International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:9, 2007

Power-Efficient AND-EXOR-INV Based Realization of Achilles' heel Logic Functions

Padmanabhan Balasubramanian, and R. Chinnadurai

***Abstract*—**This paper deals with a power-conscious AND-EXOR-Inverter type logic implementation for a complex class of Boolean functions, namely Achilles' heel functions. Different variants of the above function class have been considered viz. positive, negative and pure horn for analysis and simulation purposes. The proposed realization is compared with the decomposed implementation corresponding to an existing standard AND-EXOR logic minimizer; both result in Boolean networks with good testability attribute. It could be noted that an AND-OR-EXOR type logic network does not exist for the positive phase of this unique class of logic function. Experimental results report significant savings in all the power consumption components for designs based on standard cells pertaining to a 130nm UMC CMOS process The simulations have been extended to validate the savings across all three library corners (typical, best and worst case specifications).

***Keywords*—**Achilles' heel functions, AND-EXOR-Inverter logic, CMOS technology, Low power design.

I. Introduction

Low power VLSI design has emerged as a major technology driver in the recent past. Building low-power VLSI circuits and systems has emerged as highly in demand because of the fast growing technologies and markets in mobile computing and wireless communication systems. Due to higher integration, power density has increased. The battery technology does not advance at the same rate as the microelectronics technology. There is a limited amount of power available for the mobile systems. So designers are faced with more constraints: low-power consumption, high speed, and small silicon area. These considerations have resulted in the growing need for minimizing power in today's digital circuits. Design of circuits aiming for low power is not a straight-forward task, as it involves all the IC design stages beginning with the system behavioral description and ending with the fabrication and packaging processes. Therefore, there has been an increasing thrust towards considering power dissipation during all the stages of the design cycle. This paper considers the issue of lowering power dissipation for a unique class of combinational logic functions viz. Achilles' heel functions, realized in terms of AND-EXOR-Inverter logic style. The power optimization methodology is primarily targeted at the logic level in this article, with the final implementation targeting standard cell-based CMOS designs; though power management could be addressed at architectural, algorithmic and circuit levels [1]. However, it should be noted that power optimization, as mentioned in this paper, is achieved by an area-centric synthesis approach; nonetheless no trade-off is involved between the above two design metrics. Although the delay component has not been specifically considered in this paper, it is expected that the proposed synthesis scheme is most likely to result in a multi-level technology mapped solution with improved speed performance, for higher-order functionalities. The reason for this can be attributed to the exponential increase in the number of irredundant prime implicants, for a gradual increase in the number of primary circuit inputs, in case of other methods.

The remainder of this paper is organized as follows. Section 2 elucidates the different power dissipation components in CMOS based digital circuit implementations. Section 3 throws light on the intrinsic features of Achilles' heel logic functions. Section 4 reviews the merits and demerits of AND-EXOR logic synthesis schemes. It then explains the existing efficient synthesis procedure for two-level AND-EXOR logic realization of any arbitrary combinational logic function and a possible decomposition and technology mapping to enable a power optimal multi-level logic realization. This is followed by the proposed translated AND-EXOR-Inverter based synthesis method and its significance for the unique class of functions, considered in this paper. Section 5 portrays the power estimation methodology and the results obtained for the three different library specifications. Section 6 provides a short analysis of the results obtained and the inferences obtained from this research work. Also a suggestion for an optimization strategy that could be considered at the gate level so as to further squeeze the power envelope is highlighted.

II. Power considerations

There are three major components of power dissipation in CMOS based circuits. In simple terms, they are briefly described as: Switching power – power consumed by the circuit node capacitances during transistor switching, Short circuit power – power consumed because of the current flowing from power supply to ground during transistor

Padmanabhan Balasubramanian is with the School of Computer Science, The University of Manchester, Manchester, Lancashire M13 9NT, UK (phone: +44-161-2756294; e-mail: padmanab@cs.man.ac.uk).

Dr. R. Chinnadurai was with the National Institute of Technology (formerly Regional Engineering College), Tiruchirappalli 620 015, TN, India. He is now with the Department of Electronics and Communication Engineering, Velammal Engineering College (affiliated to Anna University), Chennai 600 066, TN, India (phone: +91-44-26591860; e-mail: rcdurai@yahoo.com).

switching and Static power – due to leakage and static currents. The sources of power dissipation are summarized by the following expression.

$$P = 0.5\, C\, V_{dd}^{2}\, f\, N + Q_{sc}\, V_{dd}\, f\, N + I_{leak}\, V_{dd} \qquad (1)$$

where P denotes the total power, V_{dd} is the supply voltage and f is the frequency of operation.

