

State Assignment for Finite State Machine Synthesis

Meng Yang

State Key Lab of ASIC and Systems, Fudan University, Shanghai, China

Email: mengyang@fudan.edu.cn

Abstract—This paper proposes simulated annealing based algorithm for the synthesis of a finite state machine to determine the optimal state assignment with less area and power dissipation. The algorithm has two annealing stages. In the first rough annealing stage it tries to search in global scope by the proposed rough search method. In the second focusing annealing stage it tries to search in local scope by using proposed focusing search methods intending changing solution slightly. In both stages, the experience of past solution is utilised by combing the best solution in the past and the current solution. The experiments performed on a large suite of benchmarks have established the fact that the proposed method outperforms the published GA-based algorithms. The results have shown the effectiveness of the proposed method in achieving optimal state assignment for finite state machine.

Index Terms—state assignment, finite state machine, optimisation algorithm, simulated annealing algorithm

I. INTRODUCTION

As the mobile applications are emerging, the power consumption of the circuits has become a major concern. Numerous researches have been investigated concerning the power issues [1-3]. Finite state machine (FSM) is mathematical model of the sequential circuits with discrete inputs, internal states and discrete outputs. The problem of finding an optimal state assignment is NP-hard [4]. As a result, the synthesis of an FSM plays an important role.

The genetic algorithm (GA) technique [5] has been successfully applied to a variety of computationally complex problems since it has a large search space. Many investigations have shown that GA can find good state assignments. Almaini, et al [6] have demonstrated that the GA method produced significantly simpler solutions. In [7] multi objective GA (MOGA) has been used to optimise area and power simultaneously. Xia and Almaini [8] have used GAs to optimise both area and power with good tradeoffs. Pradhan, et al [3] report on the application of power gating in the higher level of system design in the form of finite state machine (FSM) synthesis. In [9] Chattopadhyay has used GAs to obtain power optimised two- and multilevel FSM realisations. Chattopadhyay [10] considers D flip-flops to store the state bits and investigates the avenue of GAs to achieve area reduction under flip-flop and output polarity selection.

Other than GA based methods, a number of heuristic algorithms have been proposed, which are based on different cost functions estimating the effect of state assignment on logic minimisation. It has shown a new comprehensive method in [11] consisting of an efficient state minimisation and state assignment technique. Goren, et al [12] present a heuristic for state reduction of incompletely specified finite state machines. The proposed heuristic is based on a branch-and-bound search technique and identification of sets of compatible states of a given incompletely specified finite state machine specification. In [13], the usage of a stochastic search technique inspired by simulated annealing is explored to solve the state assignment problem.

Generally speaking, it is relatively easy to find state assignments to minimise the area only or the power dissipation only. However, it is known that minimisation of either the power or logic complexity could be at the expense of the other and in most cases it is hard to find a solution that is optimum in both domains. For large circuits, there are millions or possibly billions of assignments [14] and hence to find the state assignment for the minimisation of power consumption and area at the same time is computationally difficult. Besides, GA selects the next generation via a ranking system, which is not always necessary but takes significant runtime. In this paper, in order to reduce the computational time but at the same time retain the quality if the solution, simulated annealing based algorithm is proposed to solve FSM problem with low-power and small-area requirements. The remainder of the paper is organised as follows. Section II gives terminology of state assignment of the FSM. The two annealing stage simulated annealing approach is given in Section III. Section IV discusses the comparison results in details with respect to other approaches. Conclusions are then given in Section V.

II. TERMINOLOGY

An FSM can be characterised by a 5-tuple (I, O, M, X, Y) where I and O are the sets of primary inputs and primary outputs, M is the internal states, X and Y are the output and the next state functions, respectively. An FSM with M states requires a minimum of S state variables for the assignment, where $S = \lceil \log_2 M \rceil$ and $\lceil g \rceil$ is the smallest integer equal to or greater than g . The number of logically unique assignments for an FSM N is given as follows.

$$N = \frac{(2^S - 1)!}{(2^S - M)!S!} \tag{1}$$

The FSM is commonly represented by a state transition table (STT). Given the input transition probability of an FSM, its state transition probability can be computed. The state transition probability tp_{ij} between state s_i and state s_j occurs in an arbitrarily long sequence and is given as follows.

$$tp_{ij} = P_i p_{ij} \tag{2}$$

The steady state probabilities P_i can be obtained via Gaussian elimination methods to solve the set of linear equations as in (3) and (4).

$$\sum_{i=0}^{M-1} P_i = 1 \tag{3}$$

$$P_i = \sum_{j=0}^{M-1} P_j p_{ji} \tag{4}$$

The power consumption [15] of a sequential circuit is proportional to its switching activity which can be represented as in (5), where C is the capacitance of the output for the node, V_{dd} is the supply voltage, f_{clk} is the clock frequency and E is the expected switching activity.

$$P = \frac{1}{2} CV_{dd}^2 f_{clk} E \tag{5}$$

Since the register capacitance is fixed and cannot be affected, therefore the E is considered as part of the cost function in terms of power consumption.

$$E = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} tp_{ij} dist_{i,j} \tag{6}$$

where $dist_{i,j}$ represents the hamming distance between the coding of state s_i and state s_j and tp_{ij} is the total state transition probability from state s_i and state s_j as defined in (2).

