A Kernel-Based Partitioning Algorithm for Low-Power, Low-Area Overhead Circuit Design Using Don't-Care Sets

Ick-Sung Choi , Hyoung Kim, Shin-Il Lim, Sun-Young Hwang, Bhum-Cheol Lee, and Bong-Tae Kim

ABSTRACT— This letter proposes an efficient kernel-based partitioning algorithm for reducing area and power dissipation in combinational circuit designs using don't-care sets. The proposed algorithm decreases power dissipation by partitioning a given circuit using a kernel extracted from the logic. The proposed algorithm also reduces the area overhead by minimizing duplicated gates in the partitioned sub-circuits. The partitioned subcircuits are further optimized utilizing observability don't-care sets. Experimental results for the MCNC benchmarks show that the proposed algorithm synthesizes circuits that on the average consume 22.5% less power and have 12.7% less area than circuits generated by previous algorithms based on a precomputation scheme.

I. INTRODUCTION

The demand for portable communication and computing systems has made power consumption one of the major concerns in VLSI circuit designs. Most power optimization algorithms concentrate on minimizing power dissipation that is due to switching activities at various abstraction levels because switching activities account for over 90% of total power dissipation in digital circuits. These proposed solutions include a precomputation scheme, input transition guide techniques, and circuit partitioning techniques [1]-[3]. The circuit partitioning algorithm transforms a given circuit into multiple subcircuits. In the partitioned subcircuits, one subcircuit is activated by setting the load-enable signal of its input register. These algorithms considerably reduce power consumption, but incur area overhead because they use duplicated gates and input registers.

In this letter, we propose a kernel-based partitioning algorithm for low-power logic circuit design. The proposed algorithm also reduces the area overhead by eliminating duplicated gates in the partitioned subcircuits and by optimizing the partitioned subcircuits by utilizing don't-care sets.

II. PROPOSED KERNEL-BASED PRECOMPUTATION SCHEME

1. The Reduced Ordered Binary Decision Diagram and Kernels

Bryant proposed an efficient way to represent Boolean functions known as a reduced ordered binary decision diagram (ROBDD) [4]. He showed that most logic operations can be performed very efficiently directly on a ROBDD. A circuit can be restructured by utilizing common subexpressions to reduce area and power, and kernels are used for this in logic circuit design.

Manuscript received Aug. 9, 2002; revised Oct. 24, 2002.

Ick-Sung Choi (phone: +82 42 860 1257, e-mail: ischoi@etri.re.kr), Bhum-Cheol Lee (e-mail: bclee@etri.re.kr), and Bong-Tae Kim (e-mail: bkim@etri.re.kr) are with Network Core Technology Department, ETRI, Daejeon, Korea.

Hyoung Kim (e-mail: kho@kyungmin-c.ac.kr) is with the Electrical Engineering Department, Kyungmin University, Uijeongbu city, Gyeonggi-do, Korea.

Shin-Il Lim (e-mail: silim@bukak.seokyeong.ac.kr) is with the Electrical Engineering Department, Seokyeong University, Seoul, Korea.

Sun-Young Hwang (e-mail: hwang@sogang.ac.kr) is with the Electrical Engineering Department, Sogang University, Seoul, Korea.

2. Proposed Kernel-Based Precomputation Scheme

The proposed circuit structure is based on the transformation of a given function f using an extracted kernel as follows.

A function f can be expressed with respect to a kernel K as $f = f_K \cdot K + f_{\overline{K}} \cdot \overline{K}$, where f_K and $f_{\overline{K}}$ are generated by dividing the given function f by kernel K and $\overline{K}$, respectively. This expression can be transformed as shown in (1).

$$\begin{aligned} f &= K \cdot f_K + \overline{K} \cdot f_{\overline{K}} \\ &= K \cdot f_K \cdot (f_{\overline{K}} + \overline{f_{\overline{K}}}) + \overline{K} \cdot f_{\overline{K}} \cdot (f_K + \overline{f_K}) \\ &= K \cdot f_A + \overline{K} \cdot f_B + f_C, \end{aligned} \tag{1}$$

where $f_A = f_K \cdot \overline{f_{\overline{K}}}$, $f_B = \overline{f_K} \cdot f_{\overline{K}}$, and $f_C = f_K \cdot f_{\overline{K}}$.