The first term represents the power required to charge and discharge circuit nodes. Node capacitances are represented by C. The factor N is the switching activity, i.e., the number of gate output transitions per clock cycle.

The second term in (1) represents power dissipation during output transitions due to current flowing from the supply to ground. This current is often called short-circuit current. The factor Q_{sc} represents the quantity of charge carried by the short-circuit current per transition.

The third term in (1) represents static power dissipation due to leakage current, I_{leak}. Device source and drain diffusions form parasitic diodes with bulk regions. Reverse bias currents in these diodes dissipate power. Subthreshold transistor currents also dissipate power. In the sequel, we will refer to the above three terms as switching activity power; short-circuit power and leakage current power. In VLSI circuits that use well-designed logic-gates, switching activity power accounts for over 90% of the total power dissipation and is due to the charging and discharging of load capacitances of logic gates, gate input and intrinsic capacitances and interconnects [2]. However, for technology nodes corresponding to 90nm and less, leakage power appears to become considerable, even while the chip is in active mode.

III. Achilles' heel logic functions

An Achilles' heel logic function is basically a read-once function, in that the parse tree has no variable repeated. Read-once functions have interesting special properties [3] [4] [5] and according to [6] account for a large percentage of functions which arise in real circuit applications. They have gained interest in the field of computational learning theory [7]. Such a function is generally unate (where each variable appears in its true form or complementary form, but not both) and is normally of the form $a_1a_2 + a_3a_4 + \ldots. + a_{2m-1}a_{2m}$. This requires ($2^m$-$1$) products for representation in conventional AND-EXOR synthesis format. Different variants of this function are possible: positive – where the description sets [8] corresponding to the essential prime implicants (EPI) of the function contain only input variables in normal form and not in complementary form, negative – where the description sets of the EPI of the function comprise input variables appearing in only complementary form and pure horn – where each of the description sets of the EPI of the function strictly contain only a single variable in inverted form while the others appear in non-inverted form [25]. The input file specification of an arbitrary Achilles' heel function could be diverse. In general, it could be given by the form (p, pq); where p and q are real, positive integers with p, $q \geq 2$. In this paper, we have considered values for p from 2 till 6, with q assuming values of 2 and 3 respectively for each p, for simulation purpose.

It could be observed that logic factorization operations (Algebraic/Boolean) cannot be applied for this class of functions, as the set intersection operation between any of the essential cube description sets, comprising the function, would only yield a null set. This function category also exhibits another important feature; where the negative output phase is of very little significance and only the positive output phase is useful for physical implementation. This is evident from the numerical quantities listed in Table I for an input file specification, with p equal to 2 and q varying from 2 to 16 in steps of one. Table I also drives home the point that for a gradual increase in the number of inputs, an exponential increase in the number of irredundant prime cubes happens for the complementary output phase.

The simulation time required for the synthesis of an Achilles' heel function in conventional two-level logic, even with a standard logic minimizer such as Espresso [9], increases substantially with increase in the number of inputs due to the above phenomenon. Hence alternative strategies to just speed up the minimization process especially targeting this class of functions have been devised [10] [11].

It is worth mentioning a generalized formulation here, based on the principle of mathematical induction, for quantitatively estimating the number of EPI of the Achilles' heel function for both the normal (positive) and complementary (negative) output phases. If n represents the cardinality of the support set of an Achilles' heel function; irrespective of the input file specification, the number of EPI for the normal phase of the function would be $O[n/p]$ and that for the complementary phase, $O[p^{(n/p)}]$.

Table I
Comparison of number of essential prime cubes for positive and negative phases of output function

# Inputs	# EPI (Normal phase)	# EPI (Complementary phase)
4	2	4
6	3	8
8	4	16
10	5	32
12	6	64
14	7	128
16	8	256
18	9	512
20	10	1024
22	11	2048
24	12	4096
26	13	8192
28	14	16384
30	15	32768
32	16	65536
Total	135	131068

IV. AND-EXOR-INVERTER REALIZATIONS

Most logic synthesis tools use AND and OR gates as basic logic elements, and they derive multi-level logic circuits from AND-OR two-level circuits. Thus, the minimization of sum-of-products expressions, which corresponds to the minimization of AND-OR two-level circuits, is of vital importance in such design automation tools. However, two-level logic is of minimal significance in a VLSI design environment.

