International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:5, 2008

Analysis of Effect of Pre-Logic Factoring on Cell Based Combinatorial Logic Synthesis

Padmanabhan Balasubramanian and Bashetty Raghavendra

***Abstract*—**In this paper, an analysis is presented, which demonstrates the effect pre-logic factoring could have on an automated combinational logic synthesis process succeeding it. The impact of pre-logic factoring for some arbitrary combinatorial circuits synthesized within a FPGA based logic design environment has been analyzed previously. This paper explores a similar effect, but with the non-regenerative logic synthesized using elements of a commercial standard cell library. On an overall basis, the results obtained pertaining to the analysis on a variety of MCNC/IWLS combinational logic benchmark circuits indicate that pre-logic factoring has the potential to facilitate simultaneous power, delay and area optimized synthesis solutions in many cases.

***Keywords*—**Algebraic factoring, Combinational logic synthesis, Standard cells, Low power, Delay optimization, Area reduction.

I. Introduction

RESEARCH into minimization (two-level and multi-level) and decomposition of logic functions have been pursued over the past several decades [1] [2] [3] [4] [5] [6] [7], as they can enable a reduction in the number of elements required to realize the logic corresponding to a requisite functionality. The motivation being that this could ultimately lead to a synthesis solution for a target functionality which minimizes all or some of the practical design metrics viz; power, delay and area. It is customary in commercial synthesis environments to effect a physical realization optimized for either speed or area. Optimization for power, though given wide recognition since the past few decades, usually depends upon utilizing two of the main options that prevail at the technology front: going for introduction of multiple V_{dd} in designs (as power can scale down quadratically with linear decrease of supply voltage) or replacement of some of the low V_t elements in the non-critical logic paths with high V_t cells with the intention of reducing power dissipation without sacrificing performance. Both these approaches have been found to be beneficial for logic designs pertaining to the deep submicron range, with the latter especially suitable for minimization of the static power component as well. This work looks at a rather simple option of pre-processing combinational logic (say, described initially at the behavioral level) and analyzing whether it would help in improvement of design parameters, when succeeded by automated logic synthesis in a practical standard cell based design environment. The results obtained for some MCNC/IWLS combinatorial benchmarks [8] [9] show that even a simple processing of combinatorial logic beforehand can effect good optimization during design synthesis.

The remaining portion of this paper is organized as follows. Section 2 provides concise preliminary information about Boolean function and network. Also, terminologies pertaining to [10] and a newly proposed terminology have been described in this section for the sake of clarity. Since this work builds upon references [10] and [11], the algorithm to yield a delay optimized solution proposed in [10] and the algorithm pertaining to the novel algebraic factoring technique proposed in [11] have not been mentioned here to avoid mere repetition and so the interested reader is directed to them for details. Nevertheless, we outline the general theme underlying the different logic formats and how they are arrived at on the basis of [11] in the next section, which also highlights the motivation for this work through some sample cases and illustrations. Section 4 gives the simulation results obtained for different benchmarks. We finally conclude in section 5.

II. Background

A. Boolean function

A single output Boolean function, $F(x_{n-1}, x_{n-2},\ldots,x_0)$ is a mapping, f: $\{0,1\}^n \rightarrow \{0,1,d\}$, where '*d*' denotes a *don't care* condition. If the don't care condition does not exist, then it is a completely specified Boolean function, otherwise it is an incompletely specified one. Each of the 2^n nodes in the Boolean space corresponds to a minterm. If a minterm is mapped to output 0 (1 or *d*), then it is called an *OFF-set* (*ON-set* or *DC-set*) minterm.

B. Boolean network

A binary logic network is a directed acyclic graph (DAG) with nodes representing Boolean functions. The *sources* of the graph are the primary inputs of the network; the *sinks* are the primary outputs. The inputs of a node are called its *fan-ins* and its outputs *fan-outs*.

C. Description of a Boolean term

The description set of a Boolean sum term (product term) [11] is denoted by the set of all literals of the sum term

Padmanabhan Balasubramanian is with the School of Computer Science, The University of Manchester, Oxford Road, Lancashire M13 9PL, United Kingdom (e-mail: spbalan04@gmail.com).

Bashetty Raghavendra is working as an Assistant System Engineer in Tata Consultancy Services, Kerala, India (e-mail: raghavendra@gmail.com).