The circuit realizing the function f consists of the subcircuits realizing the following subfunctions (Fig. 1): an extracted kernel K used as precomputation logic, the functions $f_A = f_K \cdot \overline{f_{\overline{K}}}$ and $f_B = \overline{f_K} \cdot f_{\overline{K}}$, and $f_C = f_K \cdot f_{\overline{K}}$, which is a common subexpression in f_K and $f_{\overline{K}}$. The output of the precomputation logic K is connected to the select line of the output multiplexor, since depending on the value of the kernel K, only one of the subcircuits realizing f_A and f_B is activated. M_0 and M_1 are masking blocks to prevent the two subcircuits from input switching. They can be implemented in simple AND gates or pass transistors [2]. The value of f_C is not affected by the value of the kernel. Even though the transformed function appears to be complicated, it is actually much simpler than the original function when put into practice.

The functions f_A, f_B, and f_C can be further minimized utilizing don't-care sets [5]. The observability don't-care sets for f_A, f_B, and f_C can be computed as in (2), (3), and (4).

$$\begin{aligned} f_A^{DC} &= \mathrm{ODCs}(f)_{f_A} = \frac{\overline{\partial f}}{f_A} = \overline{f_{f_A=1} \oplus f_{f_A=0}} \\ &= \overline{K} + f_C \end{aligned} \tag{2}$$

$$\begin{aligned} f_B^{DC} &= \mathrm{ODCs}(f)_{f_B} = \frac{\overline{\partial f}}{f_B} = \overline{f_{f_B=1} \oplus f_{f_B=0}} \\ &= K + f_C \end{aligned} \tag{3}$$

$$\begin{aligned} f_C^{DC} &= \mathrm{ODCs}(f)_{f_C} = \frac{\overline{\partial f}}{f_C} = \overline{f_{f_C=1} \oplus f_{f_C=0}} \\ &= f - f_C. \end{aligned} \tag{4}$$

The area of the partitioned subcircuits realizing f_A and f_B can be reduced further by using observability don't-care sets f_A^{DC}, f_B^{DC}, and f_C^{DC} in the optimization process.

An example benchmark circuit 'b1' (Fig. 1 (a)) was optimized by SIS 1.2 using script.rugged.

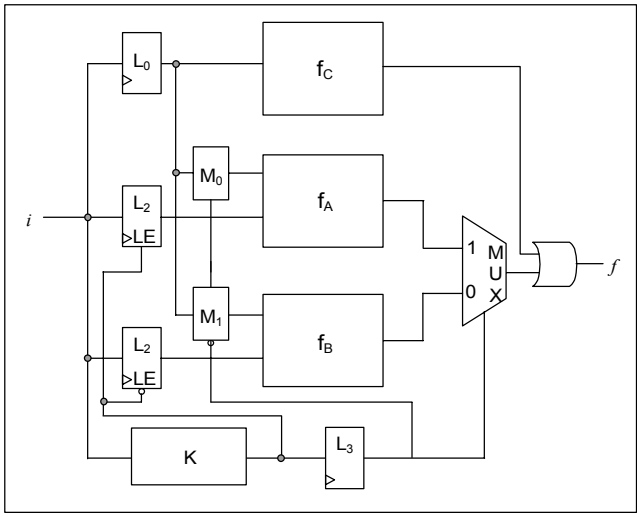

Fig. 1. The proposed precomputataion structure.

If the signal probability of the input variable is 0.5, the power dissipation of this circuit is 45 μW. The circuit obtained by the proposed scheme (Fig. 2(b)) consumes 28.2 μW, showing improved performance. The literal count of the circuit by the proposed scheme is 12, compared to 16 in the circuit generated by SIS. This example shows that the proposed algorithm reduces area.

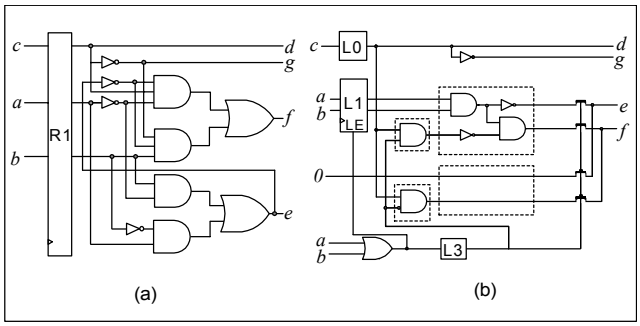

Fig. 2. The circuit generated by the proposed scheme for low-power circuit design: (a) An example circuit 'b1', (b) Circuit obtained by the proposed algorithm.

