Finite State Machine State Assignment for Area and Power Minimization

Aiman El-Maleh, Sadiq M. Sait and Faisal Nawaz Khan
Department of Computer Engineering
King Fahd University of petroleum & Minerals, Saudi Arabia
Email: {aimane, sadiq, fnawaz} @ccse.kfupm.edu.sa

Abstract— **In this paper, we address the problem of FSM state assignment to minimize area and power. The objectives are targeted as single/independent as well as multi-objective optimization (MOP) problems. Novel methods for estimating area and power of an FSM are presented. A fuzzy-based aggregation function is employed to combine the two objectives. The work employs genetic algorithm for search space exploration. Experimental results demonstrate the effectiveness of the proposed measures.**

I. Introduction

State assignment (SA) for Finite State Machine (FSM) is one of the main optimization problems in the synthesis of sequential circuits. The SA of an FSM determines the complexity of its combinational circuit and thus area and power dissipation of the implementation. State assignment involves an injective mapping f: S $\longrightarrow B^n$ where n is the code length $(n \geq \lceil \log_2 |S| \rceil)$ and B^n is an n-dimensional Boolean space, a Boolean hypercube.

The objective of state assignment varies depending on whether the targeted implementation is two-level or multilevel. The focus of this work is towards minimizing multilevel FSM implementation.

The complexity measure for multilevel circuits is the number of literals in the optimized logic network assuming minimum encoding length. Literal savings by extracting common subexpressions has been the focus of most of the work done for multilevel FSM optimization. This involves finding pairs of states that when encoded can result in extraction of common subexpressions. In contrast to two-level circuits, state pairs in multilevel implementations do not necessarily have to be given adjacent codes for literal savings [1]–[3].

The problem of multilevel area minimization of an FSM has been modeled as weighted [1]–[3] or constrained [4]–[6] graph-embedding problem on Boolean hypercubes.

Portable electronics applications have given power-aware computing a whole new importance. This is due to limitations in battery capacities and to the fact that progress in their technologies trail far behind the ever increasing computing requirements. Power consumption is thus constrained and optimized at all levels of design hierarchy [7]. The major source of power consumption in CMOS circuits is due to charging and discharging of circuit capacitances. Power consumption in a circuit can be reduced by either reducing the total switching in the logic or by reducing the switching capacitance or both of them.

There has been much interest in power reduction strategies for FSM. Most of the work reported in literature [8]–[13] tries to minimize total switching on the flip-flops. This is done by formulating the problem as a graph embedding problem where edges between a state pair indicate the steady state transition probability between them. The problem is thus reduced to minimizing the total transitions between the states. Such a formulation is referred to as *Minimum Weighted Hamming Distance* (MWHD).

The hard nature of SA problem has generated considerable interest in the use of non-deterministic heuristics such as genetic algorithm [3], [11] and simulated annealing [1], for its design automation. These algorithms are capable of efficiently searching for a near optimal solution in a large solution space and have been very successful in solving a number of combinatorial optimization problems in various disciplines of science and engineering.

In this paper, we explore the use of genetic algorithm for SA problem where the logic synthesized is implemented as multilevel circuit. We present novel and efficient strategies for estimating multilevel area and power dissipation in an FSM. We also evaluate several integration mechanisms for combining area and power measures.

II. Proposed Methodology

A. State Assignment for Area

The contemporary approach employed towards multilevel FSM area minimization is by using weighted-graph approach where weights between edges of states define the relative proximity in assignment (affinity).

Affinity cost as modeled in adjacency graphs is then used to minimize Equation 1:

$$\sum_{i=1}^{n_s} \sum_{j=1}^{n_s} A^P_{S_i,S_j} . \Delta(S_i, S_j) \qquad (1)$$

where $\Delta(S_i, S_j)$ is the Hamming distance between codes of states S_i and S_j, n_s being number of states, and $A^P_{S_i,S_j}$ being the affinity between two states as given by the used cost measure.

In [14], several literal saving measures based on weighted-graph embedding problem including those based on Jedi

[1], Mustang [2] and Armstrong [15] cost measures were investigated. It was found that all these measures weakly correlate with the actual literal savings measure.