(product term) in their actual form (whether complemented or uncomplemented), which a particular sum term (product term) is dependent upon for its evaluation to a logic 1 (logic 0) state.

D. Cubes Description Intersection set

The intersection of the description set of two Boolean cubes, say C_1 and C_2, can be defined by a cubes description intersection set, CDI. For e.g. if we have $D(C_1) = \{a,b',c,d\}$ and $D(C_2) = \{a',b',c,e\}$, then CDI $[D(C_1), D(C_2)] = \{b',c\}$. This definition is valid for Boolean sum terms as well.

III. Motivation and Method

The motivation for this work stems from the inspiration articulated for an earlier work [10], and bears some similarities. The primary difference being that the target technology is now ASIC based rather than being FPGA based. The specified combinatorial logic is first reduced into both its minimum sum-of-products (MSOP) and minimum product-of-sums formats (MPOS obtained from negative phase logic reduction) using a standard logic minimizer, Espresso [12]. Multiple output minimizations were resorted to, so that maximum amount of logic sharing would be ensured between the different outputs of the function. The MSOP and MPOS forms are then decomposed using the algebraic factoring scheme of [11]. Also the benchmark functionality was reduced on a whole with the output phase optimization (OPO) provision available in Espresso. This facilitated obtaining MSOP for certain outputs and MPOS for the remaining outputs. They were then subsequently factorized likewise.

The timing driven logic bi-decomposition procedure proposed in [13] mainly considered factoring the MSOP expression using a combination of associative, commutative and distributive Boolean laws. We consider factoring both the MSOP and MPOS expressions corresponding to each and every function output based on the technique of [11] and also a simultaneous factoring of the different output expressions, based on their output phase. These expressions described at the behavioural level, are then used as the input for automated synthesis using a commercial synthesis tool (say, Cadence Encounter RTL compiler, which has been used for this work). To differentiate between the design metrics governing the various synthesis solutions, the original MSOP of the function (MSOP for all the function outputs based on multiple outputs optimization) was also given as input to the synthesis tool, since it is difficult to directly specify the functionality for all the benchmarks. All the synthesis results reported herein pertain to a 130nm TSMC bulk CMOS process for a typical corner with a supply voltage of 1.2V at an ambient temperature of 25°C, with default switching activity rates governing the primary inputs. All the function simulations have not been constrained by a common reference clock; rather the clock frequency depends upon the critical path delay (CPD) of each individual function. The design parameters were extracted after technology mapping, with segmented wire load information included.

A. Case 1

Let us first consider the case study of a simple benchmark, *xor5*, which has 5 inputs and a single output, whose output is a logical exclusive-OR of all the inputs.

The different expression formats corresponding to *xor5* are *xor5*_MSOP (MSOP form for *xor5*), *xor5*_f_MSOP (factored MSOP form of *xor5*), *xor5*_f_MPOS (factored MPOS form of *xor5*) and *xor5*_f_OPO (factored OPO form of xor5) respectively. In fact, *xor5*_f_MSOP and *xor5*_f_OPO expressions are the same. This sort of Boolean matching is visible in many other benchmarks which have a single output.

Table I
Delay and Area metrics for *xor5*

Function format	Critical path delay (ps) and Clock period (ps)	Cell area (μm^2)
*xor5*_MSOP	338 (345)	526
*xor5*_f_MSOP	338 (345)	114
*xor5*_f_MPOS	337 (345)	115

Table II
Power parameters for xor5

Function format	Switching power (nW)	Internal power (nW)	Net power (nW)
*xor5*_MSOP	20941.18	5592.58	15348.60
*xor5*_f_MSOP	6498.43	3413.23	3085.20
*xor5*_f_MPOS	5706.33	2955.57	2750.76