III. KERNEL-BASED SYNTHESIS ALGORITHM FOR LOW-POWER, LOW-AREA OVERHEAD CIRCUIT DESIGN

The proposed algorithm consists of three steps: BDD extraction, kernel selection, and a circuit synthesis process. The Kernel selection step chooses the best partitioning solution in terms of the cost function out of possible partitioning solutions

Table 1. Comparison of power dissipation.

Bench-mark Circuits	Power (μW)				△Power (%)		
	SIS	Precomp. logic	Shan. expan-sion	Pro-posed	Proposed vs. SIS	Proposed vs. Precomp. logic	Proposed vs. Shan. expansion
b1	45	40	23	28	–37.4%	–29.6%	22.5%
cc	99	182	93	99	0.0%	–45.6%	6.5%
cht	526.4	415	306	526	0.0%	26.8%	72.0%
cm42a	110.7	108	51	106	–4.4%	–2.0%	107.5%
cm82a	101.2	112	69	101.2	0.0%	–9.6%	46.7%
Cm138a	87.2	57	49	58	–33.1%	2.3%	19.0%
Cm162a	267.3	159	87	223	–16.7%	40.1%	156.1%
cmb	228	174	156	214	–6.0%	23.2%	37.4%
cu	234	150	108	175	–25.1%	16.9%	62.4%
duke2	1262.2	1,031	720	691	–45.3%	–33.0%	–4.1%
F51m	715.6	468	348	243	–66.1%	–48.2%	–30.3%
majority	64.4	39	28	30	–54.1%	–24.2%	5.6%
misex2	363.1	433	191	352	–2.9%	–18.6%	84.5%
pcle	238.2	402	91	238	0.0%	–40.7%	161.8%
pcler8	283.2	182	103	283	0.0%	55.6%	175.0%
sao2	482.6	402	417	222	–54.1%	–44.9%	–46.8%
sct	312.4	297	167	456	45.9%	53.5%	173.0%
x2	202.5	232	125	220	8.5%	–5.3%	75.8%
Z4ml	207.5	214	102	161	–22.5%	–24.9%	57.7%
9symml	1058.1	1,312	191	543	–48.7%	–58.6%	184.1%
Average	344	320	171	306	–27.9%	–22.5%	45.1%

Table 2. Comparison of area.

Bench-mark Circuits	Area (#lits)				△ Area (%)		
	SIS	Precomp. logic	Shan. expan-sion	Pro-posed	Proposed vs. SIS	Proposed vs. Precomp. logic	Proposed vs. Shan. expansion
b1	16	19	18	12	–25.0%	–36.8%	–33.3%
cc	99	123	82	99	0.0%	–19.5%	20.7%
cht	258	262	412	258	0.0%	–1.5%	–37.4%
cm42a	34	46	48	40	17.6%	–13.0%	–16.7%
cm82a	42	38	53	42	0.0%	10.5%	–20.8%
cm138a	34	51	56	28	–17.6%	–45.1%	–50.0%
cm162a	77	106	69	55	–28.6%	–48.1%	–20.3%
cmb	84	56	156	111	32.1%	98.2%	–28.8%
cu	88	89	74	65	–26.1%	–27.0%	–12.2%
duke2	489	704	789	646	32.1%	–8.2%	–18.1%
f51m	130	207	220	135	3.8%	–34.8%	–38.6%
majority	20	24	39	17	–15.0%	–29.2%	–56.4%
misex2	153	225	219	115	–24.8%	–48.9%	–47.5%
pcle	107	112	158	107	0.0%	–4.5%	–32.3%
pcler8	147	189	210	147	0.0%	–22.2%	–30.0%
sao2	129	242	337	197	52.7%	–18.6%	–41.5%
sct	115	122	195	249	116.5%	104.1%	27.7%
x2	67	71	117	62	–7.5%	–12.7%	–47.0%
z4ml	55	59	107	47	–14.5%	–20.3%	–56.1%
9symml	204	211	184	150	–26.5%	–28.9%	–18.5%
Average	117	148	177	126	10.0%	–12.7%	–27.1%