III. SIMULATED ANNEALING BASED ALGORITHM

The problem of finding the state assignment for the minimisation of power consumption and area is computationally difficult. GA has been successfully applied to a variety of computationally difficult problems. It has been shown that it can produce good results in a reasonable computation time. The basic idea of GA is initially to generate a breeding pool of potential solutions to a problem. These solutions are encoded as chromosomes, and each chromosome is subjected to an evaluation function which assigns fitness depending on the quality of the solution it encodes. Existing solutions are recombined by crossover operator. Mutation operator makes random changes in a few randomly selected chromosomes in order to prevent premature convergence by maintaining the diversity of the population. The GA uses a tournament selection method via population ranking system to reproduce new generation. As a result,

the ranking system itself takes significant computational time. However, in most cases only the best solution at last is required, therefore the entire ranking is not always necessary. Furthermore, if some parents are not excellent enough, these solutions have little chance to join the competition in GA.

Simulated annealing (SA) algorithm on the other hand is another method to solve problems such as 3D packing problem [16] and single container loading problem [17]. It starts with high temperature. The inspiration of the algorithm demand an interesting feature related to the temperature variation to be embedded in the operational characteristics of the algorithm. This necessitates a gradual reduction of the temperature as the simulation proceeds. The algorithm starts initially with a high value temperature, and then it is decreased at each step following some annealing schedule. However there are several significant disadvantages of traditional SA, such as its slow search speed. Hence, in order to overcome the mentioned shortcomings, following simulated annealing based algorithm is proposed.

A. Solution Representation

Let the state-code array for an M -state FSM be $\langle S_0, S_1, \dots, S_{M-1} \rangle$, where M is the number of FSM states. The solution representation is an array of size equal to the number of states in the FSM. Each entry in the array is an integer between 0 and $M-1$. Take a ten-state machine $\langle S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9 \rangle$ for example. One possible representation is 9, 6, 8, 2, 3, 0, 4, 1, 5, 7, which is shown in Figure 1.

9	6	8	2	3	0	4	1	5	7
---	---	---	---	---	---	---	---	---	---

Figure 1. State assignment for ten-state machine.

B. Rough Annealing Stage

The proposed simulated annealing is designed purposely into two stages as the temperature schedules. In the rough annealing stage, it tends to alter the solution with bigger changes at high temperature, consequently escape local optima. The rotation swap method randomly selects the cutting line from the representation. The cutting line is treated as a mirror. Copy the state assignments in the left part of the mirror to the right part of the mirror. Similarly, the right part is copied to the left part. Figure 2 shows an example of exchange method.

C. Focusing Annealing Stage

The focusing annealing stage differs to the previous annealing stage, which tends to alter the solution with smaller changes at low temperature, consequently to achieve an effective convergence. In this stage, it randomly selects a state assignment and its neighbouring state assignment from the representation and swaps these two state assignments. These two states can be close to each other, as shown in Figure 3, or apart from each other, as shown in Figure 4.

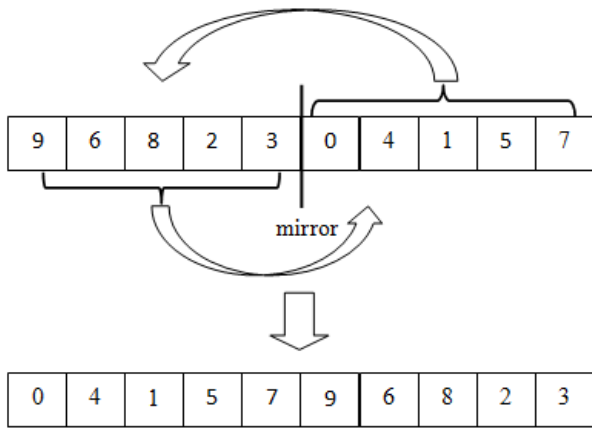


Figure 2. Exchange method.

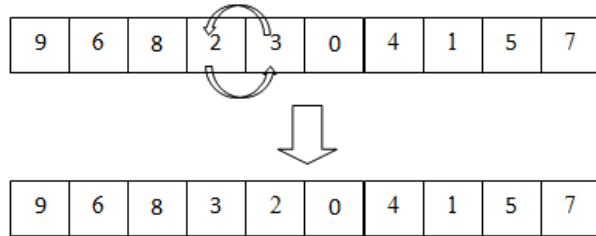


Figure 3. Swap method for two states close to each other.

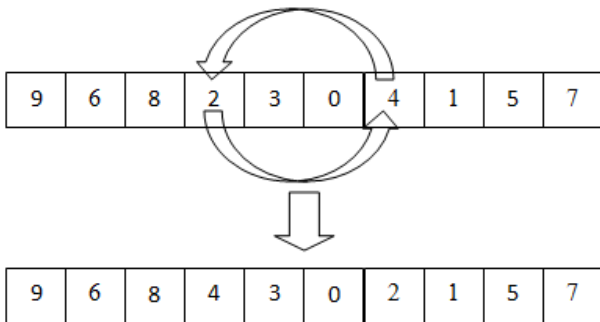


Figure 4. Swap method for two states apart from each other.