From Tables I and II, we find that the initial *xor5*_f_MPOS form not only facilitates a delay optimized realization but also a power efficient one, in comparison with the other two logical formats. Since the synthesis has been performed with focus on speed, the critical path delay of the synthesized logic pertaining to different logic formats are comparable. The differences are observable mainly with respect to power and area. The internal power refers to the power consumed within the gates (i.e. by the standard cells), while the net power refers to the power dissipated in interconnects. The switching power parameter is basically a summation of the internal and net power components. In comparison with the realization based on *xor5*_MSOP form, *xor5*_f_MSOP and xor5_f_MPOS forms pave way for realizations which are only approximately 1/3rd and 1/4th power consuming. In terms of area, the synthesis solutions resulting from *xor5*_f_MSOP and *xor5*_f_MPOS forms are comparable. However, this is not necessarily the case with each and every benchmark, as obviously, their functionalities differ significantly. This can be understood from the results reported in the next section for various benchmarks. To estimate the power, delay and area metrics of purely the combinatorial logic underlying *xor5*, the reference clock is disconnected from the combinational part; nevertheless it is incorporated to constrain the designs for simulation purposes and its period has been set as 345ps in this case, since the maximum path delay is only 338ps and hence there is a positive timing slack of at least 7ps.

B. Case 2

We now consider a 2-bit magnitude comparator as a sample. There are a total of 4 inputs and 3 outputs, with the 3 outputs indicating lesser than, greater than and equality conditions. Similar to the previous case, f_MSOP form and f_OPO forms of this function are similar. Delay and area metrics corresponding to synthesis of different logical formats are given in Table III, with the power components following it.

TABLE III
DELAY AND AREA METRICS FOR 2-BIT MAGNITUDE COMPARATOR

Function format	Critical path delay (ps) and Clock period (ps)	Cell area (μm^2)
MSOP	195 (200)	160
f_MSOP	188 (200)	81
f_MPOS	196 (200)	239

TABLE IV
POWER PARAMETERS FOR 2-BIT MAGNITUDE COMPARATOR

Function format	Switching power (nW)	Internal power (nW)	Net power (nW)
MSOP	7279.46	2134.70	5144.76
f_MSOP	3942.96	1515.12	2427.84
f_MPOS	11345.22	3936.42	7408.80

In this case, the f_MSOP form is found to yield delay and power optimized solutions in comparison with those of the other formats. The f_MPOS form leads to the least efficient realization in terms of power, delay and area. This is attributable to the regularity exhibited by the outputs in the positive phase with the result that the number of essential prime implicants is much lesser for the positive phase compared to the negative phase. This phenomenon is also exhibited by some of the benchmark functions, listed in the next section. In comparison with the realization based on MSOP form, f_MSOP form results in a synthesis which betters the former in terms of power, delay and area by 45.8%, 3.6% and 49.4% respectively.

IV. BENCHMARK RESULTS AND DISCUSSION

A variety of combinational benchmark functionalities were considered for the purpose of validation, with the biggest one comprising 94 inputs and 43 outputs. Four different logical formats were considered for each and every function. Switching power or dynamic power was found to be the dominant source of power consumption in the designs based on the 130nm TSMC CMOS process, under typical operating conditions. Hence, leakage power component has not been explicitly listed here. The switching power, longest path delay (and clock period pertaining to each and every design) and cell area for the benchmarks considered have been mentioned in Table V, while the internal and net power components have been separately mentioned in Tables VI and VII additionally.