Arithmetic and error correcting circuits can be realized with fewer gates if EXOR gates are available as well as AND and OR gates. Such circuits can be derived from AND-EXOR two-level circuits (AND-EXORs). So, the minimization of Exclusive-OR sum-of-products expressions, which corresponds to the minimization of AND-EXORs, is also important. ESOPs require fewer products than SOPs to realize randomly generated functions and symmetric functions [12] [13]. To realize an arbitrary logic function of six variables, an ESOP requires at most 16 products, whereas a SOP requires 32 products [14]. So ESOPs are important for efficient logic design. The number of products in AND-ORs can be reduced by adding decoders to the inputs. In a similar way, the number of products in AND-EXORs can be reduced by adding decoders to the inputs. A comparison in terms of the number of products and connections required for AND-OR and AND-EXOR logic with 1-bit and 2-bit decoders for arithmetic functions is given in [15], which implies that the circuits based on ESOPs are simpler than the ones based on SOPs. Though EXOR gate based designs were found to be suitable for look-up table based FPGAs (such as Xilinx LCA 3000), Maitra cascade type FPGAs (Concurrent Logic CLi6000), PLDs (examples include Signetics LHS501, AMD22XP10 and MAX EPLDs); the fourth application is gate array and standard cell VLSI. This is because EXOR gates are available in most VLSI cell libraries, and the selective and sensible use of EXOR gates can reduce the total costs even if EXORs are more expensive than OR gates in terms of area, power and speed.

Although ESOP realizations require more products than SOPs for Achilles' heel functions, we consider their implementation using standard cell libraries (based on a full-custom design) owing to their good design-for-testability attributes [16]. For e.g. in two-valued logic, a two-level sum-of-products realization of the *n*-variable even parity generator function requires all 2^n possible input vectors as tests to detect all single stuck-at faults. However, this function can also be implemented as a multi-level tree of two-input EXOR gates, and this realization requires only two tests to detect all single stuck-at faults. In two-valued systems, testing the multi-level tree of EXOR gates is easy because in a fan-out free linear circuit, any single fault propagates to the output independent of the input vector applied. Hence the focus of this paper is to obtain synthesis solutions using AND and EXOR gates and employ inverting buffers, where necessary (as input inversions are not assumed for cell-based designs), so that the final physical implementation employing standard cells would not only incorporate good testability properties [26], but also good power dissipation characteristics.

Although no efficient minimization algorithm for Exclusive-OR Sum-of-Products (ESOP) is known for more than five variables, heuristics have been formulated which obtain near minimum or exact minimum ESOP forms [17] [18] [19] [20] [21] [22]. Amongst these, [21] is known to be especially an efficient heuristic for EXOR based logic simplification. However, the above heuristic do not take into account don't care terms while synthesizing EXOR based solutions.

As can be seen from Table II, the existing AND-EXOR synthesis technique would require product terms of $O[2^k\text{-}1]$ for representation of an Achilles' heel function, with *k* being the number of essential prime implicants of the function (positive phase), expressed as a disjunction. So, the increase in the number of cubes required for the function to represent it exactly in two-level logic soars with increase in the number of inputs. Also two-level solutions cannot be directly synthesized using standard cells of a traditional CMOS library due to fan-in restrictions. Hence EXOR gates with fan-in greater than three and AND gates with fan-in greater than four are decomposed into multiple fan-out free tree structures, in line with the technology binding mechanism highlighted in [23]. This is found to effect a low-power multi-level realization.

Due to the significant increase in the number of cubes required for the function based on synthesis using the standard method and consequently, it results in more power consumption; a method has been proposed to reduce the number of cubes and gates (cells) required for implementation with AND and EXOR gates, by introducing inverters which perform cube inversions apart from negating inputs as required. Nevertheless, this is achieved by a translation of the original function description in disjunctive form, by applying the following two axioms. Also, the logic network representation for the conventional synthesis solution would constitute sort of a leaf-DAG (directed acyclic graph) structure for this class of functions, with DAG-ness exhibited only by the leaves (edges of the network) associated with the primary circuit inputs. From the above discussion, it becomes clear that there is a slight compromise introduced into the actual physical description in order to enable design implementations which exhibit reasonably good testability properties and simultaneously low power consumption. Due to this relaxation, we find that the number of cubes needed for an Achilles' heel function representation is now $O[k]$, similar to that of a disjunctive normal form, but with additional product term inversions of $O[k\text{-}1]$.