There are several complications in using weighted graph approach for literal savings measure estimation based on common cube extraction. First, the common cubes interact with each other, annulling certain predicted savings. Second, such measures optimize relative literal savings rather than absolute number of literals in an implementation. Consequently, they may lead the search to solutions having higher relative savings with higher implementation costs. Such an interaction is very difficult to predict at higher level of abstractions.

In order to use an efficient but accurate estimate of actual literal savings, we propose the use of EXPAND function, employed in ESPRESSO [16], as a measure for estimating area of an FSM. EXPAND is the first step employed in two-level logic minimization and provides a cover for the function that is minimal with respect to single-cube containment. However, the cover might contain some redundant terms. It should be observed that ESPRESSO [16] is based on iterative application of several functions including EXPAND, IRREDUNDANT and REDUCE. Hence, using EXPAND as a cost measure is more efficient than using ESPRESSO. As will be illustrated by experimental results in Section III, it is found that using EXPAND with single output optimization strongly correlates with actual multilevel literal savings measure.

B. State Assignment for Low Power

Switching activity in combinational logic of the controllers is due to logic transitions on the flip-flops as well as on primary inputs. Thus, to reduce power dissipation in an FSM, one can:

1) Minimize switching activity at the flip-flops.
2) Minimize the capacitance on flip-flops being switched, i.e., fanout branches (fanout) from flip-flops.
3) Minimize the combinational logic being switched.

State assignment for power minimization can also be modeled by a weighted graph where weights between edges of states represent the total switching probability between the two states. By assigning shorter distance codes to states connected with higher weights, i.e., higher transition probability, the overall switching on the state lines of the FSM can be minimized. Thus, a cost model for minimizing power consumption can be to have *Minimum Weighted Hamming Distance* (MWHD). Mathematically, this can be achieved by minimizing Equation 1 with the affinity being modeled as:

$$A^P_{S_i,S_j} = P_{ij} + P_{ji} \tag{2}$$

where P_{ij} is the state transition probability from state S_i to state S_j.

The first formulation for power minimization used in this work tries to optimize the above MWHD cost. MWHD approach tries to minimize the total transition probability of the state machine in the hope that the total number of logic transitions in the synthesized circuit will also get reduced, i.e., it tries to maximize power reductions mainly due to point 1 discussed above.

In this work, we propose a new cost function for minimizing power dissipation in an FSM. The idea is based on minimizing fanout branches out of frequently switched sequential elements. Reducing fanout branches on highly switching flip-flops reduces the switching capacitance and hence reduces power dissipation. The transition probabilities of sequential elements are computed based on steady state transition probabilities computed from the state table. The flip-flops' transition frequencies are weighted with their respective fanout counts for determining minimum weighted fanout (MWF) solution by minimizing Equation 3:

$$MWF = \sum_{i=1}^{n} T_i B_i \tag{3}$$

where B_i and T_i are the number of fanout branches and the transition frequency of flip-flop i, respectively. Expand cover is used in estimating flip-flop fanout branches.

C. Genetic Algorithm

Genetic algorithms have been adopted to explore the solution space in search of good state assignments. The chromosomal representations employed and the crossover operator used is similar to the one suggested by Amaral et al. [3]. In this representation, each state code is described as an array of bits equal to the number of storage elements required. Parent selection for crossover is based on the roulette wheel method. Crossover is performed by randomly selecting a subset of state encoding columns from the first parent and the rest from the second parent. The mutation operator used swaps two randomly selected states in a randomly selected parent. The number of mutations occurring in a generation is controlled by the mutation rate. A value of 20% is selected and mutation is applied to all but the best solution.

After every generation, members for the next generation are selected from both parents and newly generated offsprings. A combination of greedy (for the best half of parents and new offsprings) and random from the second half is used. Diversity in population is maintained by discarding duplicate offsprings. In this work, a population size of 64 and a maximum generation size of 350 are used.

