present}$(d) is less than $P_{sw/ac}$(d)) and ($P_{sw/present}$(d) is less than $P_{sw/bc}$(d)), no swapping is required.

If ($P_{sw/ac}$(d) is less than $P_{sw/present}$(d)) and ($P_{sw/ac}$(d) is less than $P_{sw/bc}$(d)), swap nodes a and c.

If ($P_{sw/bc}$(d) is less than $P_{sw/present}$(d)) and ($P_{sw/bc}$(d) is less than $P_{sw/ac}$(d)), swap nodes b and c.

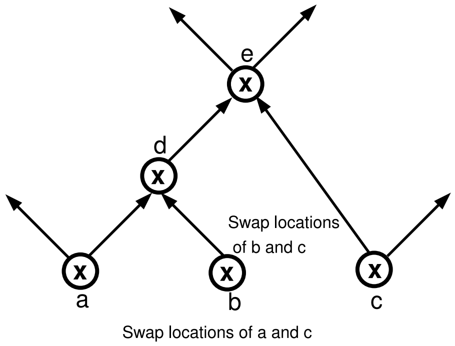

Fig. 2. Rule R1 on AIG

The rule R2 is explained in Fig. 3. According to the Fig. 3, in rule R2 while doing rule R1, consider we are swapping b and c such that a and c are grouped. Suppose there exists an AIG node with fanins a and c with the same complemented attributes as they have in nodes d and e respectively. Hence according to rule R2, while swapping such nodes, redundant nodes are created with same fanins a and c. Such nodes can be removed from the AIG network leaving one in the graph. Such swapping of nodes will not only reduce the over all switching probability of the AIG network but will also reduce the number of nodes.

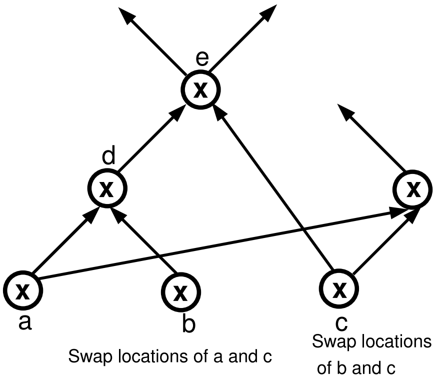

Fig. 3. Rule R2 on AIG

The rule R3 is explained in Fig. 4. According to the

Fig. 4, consider nodes e and f as the fanins of g with no complemented attributes with g, secondly e and f shall not have any other fanouts except g. In such a case, we can swap either of the nodes a and b with either of the nodes c and d. This swapping will give new option of nodes which may either decrease the switching probability or may create redundant nodes. In the Fig. 4, if we swap nodes b and c, b and d nodes are then grouped together. This swapping will create redundant nodes which can be removed from the graph. On the other hand, consider $P_{sw/present}$(e) and $P_{sw/present}$(f) as the switching probability at node e and f respectively before swapping. Consider $P_{sw/ac}$(e) and $P_{sw/ac}$(f) as the switching probability at node e and f respectively after swapping nodes a and c. There can be a case when ($P_{sw/present}$(e) + $P_{sw/present}$(f)) is greater than ($P_{sw/ac}$(e) + $P_{sw/ac}$(f)). In that case, switching probability is getting reduced. Hence rule R3 actually combines the options available by rules R1 and R2.

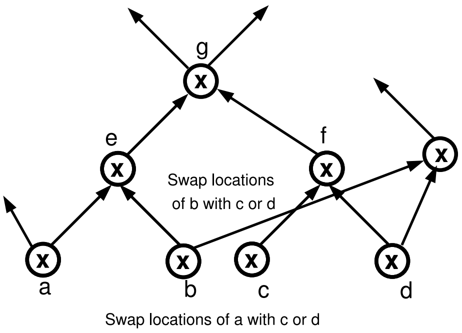

Fig. 4. Rule R3 on AIG.

These three rules are applied on each of the AIG node. If according to the conditions, all three rules are possible on a given node, then the rule which gives maximum reduction of switching probability is chosen. Same applies when two rules are possible on a given node. The algorithm applied here is a Greedy Algorithm [4] i.e. the rules are applied recursively until we get the minimum sum of switching probability. The optimisations proposed here are local and more emphasis will be on global optimisation methods in future work.

III. Experimental Results

We applied the RESWITCH tool on a set of MCNC Benchmark circuits. BLIF file [5] of such circuit is taken and converted to an AIG network in ABC. The AIG network is then synthesised using the synthesis script *resyn* in ABC. In this script, the series of synthesis commands are *balancing*, *rewriting*, *rewriting* with zero cost replacement, *balancing*, *rewriting* with zero cost replacement and *balancing* in the end. These synthesis commands are used for the reduction of nodes and complexity. After the application of synthesis script, switching probability is estimated on each of the AIG node as mentioned in Section 1 and RESWITCH tool is applied. Table I presents the over all decrease in the sum of the switching probability of the nodes. It also specifies the decrease in the number of nodes of the AIG network. In the table, SP_a and SP_b stands for switching probability before and after the application of RESWITCH . NC_a and NC_b stands for node count before and after RESWITCH application. SP_d is the decrease in switching probability in percentage. The overall decrease in the switching probability is 14.90%.