It has been inferred that the above generalizations for the number of cubes in the function representation, as per the traditional method and the proposed method holds good, irrespective of the input file specification.

A. Axiom 1

Two mutually exclusive or mutually disjoint product terms (cubes) which are OR-ed could be EXOR-ed without affecting the logic functionality. This basically amounts to proving the validity of the Shannon's theorem for Galois field, *GF* (*2*).

B. Axiom 2

Based on lemma 1, the absorption law of Boolean algebra could be rewritten as $a + a'b = a + b = a \oplus a'b$.

V. Power estimation methodology and Simulation results

The EXOR-based synthesis solutions for the Achilles' heel functions (in different flavors) were technology mapped by following the technology binding procedure, illustrated in [23], which promises reduced power consumption for a full-custom design approach, with a user-defined base-function set. The power dissipation results were then obtained for the case of maximum input activity using Synopsys tools [24] by targeting a 130nm UMC CMOS technology process, across three different library specifications viz. typical case (V_{dd} of 1.2V, Ambient temperature of 25°C), worst case (V_{dd} of 1.08V, Ambient temperature of 125°C) and best case (V_{dd} of 1.32V, Ambient temperature of 0°C), in order to extensively validate the power savings garnered by the proposed implementation over that of a decomposed conventional synthesis solution. The simulation results were obtained for an input frequency of 100 MHz, corresponding to the above three process corners and are indicated by Tables III, IV and V respectively, mentioned in the appendices. The positive, negative and pure-horn Achilles' heel functions are denoted by the terminologies PAH, NAH and PHAH respectively. The indices before and after these terminologies indicate the values of p and pq; where for p varying from 2 to 6, q assumes values of 2 and 3 for each of the values of p.

VI. Discussion of results and Conclusion

A power optimized AND-EXOR-Inverter based realization for a unique class of logic functions, namely Achilles' heel logic functions has been dealt with in this work. The objective has been two-fold: implementation incorporating good design-for-testability (DFT) attributes and simultaneously low power consumption, utilizing standard cells. The proposed synthesis strategy has been compared with a decomposed standard ESOP simplification strategy. Maximum fan-in based decomposition was performed in order to facilitate power-aware technology mapping of the two-level reduced ESOP solutions in multi-level format.

The power estimation has been done considering maximum input activity, with a clock of 100MHz, targeting a 130nm standard CMOS process across three different library corners. The proposed methodology highlights the fact that a slight compromise introduced into the conventional AND-EXOR-Inverter based logic realizations, by introducing input cube inversions, facilitates a power optimal solution, for realization with standard cells, whilst incorporating good DFT properties. The experimental results report significant overall average savings for the proposed method by 50.28%, 50.42% and 43.77%, with respect to total power, dynamic power and leakage power, over a decomposed multi-level implementation of a traditional two-level AND-EXOR synthesis solution (with inverters as required to obtain negations of primary inputs). A possible logic optimization step, which could be considered with the intent of further exploiting the power envelope, would be to consider the issue of input reordering/transistor reordering [27], as a measure aimed at further minimizing the switching power component, provided apriori information about the input signal probabilities are made available.

References

[1] A.P. Chandrakasan, and R.W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proceedings of the IEEE*, vol. 83(4), pp. 498-523, April 1995.

[2] A.P. Chandrakasan, S. Sheng, and R.W. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol. 27(4), pp. 473-484, April 1992.

[3] V. Gurvich, "Criteria for repetition-freeness of functions in the algebra of Logic," *Soviet Math*., vol. 43(3), pp. 721-726, 1991.

[4] M. Karchmer, N. Linial, I. Newman, M. Saks and A. Wigderson, "Combinatorial characterization of read-once formulae," *Discrete Mathematics*, vol. 114(1-3), pp. 275-282, April 1993.

[5] I. Newman, "On read-once Boolean functions," in *Boolean function complexity: Selected papers from LMS Symposium*, pp. 24-34, 1990.

[6] J. Peer, and R. Pinter, "Minimal decomposition of Boolean functions using non-repeating literal trees," *Proc. IFIP International Workshop on Logic and Architecture Synthesis*, pp. 129-139, December 1995.