TABLE V
DELAY AND AREA METRICS FOR LOGICAL FORMATS CORRESPONDING TO DIFFERENT MCNC/IWLS COMBINATIONAL BENCHMARKS

Benchmark and its specification	Logic format	Switching power (nW)	CPD (ps) and Clock period (ps)	Cell area (μm^2)
newtag (8 I/p, 1 O/p)	MSOP	5583.668	214 (225)	146
	f_MSOP	3837.579	217 (225)	95
	f_MPOS	2420.998	219 (225)	56
	f_OPO	2420.998	219 (225)	56
misj (35 I/p, 14 O/p)	MSOP	13191.873	223 (225)	331
	f_MSOP	7632.579	218 (225)	204
	f_MPOS	7999.962	221 (225)	210
	f_OPO	8138.095	221 (225)	207
clpl (11 I/p, 5 O/p)	MSOP	7256.674	369 (380)	180
	f_MSOP	9099.178	375 (380)	166
	f_MPOS	9099.178	375 (380)	166
	f_OPO	9099.178	375 (380)	166
c17 (5 I/p, 2 O/p)	MSOP	2700.514	145 (150)	59
	f_MSOP	1630.233	134 (150)	37
	f_MPOS	1684.418	145 (150)	37
	f_OPO	1503.476	146 (150)	34
con1 (7 I/p, 2 O/p)	MSOP	4459.920	194 (200)	102
	f_MSOP	5520.707	194 (200)	121
	f_MPOS	8197.420	195 (200)	183
	f_OPO	4866.812	195 (200)	107
newtpla1 (10 I/p, 2 O/p)	MSOP	7928.536	225 (230)	182
	f_MSOP	3728.885	223 (230)	81
	f_MPOS	3728.885	223 (230)	81
	f_OPO	3728.885	223 (230)	81
arpanet (9 I/p, 1 O/p)	MSOP	14868.155	293 (300)	356
	f_MSOP	8457.634	294 (300)	194
	f_MPOS	7611.737	292 (300)	148
	f_OPO	7611.737	292 (300)	148
newtpla2 (10 I/p, 4 O/p)	MSOP	11624.800	313 (320)	329
	f_MSOP	6097.805	308 (320)	166
	f_MPOS	8055.435	313 (320)	224
	f_OPO	9304.380	312 (320)	233
newapla1 (12 I/p, 7 O/p)	MSOP	9462.503	313 (320)	250
	f_MSOP	6375.537	304 (320)	168
	f_MPOS	6560.297	260 (320)	156
	f_OPO	6560.297	260 (320)	156
newill (8 I/p, 1 O/p)	MSOP	10587.621	313 (320)	292
	f_MSOP	9680.611	314 (320)	226
	f_MPOS	9445.453	312 (320)	221
	f_OPO	9680.611	314 (320)	226
exam3_d (4 I/p, 1 O/p)	MSOP	2186.772	212 (230)	51
	f_MSOP	1858.872	221 (230)	36
	f_MPOS	1969.611	203 (230)	39
	f_OPO	1858.872	221 (230)	36
wim (4 I/p, 7 O/p)	MSOP	9245.958	180 (190)	188
	f_MSOP	10022.522	185 (190)	205
	f_MPOS	7174.969	189 (190)	158
	f_OPO	6821.124	185 (190)	139
t4 (12 I/p, 8 O/p)	MSOP	20892.227	225 (230)	509
	f_MSOP	22653.765	225 (230)	485
	f_MPOS	14267.827	225 (230)	300
	f_OPO	11197.176	224 (230)	253
newcwp (4 I/p, 5 O/p)	MSOP	11476.524	223 (230)	226
	f_MSOP	6783.209	225 (230)	129
	f_MPOS	8970.173	223 (230)	192
	f_OPO	6085.348	218 (230)	117
dc1 (4 I/p, 7 O/p)	MSOP	14051.409	216 (220)	348
	f_MSOP	14194.135	215 (220)	319
	f_MPOS	16820.461	216 (220)	362
	f_OPO	14345.059	215 (220)	311
alcom (15 I/p, 38 O/p)	MSOP	26928.134	235 (240)	659
	f_MSOP	22446.623	233 (240)	570
	f_MPOS	24270.538	234 (240)	553
	f_OPO	23589.475	232 (240)	547
tcheck (3 I/p, 3 O/p)	MSOP	1645.634	198 (210)	44
	f_MSOP	1338.585	161 (210)	27
	f_MPOS	1338.585	161 (210)	27
	f_OPO	1338.585	161 (210)	27
	MSOP	83572.388	525 (535)	2437

ts10 (22 I/p, 16 O/p)	f_MSOP	33341.773	526 (535)	996
	f_MPOS	43664.712	452 (535)	876
	f_OPO	33341.773	526 (535)	996
mish (94 I/p, 43 O/p)	MSOP	20699.700	244 (250)	494
	f_MSOP	22163.286	244 (250)	475
	f_MPOS	20050.019	244 (250)	436
	f_OPO	22162.972	244 (250)	475
4mod5 (4 I/p, 1 O/p)	MSOP	3034.076	203 (225)	63
	f_MSOP	1427.710	167 (225)	29
	f_MPOS	1427.710	167 (225)	29
	f_OPO	1427.710	167 (225)	29
5mod5 (5 I/p, 1 O/p)	MSOP	14253.435	268 (270)	356
	f_MSOP	6257.565	260 (270)	139
	f_MPOS	6288.855	262 (270)	134
	f_OPO	6257.565	260 (270)	139
dekoder (4 I/p, 7 O/p)	MSOP	24282.264	223 (232)	359
	f_MSOP	25025.168	227 (232)	376
	f_MPOS	21322.331	220 (232)	303
	f_OPO	19478.134	223 (232)	293