The area and power objectives are aggregated using product based and Ordered Weighted Averaging (OWA) operator [17]. In OWA, we employ both "orlike" (Max) and "andlike" (Min) fuzzy operators as given in Equation 4:

$$\mu \; = \; \beta \times O(\mu_a, \mu_p) + (1-\beta) \times \frac{1}{2}(\mu_a + \mu_p) \tag{4}$$

where O is max/min type fuzzy operator, μ_i represents cost for area or power objective and β is a constant parameter that represents the degree to which OWA operator resembles a pure "or" or pure "and", respectively. In this work we employ $\beta = 0.5$

TABLE I

COMPARISON OF EXPAND-SO MEASURE WITH OTHER AREA MINIMIZATION HEURISTICS.

Benchmark	EXPAND-SO	ESPRESSO-SO+FX	EXPAND-MO	Jedi ([1])	Nova	Mustang [2]	Armstrong([3])
bbara	56(52)	51	57	73(57)	57	64	59(86)
bbsse	110(105)	100	120	134(111)	140	106	127(180)
cse	198(228)	183	239	240(200)	214	206	220(NA)
dk14	104(86)	101	115	108(76)	111	117	124(NA)
donfile	87(72)	68	106	82(76)	154	160	171(NA)
ex2	78(68)	66	130	123(122)	127	119	131(NA)
ex3	56(48)	53	67	65(66)	71	71	68(NA)
keyb	199(205)	161	161	260(140)	201	167	334(NA)
lion9	11(11)	10	25	19(13)	27	17	27(21)
planet	486(436)	469	557	603(547)	591	544	607(NA)
pma	165(152)	153	189	263(NA)	241	NA	218(NA)
s1	227(105)	155	285	282(152)	340	183	291(NA)
s1494	570(624)	543	717	679(NA)	715	NA	696(NA)
s832	231(218)	215	307	357(NA)	274	NA	301(NA)
sand	498(494)	473	514	554(437)	558	462	619(NA)
shiftreg	2(2)	2	4	2(2)	2	2	2(10)
styr	419(429)	423	466	518(508)	502	546	546(NA)
tbk	353(268)	312	493	305(278)	365	547	711(NA)
traian11	22(20)	18	29	34(27)	32	37	32(47)
%Improvement		-8.89	15.48	17.63(6.51)	18	22.56	26.72(41.57)

III. EXPERIMENTAL RESULTS

In this section, experimental results of the proposed state assignment measures for area and power minimization are presented. Experiments are performed on a subset of MCNC-93 FSM benchmark circuits with different complexities.

A. Area

In Table I, we compare the performance of several cost functions targeting area minimization including Espresso with single-output minimization followed by FX (ESPRESSO-SO+FX), Expand with single-output minimization (EXPAND-SO), Expand with multiple-output minimization (EXPAND-MO), Jedi [1], Nova [6], Mustang [2], and Armstrong [3].

The values in the table represent the literal count obtained after synthesizing the obtained solutions using ESPRESSO-SO+FX. Results shown in brackets are those reported in literature based on synthesis using *script.rugged* except for Amaral et al. [3] where results are reported in factored form.

The last row in the table shows the percentage improvement of EXPAND-SO over all other cost measures. It is observed that EXPAND-SO achieves significant improvements over all other techniques and only lags behind the accurate cost measure using ESPRESSO-SO+FX by nearly 9%.

B. Power

The performance of the proposed MWF measure is next compared with MWHD and default Jedi state assignment algorithm in Table II. Power consumptions are measured in microwatts using *power_estimate -t SEQUENTIAL* option of SIS and assuming default conditions. It can be seen that the MWF measure achieves better results than MWHD and Jedi on both area and power, and on average it achieves 6.47% less power and 14.53% less area, per circuit, than Jedi algorithm.

The multiobjective optimization of both area and power is shown by integrating the Expand-SO area estimate with MWF measure. We employ both the product of area and power estimates (*MWFA*) as well as fuzzy-based aggregations, employing *Max* and *Min* type fuzzy operators using equal weighted OWA aggregation. As can be seen from the results, the multiobjective optimization of both area and power achieves overall higher reduction in both area and power with MWF(Fuzzy-Max) resulting in the least power dissipation.