[7] N.H. Bshouty, T.R. Hancock, and L. Hellerstein, "Learning Boolean read-once formulas with arbitrary symmetric and constant fan-in gates," *Proc. 5th Annual ACM Conference on Computational Learning Theory*, pp. 1-15, 1992.

[8] P. Balasubramanian, and R.T. Naayagi, "Critical path delay and net delay reduced tree structure for combinational logic circuits," *International Journal of Electronics, Circuits and Systems*, vol. 1(1), pp. 19-29, 2007.

[9] P.C. McGeer, J.V. Sanghavi, R.K. Brayton, and A.L. Sangiovanni-Vincentelli, "Espresso-Signature: a new exact minimizer for logic functions," *IEEE Trans. on VLSI Systems*, vol. 1(4), pp. 432-440, December 1993.

[10] P. Srinivasa Rao, and J. Jacob, "A fast two-level logic minimizer," Proc. *11th International Conf. on VLSI Design*, pp. 528-533, January 1998.

[11] N.N. Biswas, C. Srikanth, and J. Jacob, "Cubical CAMP for minimization of Boolean functions," *Proc. 9th International Conf. on VLSI Design*, pp. 264-269, January 1996.

[12] T. Sasao, and P. Besslich, "On the complexity of mod-2 sum PLA's," *IEEE Trans. on Computers*, vol. 32(2), pp. 262-266, February 1990.

[13] T. Sasao, "EXMIN: A simplification algorithm for Exclusive-OR Sum-of-Products expressions for multiple-valued input two-valued output functions," *Proc. 20th International Symposium on Multiple-Valued Logic*, pp. 128-135, May 1990.

[14] N. Koda, and T. Sasao, "An upper bound on the number of products in AND-EXOR minimum expressions," *IEICE Trans. on Fundamentals*, vol. J75-D-I (3), pp. 135-142, March 1992.

[15] T. Sasao, *Logic Synthesis and Optimization*, Kluwer Academic Publishers, Massachusetts, USA, 1993.

[16] U. Kalay, M. Perkowski, and D. Hall, "A minimal universal test set for self test of EXOR Sum-of-Products circuits," *IEEE Trans. on Computers*, vol. 49(3), pp. 267-276, March 1999.

[17] D. Brand, and T. Sasao, "Minimization of AND-EXOR expressions using rewrite rules," *IEEE Trans. on Computers*, vol. 42(5), pp. 568-576, May 1993.

[18] H. Fleisher, M. Travel, and J. Yeager, "Computer algorithm for minimizing Reed-Muller canonical forms," *IEEE Trans. on Computers*, vol. C-36(2), pp. 247-250, February 1987.

[19] Y. Ye, and K. Roy, "An XOR-based decomposition diagram and its application in synthesis of AND-XOR networks," *IEICE Trans. on Fundamentals*, vol. E80-A (10), pp. 1742-1748, October 1997.

[20] Tsutomu Sasao, "EXMIN2: A simplification algorithm for Exclusive-OR-Sum-of-Products expressions for multiple-valued-input two-valued-output functions," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 12(5), pp. 621-632, May 1993.

[21] A. Mishchenko, and M. Perkowski, "Fast heuristic minimization of Exclusive-Sums-of-Products", *Proc. 5th International Reed-Muller Workshop*, pp. 242-250, 2001.

[22] S. Stergiou, and G. Papakonstantinou, "Exact minimization of ESOP expressions with less than eight product terms," *Journal of Circuits, Systems and Computers*, vol. 13(1), pp. 1-15, February 2004.

[23] P. Balasubramanian, and R. Arisaka, "A set theory based factoring technique and its use for low power logic design," *International Journal of Electrical, Computer, and Systems Engineering*, vol. 1(3), pp. 188-198, 2007.

[24] Synopsys Inc., Available: http://www.synopsys.com

[25] T. Eiter, T. Ibaraki, and K. Makino, "Bidual Horn functions and extensions, " *Discrete Applied Math.,* vol. 96-97, pp. 55-88, Oct. 1999.

[26] A. Sarabi, "Design of testability properties of AND/XOR networks," *Proc. IFIP Workshop on Applications of Reed-Muller Expansion in Circuit Design*, pp. 138-142, Germany, 1993.

[27] W. –Z. Shen, J.-Y. Lin, and F.-W. Wang, "Transistor reordering rules for power reduction in CMOS gates," *Proc. ASP-DAC*, pp. 1-6, 1995.