TABLE I
SP without and with the application of rewriting rules R1,R2 and R3

Circuit	SP_b	SP_a	NC_b	NC_a	SP_d
b1	2.56	2.32	10	10	9.37%
cc	12.67	10.47	51	44	17.36%
cm162a	6.57	5.73	33	32	12.78%
mux	15.40	13.24	81	81	14.02%
alu2	63.41	55.75	361	355	12%
alu4	117.97	101.537	657	652	14%
x2	7.53	6.30	45	44	16.33%
cmb	5.86	4.43	37	37	23.37%

IV. Background behind the tool AIG2Net

Technology mapping refers to the final step of the logic synthesis and optimisation task. In technology mapping, library gates are selected to realise the design that has been structurally optimised, here by the three rewriting rules. The AIG network is obtained in ABC using a BLIF file of MCNC Benchmark circuits. This network is converted to a netlist using AIG2Net. The netlist is actually a technology mapped Verilog which can be read and compiled in design compiler DC [6]. In this translation, each AND node of the network is converted to AND cell from a specific library. The complemented edges are replaced with inverter cells and non complemented edges are replaced with buffer cells. Even the name mapping of AIG nodes and edges to the netlist cells and wires is kept in mind here. In this way exact technology mapping of the AIG network to a netlist is possible. Fig. 5 shows a part of the mapped Verilog netlist of the AIG network obtained from AIG2Net.

After the application of rewriting rules and technology mapping, we have used tool DC from Synopsys to report the power. For the dynamic power dissipation,

```
  .......................
wire n17;
AN2D0HVT U17 (.A1(a), .A2(g), .Z(n17));
wire n18;
AN2D0HVT U18 (.A1(a), .A2(c), .Z(n18));
wire n5;
INVD0HVT U5z (.I(e), .ZN(n5));
  .............................
```

Fig. 5. Mapped Verilog netlist of an AIG network

we provide a user defined switching activity file SAIF [7]. The SAIF files are used and generated by various commercial tools. These files define the switching activity of all the primary inputs, primary outputs and internal cells of the netlist. A similar user defined SAIF file is created for the circuits specifying the switching activity approximated as switching probabilities of all the AND nodes (replaced by AND cells). Such SAIF file will help us map the switching probabilities as calculated in Section 1 on the netlist for exact dynamic power estimation. Fig. 6 shows a part of the user defined SAIF file given as an input to the netlist for power calculation.

```
  ............................
set_switching_activity p -static_prob
0.032000 -toggle_rate 0.061952
set_switching_activity n17 -static_prob
0.138807 -toggle_rate 0.239079
set_switching_activity n18 -static_prob
0.138807 -toggle_rate 0.239079
  ..............................
```

Fig. 6. User defined switching activity file for the netlist

TABLE II
Power dissipation with and without the RESWITCH tool

Circuit	$P_{beforeRESWITCH}$	$P_{afterRESWITCH}$
cmb	20.2025	16.2896
b1	8.0618	5.7468
c8	72.6992	66.6414
cm162a	21.2321	17.7750
alu2	204.3606	190.9485
x2	23.7918	20.3919
cc	39.1582	33.8001

Circuits used are MCNC Benchmark circuits which are generally used in literature. Table II presents few examples. Power results are in mW. P stands for power. In the whole process of power optimisation and technology mapping, there is no area penalty. While dealing with the AIG of the circuit, the number of nodes either remains same or gets reduced. Also after technology mapping, there is no area penalty. The formal verification is done using Synopsys Formality.

V. Conclusion

The paper deals with two important problems of the digital design. One is power optimisation and another is technology mapping. For this we implemented tools RESWITCH and AIG2Net for power optimisation and technology mapping respectively. The average reduction in switching probability on applying RESWITCH came out to be 14.9 %. Our tool provides a reduction in switching probability along with no area penalty. In some cases, while using RESWITCH, both power and area are getting optimised, unlike previous methods. We use AIG as the data structure for our work as it gives much compact representation than BDD and it allows both timing and power to be analysed and optimised as a future work. Even the technology mapping tool AIG2Net is fast, accurate and makes exact name mapping of AIG network (nodes and edges) to the netlist (cells and wires) possible. Use of AIGs in our work not only helps us in easy and fast technology mapping for realistic observation but also opens new ideas of delay optimisation in parallel with power optimisation. As a future work, we intend to integrate these tools with more optimisation techniques to aid efficient optimisation of digital circuits.

Acknowledgment

We gratefully acknowledge the support of IDA Ireland and Synopsys in funding our research as indicated in this paper.

References

[1] A. Mishchenko and S. Chatterjee and R. Brayton. “Dag-aware AIG rewriting a fresh look at combinational logic synthesis”. *Proceedings of the 43rd annual conference on Design automation*, pp. 532–535, 2006.

[2] M. T. P. Lindgren, M. Kerttu and R. Drechsler. “Low power optimisation technique for BDD mapped circuits”. *ASP-DAC*, 2001.

[3] R.Mehrotra, K.L.Man, E.Popovici and M.Schellekens. “Data Structure Manipulation for NNIG and PTNNIG: Towards a Unified Power and Timing Analysis”. *3rd International conference on Signals, Circuits and Systems.*.

[4] “Greedy Algorithm”. `http://en.wikipedia.org/wiki/Greedy_algorithm`.

[5] E. Sentovich, K. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. Stephan, R. Brayton, and A. Sangiovanni-Vincentelli. “Sis: A system for sequential circuit synthesis”. *Electronics Research Laboratory*, Memorandum No. UCB/ERL/ M92/41, May 1992.

[6] Synopsys. “Synopsys design platforms”. `http://www.synopsys.com/products/products.html`.

[7] Synopsys SAIF. “Switching Activity Interchange Format”. `http://www.synopsys.com/partners/tapin/saif.html`.