TABLE VI
INTERNAL POWER (IN NANOWATTS) FIGURES FOR LOGICAL FORMATS CORRESPONDING TO DIFFERENT MCNC/IWLS COMBINATIONAL BENCHMARKS

Benchmark	MSOP	f_MSOP	f_MPOS	f_OPO
newtag	1627.268	1013.019	704.878	704.878
misj	3639.273	1656.219	1854.402	1948.615
clpl	1855.954	3245.578	3245.578	3245.578
c17	781.714	522.873	504.338	463.796
con1	1375.080	1552.427	2557.660	1348.532
newtpla1	2153.416	913.325	913.325	913.325
arpanet	4107.035	2545.354	2744.537	2744.537
newtpla2	3040.960	1419.605	1881.435	2374.740
newapla1	2127.143	1352.817	1313.657	1313.657
newill	2724.141	3106.291	2875.813	3106.291
exam3_d	596.292	709.392	543.291	709.392
wim	2865.318	3211.682	2259.529	1948.524
t4	6342.107	7547.085	4307.347	3255.216
newcwp	3829.764	3078.089	3141.053	2122.828
dc1	4287.849	4792.015	5162.941	4516.699
alcom	7999.694	5933.423	6131.578	5551.675
tcheck	372.674	468.825	468.825	468.825
ts10	17614.988	9562.693	11260.032	9562.693
mish	5172.540	6600.306	6078.599	6599.992
4mod5	764.636	735.790	735.790	735.790
5mod5	4184.235	2036.565	1994.775	2036.565
dekoder	7316.814	8067.818	8278.091	5832.064

TABLE VII
NET POWER (IN NANOWATTS) VALUES FOR LOGICAL FORMATS CORRESPONDING TO DIFFERENT MCNC/IWLS COMBINATIONAL BENCHMARKS

Benchmark	MSOP	f_MSOP	f_MPOS	f_OPO
newtag	3956.400	2824.560	1716.120	1716.120
misj	9552.600	5976.360	6145.560	6189.480
clpl	5400.720	5853.600	5853.600	5853.600
c17	1918.800	1107.360	1180.080	1039.680
con1	3084.840	3968.280	5639.760	3518.280
newtpla1	5775.120	2815.560	2815.560	2815.560
arpanet	10761.120	5912.280	4867.200	4867.200
newtpla2	8583.840	4678.200	6174.000	6929.640
newapla1	7335.360	5022.720	5246.640	5246.640
newill	7863.480	6574.320	6569.640	6574.320
exam3_d	1590.480	1149.480	1426.320	1149.480
wim	6380.640	6810.840	4915.440	4872.600
t4	14550.120	15106.680	9960.480	7941.960
newcwp	7646.760	3705.120	5829.120	3962.520
dc1	9763.560	9402.120	11657.520	9828.360
alcom	18928.440	16513.200	18138.960	18037.800
tcheck	1272.960	869.760	869.760	869.760
ts10	65957.400	23779.080	32404.680	23779.080
mish	15527.160	15562.980	13971.420	15562.980
4mod5	2269.440	691.920	691.920	691.920
5mod5	10069.200	4221.000	4294.080	4221.000
dekoder	16965.450	16957.350	13044.240	13646.070

The internal power and net power figures corresponding to the different logical formats of various combinational benchmarks listed in Tables VI and VII are graphically illustrated in figures 1 and 2, to facilitate a quick comparison. From figure 1, we understand that MSOP yields a realization which is poorer than those resulting from other expressions in terms of internal power for 12 out of 22 cases, while figure 2 shows that MSOP leads to a synthesis solution which is poorer in terms of net power in comparison with that resulting from other initial expressions for 16 out of 22 cases. The latter is mainly due to the extensive number of standard cells required for the realization based on MSOP, which consequently increases the number of interconnects and thereby more net power dissipation. This is substantiated by the values of figure 3, wherein MSOP is found to yield a less area efficient solution amongst 18 of the 22 circuits considered. Situations exist where either f_MSOP/f_MPOS yield optimum solutions.