P. Balasubramanian completed his B.E in Electronics and Communication Engineering from University of Madras in 1998 and his M. Tech in VLSI System from National Institute of Technology, Tiruchirappalli in 2005. He was earlier Lecturer in the School of Electrical Sciences at Vellore Institute of Technology (University) TN, India. At present, he is pursuing research at The University of Manchester, UK. He has considerable publications in International Journals and Conferences. His research interests are in asynchronous design, combinational logic and data path optimization techniques for low power/ high-performance and CMOS based IC design.

Dr. R. Chinnadurai completed his B.Sc. (Physics) in 1972 from University of Madras; B. Tech in Electronics Engineering in 1975 and M. Tech in Applied Electronics in 1977, from Madras Institute of Technology. He worked for a brief period as Associate Lecturer in A.C. College of Engineering and Technology in 1977. He then worked as Lecturer in the Department of Electronics and Communication Engineering of National Institute of Technology (then Regional Engineering College), Tiruchirappalli, from 1977-1987. He subsequently completed his Ph.D. in Microwave Engineering in 1990 from Bharathidasan University, Tiruchirappalli. He served as an Assistant Professor in National Institute of Technology, Tiruchirappalli from 1987 till 2007. At present, he is a Professor in the Department of Electronics and Communication Engineering of Velammal Engineering College (affiliated to Anna University), Chennai, TN, India. He was a short-time visitor on a British Science and Engineering Research Council Research Experience Fellowship to the University of Nottingham, UK for collaborative research work on Spreading of Surface Acoustic Waves in 1987. He visited The University of Manchester (then University of Manchester Institute of Science and Technology), UK between January 1997 and June 1997 as part of the Indo-UK REC's Technical Project Co-operation programme and underwent training in Non-Conventional Energy Sources. He was the Principal Investigator/Co-Investigator of six successfully completed research projects in the area of Sonar Signal Processing, funded by Naval Science and Technical Laboratories, Vizhakapatnam over a period from 1985-1998. He has supervised PhD thesis in the area of Microwave Engineering. He was awarded the "Man of the Year – 1998" honor by the American Biographical Institute. He has published two papers in IET Proceedings and several papers in National and International Journals and Conferences. His interests include Microwave Engineering, Analog VLSI, Microelectronic circuits and CMOS based digital circuit design.

APPENDIX I

TABLE II
COMPARISON BETWEEN CONVENTIONAL AND PROPOSED AND-EXOR REALIZATION FOR DIFFERENT FUNCTION VARIANTS

Function type	# EPI (Normal output phase)	# Product terms for representation		# Product term inversions		# Input inversions	
		CAEI	PAEI	CAEI	PAEI	CAEI	PAEI
PAH	*k*	2^k-1	*k*	0	*k*-1	0	0
NAH	*k*	2^k-1	*k*	0	*k*-1	*n*	*n*
PHAH	*k*	2^k-1	*k*	0	*k*-1	*k*	*k*

CAEI – Conventional AND-EXOR-Inverter logic; PAEI – Proposed AND-EXOR-Inverter logic; *n, k* are positive integers; *k* > 1.