using a simulated annealing algorithm. Higher level kernels contain multiple lower level kernels. For this reason, we only manage one kernel as a solution. The move function selects the newly generated kernel randomly according to the annealing schedule. If the cost of a neighboring solution decreases, the generated partitioning solution is accepted. If the cost increases, the move is accepted with a probability varying according to the temperature. The simulated annealing algorithm terminates when the quality of a solution does not improve for a constant number of moves. The cost function has been derived to select the best kernel that can lead to reduction of power under area constraints. The cost function is given as C = $P_K + P_C + \mathrm{p}(K) \times P_A + (1 - \mathrm{p}(K)) \times P_B + P_{mask}$, where p($K$) is the signal probability of kernel K output, while P_K, P_A, P_B, P_C, and P_{mask} are the power consumed by the sub-circuits realizing K, f_A, f_B, f_C, and (M_0+M_1), respectively. The kernel search is performed using BDD, and its cost function is calculated by the estimated circuit power using the switching activities of roughly generated subcircuits or their BDD sizes. In the circuit synthesis process, all the subcircuits are generated from the selected kernel using (1)-(4). The runtime for the best kernel search and cost evaluation is small because the BDD operation is very efficient and the number of kernels is not large in most cases.

IV. EXPERIMENTAL RESULTS

The proposed algorithm was implemented and integrated within the SIS logic synthesizer [6]. The SIS optimizes the original MCNC benchmark circuits. Table 1 presents a comparison of power dissipations between the precomputation scheme and the proposed scheme for MCNC benchmark circuits. The measurements are made at a clock frequency of 20 MHz using the zero-delay model. As the table shows, power dissipations are significantly reduced by applying the

proposed scheme to the benchmark circuits. The reduction of power dissipation in the circuits generated by the proposed algorithm is 27.9% less than the original circuits optimized for area and 22.5% less than the circuits based on the precomputation scheme. Table 2 shows that the area measured in literal counts has also been reduced.

V. CONCLUSION

In this letter, we presented a new algorithm for low-power, low-area overhead logic circuit design based on partitioning. The proposed algorithm partitions a given circuit to generate circuits that dissipate minimal power. We reduce the area by sharing the duplicated logic parts in the partitioned subcircuits. The partitioned circuits are further optimized to reduce the area by utilizing don't-care sets. Experimental results show that the proposed algorithm is efficient for designing low-power, low-area combinational circuits.

REFERENCES

[1] M. Alidina, J. Monteiro, S. Devadas, and A. Ghosh, "Precomputation-Based Logic Optimization for Low Power," *Proc. ICCAD*, Nov. 1994, pp. 74-81.

[2] V. Tiwari, S. Malik, and P. Ashar, "Guarded Evaluation: Pushing Power Management to Logic Synthesis/Design," *IEEE Trans. CAD of Integrated Circuits and Systems*, vol. 17, no. 10, Oct. 1998, pp. 1051-1060.

[3] I.S. Choi and S.Y. Hwang, "A Circuit Partitioning Algorithm for Low Power Design under Area Constraints Using Simulated Annealing," *IEE Proceedings - Circuits, Devices, and Systems*, vol. 146, no. 1, Feb. 1999, pp. 8-15.

[4] R. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Trans. on Computers*, vol. C-35, no. 8, Aug. 1986, pp. 677-691.

[5] M. Damiani and De Micheli, "Don't Care Set Specifications in Combinational and Synchronous Logic Circuits," *IEEE Trans. CAD of Integrated Circuits and Systems*, vol. 12, no. 3, Mar. 1993, pp. 365-388.

[6] A. Aziz, F. Balarin, R. Brayton, and A. Sangiovanni-Vincentelli, "Sequential Synthesis Using SIS," *Proc. ICCAD*, Nov. 1995, pp. 612-617.

[7] C.-S. Yim, J.-S. Roh, E.-S. Choi, J.-H. Baek, and S.-J. Cho, "Resynchronization of Modified JPEG Using a Power Allocation Scheme in a Direct Sequence CDMA System," *ETRI J.*, vol. 24, no. 5, Oct. 2002, pp. 405-408.

[8] H.S. Lee, U.G. Joo, H.H. Lee, and W.W. Kim, "Optimal Time Slot Assignment Algorithm for Combined Unicast and Multicast Packets," *ETRI J.*, vol. 24, no. 2, Apr. 2002, pp. 172-175.