V. CONCLUSIONS AND SCOPE FOR FURTHER WORK

The effect of pre-logic factoring on combinatorial circuit synthesis of benchmark functions, based on standard library cells has been analyzed in this work; building up on an earlier work which addressed FPGA based logic design for simple arbitrarily chosen combinational logic. However, instead of representation of the combinational circuit functionality purely on the basis of a directed acyclic graph consisting of two-input AND and OR gates and inverter nodes in the earlier work, we proceed with a behavioral modeling of the circuit functionality considered for the sake of simplicity, based on the different logical expressions governing it, which are subsequently synthesized. It is also possible to proceed with the automated synthesis based on different initial graph representations [14] and this point to an altogether different direction for further research. The application of algebraic factoring operation was on each and every individual output comprising a benchmark. Overall, the experimental results demonstrate significant savings in terms of power and area, with the synthesis targeting performance. It can be inferred from the results mentioned in Table V that simultaneous power, delay and area optimization has been feasible in many cases with relative ease, notably in case of *ts10* which exhibits maximum power dissipation among the different functions considered. A similar study can be undertaken based on the Boolean factoring approach which might enable better synthesis solutions but, as is well known, Boolean factoring scheme is relatively complex and is more computationally intensive than an algebraic factorization procedure. Alternatively, it might be of interest to study the impact that multi-level logic realizations could have on an automated logic synthesis process and this has been reserved for future work.

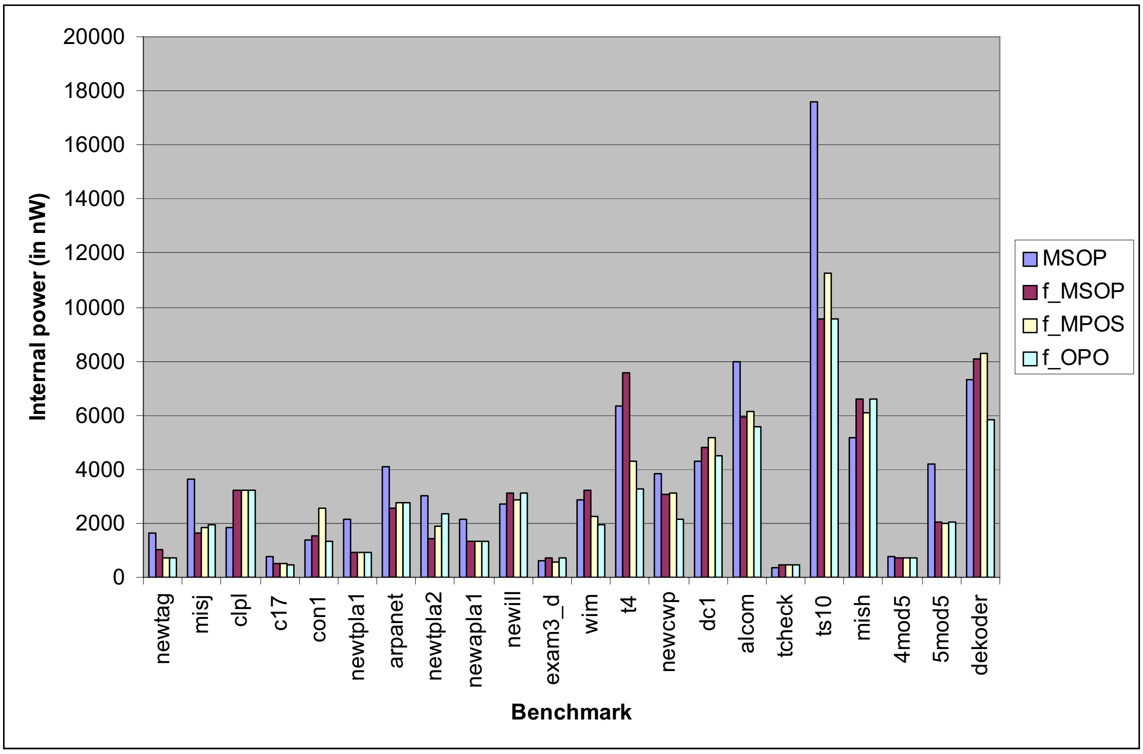

Fig. 1. Graphical sketch of internal power component corresponding to different logical formats of the combinatorial benchmarks