APPENDIX II

TABLE III
POWER CONSUMPTION RESULTS CORRESPONDING TO TYPICAL CASE LIBRARY SPECIFICATION

Function specification	Conventional AND-EXOR-Inverter logic			Proposed AND-EXOR-Inverter logic		
	Total power (μW)	Dynamic power (μW)	Leakage power (nW)	Total power (μW)	Dynamic power (μW)	Leakage power (nW)
2PAH4	2.48518	2.45805	27.1333	2.24024	2.2205	19.7438
2PAH6	6.12634	6.05236	73.9831	4.82456	4.78486	39.693
2PHAH4	3.37926	3.34901	30.2497	2.77155	2.74878	22.7752
2PHAH6	8.59133	8.51267	78.6577	5.58362	5.53913	44.497
2NAH4	4.29473	4.26136	33.3661	3.29314	3.26725	25.8916
2NAH6	11.1139	11.0306	83.3322	6.36895	6.31978	49.1716
3PAH6	2.06869	2.03984	28.8522	1.71376	1.69545	18.3125
3PAH9	5.45282	5.36944	83.3771	3.82848	3.79071	37.7681
3PHAH6	2.96052	2.92855	31.9685	2.22494	2.20351	21.4289
3PHAH9	7.89574	7.80769	88.0517	4.59575	4.5533	42.4427
3NAH6	4.76229	4.72409	38.2013	3.24635	3.21869	27.6617
3NAH9	12.835	12.7376	97.4008	6.13274	6.08095	51.7918
4PAH8	1.85928	1.8287	30.5776	1.33634	1.31464	21.7051
4PAH12	5.39395	5.29992	94.0255	2.20697	2.16588	41.0837
4PHAH8	2.72308	2.68939	33.694	1.8297	1.80488	24.8215
4PHAH12	7.76807	7.66937	98.7001	2.94691	2.90115	45.7582
4NAH8	5.39103	5.34799	43.0431	3.35196	3.31779	34.1706
4NAH12	15.0958	14.9831	112.724	5.23025	5.17047	59.7819
5PAH10	1.77155	1.73356	37.9905	1.23992	1.21014	29.776
5PAH15	5.31916	5.21113	108.024	2.31037	2.25716	53.2122
5PHAH10	2.64205	2.60094	41.1069	1.73335	1.70046	32.8924
5PHAH15	7.71539	7.60269	112.698	3.05053	2.99264	57.8867
5NAH10	6.207	6.15343	53.5724	2.32781	2.28246	45.3579
5NAH15	17.4947	17.3633	131.396	6.12246	6.04588	76.585
6PAH12	1.7851	1.7449	40.1966	0.957426	0.928231	29.1945
6PAH18	5.39326	5.27389	119.368	1.70293	1.65137	51.5639
6PHAH12	2.65202	2.6087	43.3129	1.45083	1.41852	32.3109
6PHAH18	7.7769	7.65286	124.042	2.44304	2.38681	56.2835
6NAH12	7.09938	7.04049	58.8948	4.00328	3.95539	47.8928
6NAH18	19.9633	19.8159	147.415	6.27169	6.19208	79.6114
Total	196.0168	193.8915	2125.355	97.33985	96.11886	1221.066
Mean	6.53389	6.46305	70.84517	3.24466	3.20396	40.7022
(% Decrease)	-	-	-	(50.34%)	(50.43%)	(42.55%)

APPENDIX III

TABLE IV
POWER CONSUMPTION RESULTS CORRESPONDING TO WORST CASE LIBRARY SPECIFICATION

Function specification	Conventional AND-EXOR-Inverter logic			Proposed AND-EXOR-Inverter logic		
	Total power (µW)	Dynamic power (µW)	Leakage power (nW)	Total power (µW)	Dynamic power (µW)	Leakage power (nW)
2PAH4	1.97715	1.91215	64.9924	1.77364	1.72688	46.7583
2PAH6	4.87962	4.70235	177.271	3.81229	3.71826	94.024
2PHAH4	2.67602	2.60382	72.2066	2.1985	2.14467	53.8372
2PHAH6	6.76014	6.57205	188.092	4.41455	4.30954	105.006
2NAH4	3.38816	3.30874	79.4209	2.61637	2.55532	61.0514
2NAH6	8.67294	8.47403	198.8913	5.04258	4.92675	115.827
3PAH6	1.66937	1.59899	70.3827	1.3653	1.32179	43.507
3PAH9	4.41464	4.21039	204.253	3.03215	2.94312	89.0397
3PHAH6	2.37096	2.29336	77.5969	1.77747	1.72675	50.7212
3PHAH9	6.29234	6.07726	215.074	3.65098	3.55112	99.861
3NAH6	3.7877	3.69568	92.0254	2.60154	2.53639	65.1497
3NAH9	10.0887	9.85203	236.717	4.89069	4.76919	121.504
4PAH8	1.51791	1.442	75.9148	1.0833	1.03122	52.079
4PAH12	4.42261	4.1881	234.509	1.79616	1.69663	99.5245
4PHAH8	2.2043	2.12117	83.129	1.4853	1.42601	59.2933
4PHAH12	6.26395	6.01861	245.33	2.39914	2.2888	110.346
4NAH8	4.30644	4.20167	104.772	2.71471	2.63378	80.936
4NAH12	11.91	11.6322	277.794	4.2433	4.10049	142.81
5PAH10	1.47223	1.37741	94.817	1.0305	0.958003	72.4963
5PAH15	4.41504	4.14464	270.406	1.90354	1.77626	127.279
5PHAH10	2.16293	2.06089	102.031	1.43257	1.35286	79.7105
5PHAH15	6.26787	5.98664	281.227	2.50664	2.36854	138.1
5NAH10	4.97205	4.84117	130.888	1.97373	1.86516	108.567
5NAH15	13.8014	13.4769	324.513	4.98505	4.80366	181.385
6PAH12	1.49913	1.39731	101.82	0.810926	0.739706	71.2202
6PAH18	4.52823	4.22539	302.845	1.43425	1.31011	124.143
6PHAH12	2.1852	2.07617	109.034	1.213	1.13457	78.4344
6PHAH18	6.36746	6.0538	313.667	2.03736	1.90239	134.964
6NAH12	5.69138	5.54627	145.105	3.27544	3.16094	114.506
6NAH18	15.766	15.3983	367.773	5.13099	4.94192	189.071
Total	156.7319	151.4895	5242.498	78.63197	75.72083	2911.152
Mean	5.22439	5.04965	174.74993	2.62107	2.52403	97.0384
(% Decrease)	-	-	-	(49.83%)	(50.02%)	(44.47%)