C. Literature Comparison

In Table III, we compare the performance of the MWF(Fuzzy-Max) measure to other techniques reported in literature [10]–[13]. Comparison is made relative to the output-oriented (default) Jedi state assignment algorithm, with results reported as %reduction relative to Jedi. The last row in the table shows the percentage improvement achieved by our technique over other techniques.

It can be seen that our approach achieves higher overall savings over all the approaches. The MWF(Fuzzy-Max), though being slightly better in power dissipation, significantly outperforms Pedram's approach [10] in area savings. Significant power savings are also observed between our approach and Ciesielski et al. [12] and IITG8 [13] measures. The ingenuity of our measure can also be seen from its complete dominance over Almiani et al.'s approach [11] in both area and power optimizations, although the latter employs a more expensive Espresso iteration in their cost calculation.

IV. CONCLUSION

In this work, we have presented a genetically engineered state assignment solution for area and power minimization. We have proposed efficient cost functions that highly correlate with the actual literal count and power dissipation of a multilevel circuit implementation. Experimental results demonstrate the effectiveness of the proposed measures in achieving lower area and power dissipation solutions in comparison to techniques reported in the literature.

ACKNOWLEDGMENT

This work is supported by King Fahd University of Petroleum & Minerals under Project#FT2005-63.

TABLE II

AREA AND POWER DISSIPATION FOR VARIOUS HEURISTICS.

	MWHD		MWF		Jedi		MWFA		MWF(Fuzzy-Max)		MWF(Fuzzy-Min)	
Benchmark	Power	Area	Power	Area	Power	Area	Power	Area	Power	Area	Power	Area
bbara	214.7	82	150.5	55	187.7	74	181.2	65	181.2	58	181.2	58
bbsse	446.1	140	412.2	122	538.8	149	394.5	118	437.1	123	448.2	128
cse	528.9	217	424.8	211	495.8	251	391.3	209	455.2	205	459.9	210
dk14	661.2	140	561.4	103	714.4	157	561.4	103	551.3	101	579.2	115
donfile	895.9	206	513.7	109	380.8	89	474.1	100	355.2	82	295.8	75
keyb	655.3	263	645	237	767.6	260	517.3	215	535.6	210	511.7	192
lion9	142	20	116.7	19	145.6	19	100.8	15	105.3	16	129.9	11
planet	1788.6	656	1795.1	553	2001.5	675	1889.7	510	1702.5	470	1843.2	465
pma	653.4	198	778	180	883.7	236	693.1	165	651.2	155	675.2	145
s1	1165.1	406	766.5	187	1205.3	353	771.4	197	751.2	191	625.8	161
s1494	1376.3	734	1553.1	625	1668.9	679	852.4	569	838.2	530	1025.5	505
s832	922.1	368	677.5	271	1068.4	376	665.2	260	650.2	242	621.5	249
sand	1645.5	599	1541.4	559	1458.9	651	1617.2	585	1289.9	490	1352.2	498
shiftreg	163.3	27	96.3	2	132.5	9	98.8	4	96.3	2	96.3	2
styr	1277.5	540	1062.9	431	1118.6	567	1086.8	453	1022.3	432	1100	376
tbk	1682	630	1589.3	488	721.2	305	1766.6	556	1095.2	422	1350	398
train11	180.4	38	136.3	23	218.2	35	142.4	22	122.2	23	150.1	21
Average	846.96	309.65	754.16	245.59	806.35	287.35	717.89	243.88	637.65	220.71	673.28	212.29

TABLE III

POWER AND AREA %-REDUCTION COMPARISON WITH JEDI.

	MWF(Fuzzy-Max)		Pedram [10]		Ciesielski [12]	IITG8 [13]	Almaini [11]	
	Power	Area	Power	Area	Power	Power	Power	Area
bbara	3.46	21.62	17.97	-10.14		16.07	-25.68	21.62
bbsse	18.88	17.45	18.37	6.56	5.66			
cse	8.19	18.33	12.15	-1.41		18.48	2.58	18.33
dk14	22.83	35.67	4.92	-0.98		16.19		
donfile	6.72	7.86	6.22	22.64			-5.57	7.86
keyb	30.22	19.23			35.56	20.87	2.53	19.23
lion9	27.68	15.79						
planet	14.93	30.37					-19.22	30.37
pma	26.3	34.32						
s1	37.68	45.89				-22.46	-7.32	45.89
s1494	49.78	21.94			6.89			
s832	39.14	35.64			7.75	26.68		
sand	11.58	24.73	10.52	16.12			-19.29	-30.32
shiftreg	27.32	77.78				-29.08		
styr	8.61	23.81				-9	9.16	23.81
tbk	-51.86	-38.36			5.03			
train11	44	34.28				11.61	13.2	34.29
%Improvement			2.09	73.9	29.31	74.74	130	24.36