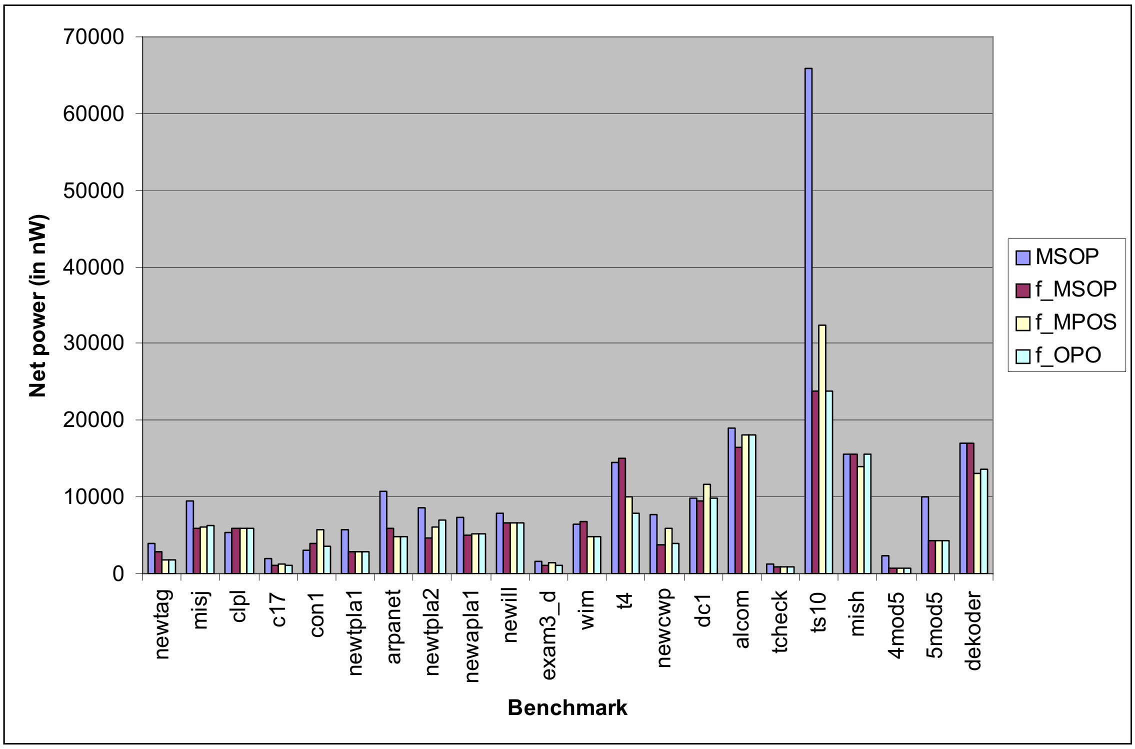

Fig. 2. Graphical sketch of net power component corresponding to different logical formats of the combinatorial benchmarks