APPENDIX IV

TABLE V
POWER CONSUMPTION RESULTS CORRESPONDING TO BESTCASE LIBRARY SPECIFICATION

Function specification	Conventional AND-EXOR-Inverter logic			Proposed AND-EXOR-Inverter logic		
	Total power (μW)	Dynamic power (μW)	Leakage power (nW)	Total power (μW)	Dynamic power (μW)	Leakage power (nW)
2PAH4	3.27922	3.19828	80.9422	2.95204	2.89382	58.2243
2PAH6	8.12615	7.90527	220.87	6.34479	6.22773	117.063
2PHAH4	4.46258	4.37262	89.9659	3.64056	3.57353	67.0364
2PHAH6	11.4297	11.1953	234.406	7.34301	7.21209	130.922
2NAH4	5.66505	5.56606	98.9897	4.31373	4.23767	76.0602
2NAH6	14.7904	14.5425	247.942	8.35767	8.21321	144.458
3PAH6	2.74942	2.66181	87.6115	2.27158	2.21743	54.1473
3PAH9	7.26446	7.01003	254.43	5.0833	4.97214	111.153
3PHAH6	3.90565	3.80901	96.6353	2.92652	2.86335	63.1711
3PHAH9	10.4799	10.212	267.966	6.06725	5.94256	124.689
3NAH6	6.23989	6.12521	114.683	4.23297	4.15175	81.2186
3NAH9	16.9599	16.6649	295.037	8.03428	7.88252	151.76
4PAH8	2.48538	2.39128	94.1047	1.78634	1.72155	64.7932
4PAH12	7.21597	6.92514	290.834	2.95685	2.83306	123.791
4PHAH8	3.60128	3.49815	103.128	2.41746	2.34364	73.817
4PHAH12	10.3279	10.0235	304.37	3.90353	3.7662	137.327
4NAH8	7.03418	6.90398	130.2	4.35789	4.257	100.888
4NAH12	19.893	19.548	344.977	6.81403	6.63609	177.934
5PAH10	2.3733	2.25549	117.815	1.66396	1.57375	90.2105
5PAH15	7.1155	6.7795	335.996	3.09752	2.93831	159.204
5PHAH10	3.49761	3.37077	126.838	2.29488	2.19564	99.2343
5PHAH15	10.2553	9.90582	349.531	4.04395	3.87121	172.739
5NAH10	8.08578	7.92284	162.934	2.95865	2.82332	135.329
5NAH15	23.031	22.6273	403.674	7.96347	7.73658	226.882
6PAH12	2.3951	2.26877	126.332	1.29499	1.20614	88.85
6PAH18	7.23304	6.85722	375.818	2.30348	2.14832	155.167
6PHAH12	3.51346	3.3781	135.356	1.92588	1.82801	97.8738
6PHAH18	10.3513	9.96192	389.353	3.24981	3.08117	168.702
6NAH12	9.24045	9.05997	180.475	5.18358	5.04059	142.993
6NAH18	26.2599	25.8029	457.032	8.13649	7.90011	236.381
Total Mean (% Decrease)	259.2618 8.64206 -	252.7436 8.42479 -	6518.246 217.27487 -	127.9205 4.26402 (50.66%)	124.2885 4.14295 (50.82%)	3632.019 121.0673 (44.28%)