REFERENCES

[1] B. Lin and A. R. Newton. Synthesis of multiple-level logic from symbolic high-level description languages. *IFIP International Conference on Very Large Scale Integration*, pages 187–196, August 1989.

[2] S. Devadas, H.T. Ma, A.R. Newton, and Sangiovanni-Vincentelli. MUSTANG: State Assignment of Finite State Machines for Optimal Multi-Level Logic Implememations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 7:1290–1300, Dec. 1988.

[3] J. Amaral, K. Turner, and J. Ghosh. Designing Genetic Algorithm for State Assignment Problem. *IEEE Trans on SMC*, 25:659–694, 1995.

[4] G. DeMicheli, R. K. Brayton, and A. Sangiovanni Vincenteli. Optimal state assignment for Finite State Machines. *IEEE Transaction on Computer Aided Design*, CAD-4:3, 269–285, 1985.

[5] X. Du, G. Hachtel, B. Lin, and A. R. Newton. MUSE: A Multilevel symbolic encoding algorithm for state assignment. *IEEE Trans. Computer Aided Design*, CAD-10:28–38, 1991.

[6] T. Villa and A. Sangiovanni-Vincentelli. Nova: state assignment of finite state machines for optimal two-level logic implementations. *Proceedings of the 1989 26th ACM/IEEE conference on Design automation conference*, pages 327–332, 1989.

[7] M. Pedram. Design technologies for low power VLSI. *In Encyclopedia of Computer Science and Technology, Marcel Dekker, Inc.*, pages 73–96, 1997.

[8] L. Benini and G. DeMicheli. State encoding for low power embedded controllers. *IEEE Journal of Solid-State Circuits*, 30:258–268, 1995.

[9] E. Olson and S. M. Kang. State assignment for low-power FSM synthesis using genetic local search. *IEEE Custom Integrated Circuits Conference*, pages 140–143, 1994.

[10] Chi-Ymg Tsui, M. Pedram, Chih-Ang Chen, and A. M. Despain. Low-Power State Assignment Targeting Two And Multi-level Logic Implementations. *IEEE/ACM International Conference on Computer-Aided Design*, pages 82–87, 1994.

[11] Y. Xia and A. E. A. Almaini. Genetic algorithm based state assignment for power and area optimisation. *IEEE Proceedings Computers and Digital Techniques*, 149:128–133, 2002.

[12] Sangju. Park, Sangwook Cho, Seiyang Yang, and Maciej Ciesielski. A New State Assignment Technique for Testing and Low Power. *Annual ACM IEEE Design Automation Conference*, pages 510–513, 2004.

[13] S. Chattopadhyay and P. N. Reddy. Finite state machine state assignment targeting low power consumption. *IEEE Proceedings Computers and Digital Techniques*, 151:61–70, 2004.

[14] Faisal N. Khan. FSM State Assignmnet for Area, Power and Testability using Non-Deterministic Evolutionary Heuristics. Master's thesis, KFUPM, Saudi Arabia, 2005.

[15] D. B Armstrong. A programmed algorithm for assigning internal codes to sequential machines. *IRE Transactions on Electronic Computers*, pages 466–472, 1962.

[16] Robert K. Brayton, Gary D. Hachtel, Curtis T. McMullen, and Alberto L. Sangiovanni-Vincentelli. *Logic Minimization Algorithms for VLSI Synthesis*. Kluwer Academic Publishers, 1984.

[17] R. Yager. Multiple Objective Decision-making Using Fuzzy Sets. *International Journal of Man-Machine Studies*, pages 9:375–382, 1977.