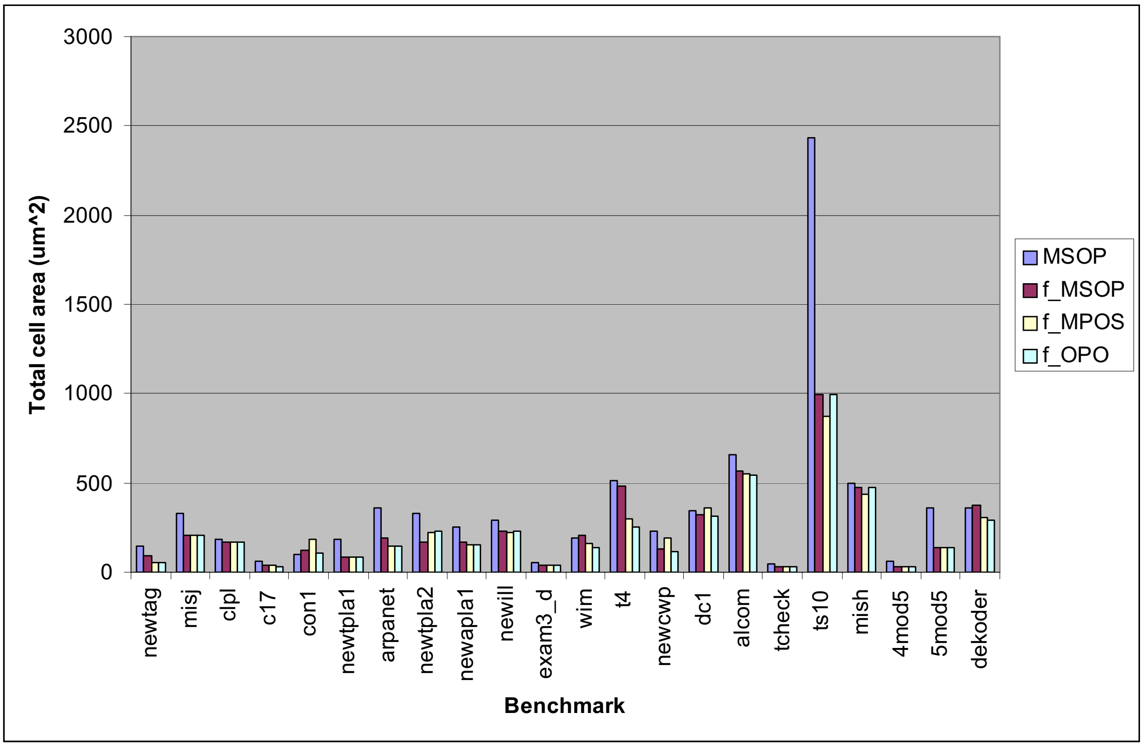

Fig. 3. Graphical plot of total cell area pertaining to different logical formats of the combinatorial benchmarks

VI. Acknowledgment

This work was performed when the co-author (second author) was affiliated with the School of Electrical Sciences, Vellore Institute of Technology (University and IET, UK Accredited), Vellore, TN, India.

References

[1] R. Ashenhurst, “The decomposition of switching functions,” *Proc. International Symposium on Switching Theory*, 1959, pp. 74-116.

[2] R. Brayton and C. McMullen, “The decomposition and factorization of Boolean expressions,” *Proc. International Symposium on Circuits and Systems*, 1982, pp. 49-54.

[3] R.K. Brayton, “Factoring logic functions,” *IBM Journal of Research and Development*, vol. 31, no. 2, March 1987, pp. 187-198.

[3] R. Rudell, “Logic Synthesis for VLSI Design”, *PhD thesis*, University of California, Berkeley, 1989.

[5] J. Vasudevamurthy and J. Rajski, “A method for concurrent decomposition and factorization of Boolean expressions,” *Proc. International Conf. on Computer-Aided Design*, 1990, pp. 510-513.

[6] G. Caruso, “Near optimal factorization of Boolean functions,” *IEEE Trans. on CAD*, vol. 10, no. 8, August 1991, pp. 1072-1078.

[7] R.K. Brayton, G.D. Hachtel and A.L. Sangiovanni-Vincentelli, “Multilevel logic synthesis,” *Proc. of the IEEE*, vol. 78, no. 2, February 1990, pp. 264-300.

[8] S. Yang, “Logic synthesis and optimization benchmarks User guide version 3.0,” *MCNC Research Triangle Park*, NC, January 1991.

[9] K. McElvain, “IWLS ’93 Benchmark set: version 4.0,” *distributed as part of the MCNC International Workshop on Logic Synthesis*, 1993.

[10] 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, no. 1, Winter 2007, pp. 19-29.

[11] 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, no. 3, Summer 2007, pp. 188-198.

[12] R.K. Brayton, A.L. Sangiovanni-Vincentelli, C.T. McMullen and G.D. Hachtel, *Logic Minimization Algorithms for VLSI Synthesis*, Kluwer Academic Publishers, Norwell, MA, USA, 1984.

[13] J. Cortadella, “Timing-driven logic bi-decomposition,” *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, vol. 22, no. 6, June 2003, pp. 675-685.

[14] P. Balasubramanian, R.T. Naayagi, A. Karthik and B. Raghavendra, “Evaluation of logic network representations for Achilles heel Boolean functions,” *International Journal of Computers, Systems and Signals*, vol. 9, no. 1, 2008, pp. 42-55.