D. Crossover

Crossover operator is designed to escape the local optimum in the GA. In this paper crossover operator is adapted in the simulated annealing to generate a new solution by combining the best solution with the current solution. Consequently, some potential parts of the solution can be inherited from the previous solution.

It operates as a position-based crossover [18]. An array of binary bits is initialised randomly, in which the length of the array equals to the length of the representation. When 1s appear in the array, copy the states from the best solution to new solution. When 0s appear in the array, copy the states from the current solution to new solution. Since each entry of state assignment solution is unique, continuous check is required avoiding the repetition of the same states which is not allowed. If state entry is duplicated, the first unassigned state entry is assigned. Figure 5 shows an example of relay operator to generate

new solution. State entry 9, 2, 3, and 7 from the best solution form part of new solution. State entry 8, 3, 9, 2, 4 and 6 from current solution form the other part of new solution. Modification of the new solution is carried out to make sure that duplicated state entry 2, 3, and 9 do not appear in the solution. By doing so, unassigned state entry 0, 1 and 5 are filled in.

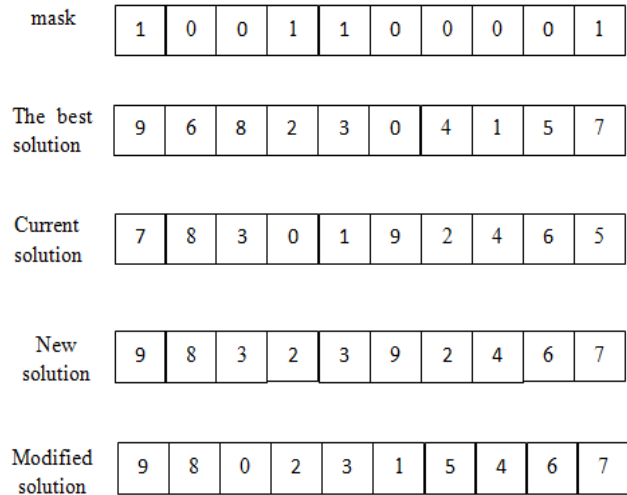


Figure 5. Crossover operator.

E. Cost Function

By minimising (6), low power dissipation could be achieved. Unfortunately, this may lead to area overhead, resulting in power overhead in the combinational part of the circuit. Therefore, objectives of area and power should be optimised simultaneously. The total cost function is $C_{total} = \gamma C_{area} + (1-\gamma) E$, where C_{area} is area function, E is power function and γ is a parameter specifying the tradeoff of E with respect to C_{area} .

F. Proposed Algorithm

To begin with, an initiation solution is randomly produced. The annealing approach is divided into rough annealing and focusing annealing, which is identified by Temperature threshold $T_{threshold}$. In the rough annealing stage the temperature is high and uses exchange method. In the focusing annealing stage the temperature is low and uses swap methods. The solution is evaluated by the cost function. The solution is accepted according to a probability $P = \exp(-\Delta C/T)$, where ΔC is the difference cost between new solution and current solution, T is the temperature. P is between 0 and 1. If the solution is improved and kept according to P , the new solution will be compared with the best recorded solution to decide whether updating the best record is necessary. If new solution is rejected instead, this solution is then ignored and reverses the current solution. Outer loop updates the temperature. The temperature schedules at each outer iteration. In the inner loop it generates possible solutions at each temperature. The algorithm terminates when reaching the lowest temperature and the output will be the best record. The pseudo code is shown in Algorithm 1.

Algorithm 1: Two stage Simulated Annealing

Inputs: Temperature coefficient α
 Number of state variables N
 Exit temperature T_0
 Temperature threshold $T_{\text{threshold}}$
 Outputs: The best solution S_{best}
 begin
 initialise a solution S_{current} and temperature T
 while ($T < T_0$) { // outer loop
 $k = 0$
 while ($k < N^3$) { // inner loop
 if ($T > T_{\text{threshold}}$) $S_{\text{new}} =$ generate solution via exchange method
 else
 $S_{\text{new}} =$ generate solution via swap methods
 $k = k + 1$
 $r = \text{random}(0, 1)$
 $\Delta C = \text{cost}(S_{\text{new}}) - \text{cost}(S_{\text{current}})$
 if ($r < \exp(-\Delta C/T)$) {
 $S_{\text{current}} = S_{\text{new}}$
 if ($\text{cost}(S_{\text{best}}) > \text{cost}(S_{\text{current}})$)
 $S_{\text{best}} = S_{\text{current}}$
 }
 } // end of inner loop
 $T = \alpha T$
 } // end of outer loop
 Output the best solution S_{best}

IV. EXPERIMENTAL RESULTS

The algorithm has been implemented in C++ and the results have been obtained on a personal computer with an INTEL CPU 2.4 GHz and 4 GB RAM. A Gaussian elimination method is used to find the total transition probabilities according to the STT of an FSM. ESSPRESSO is used to minimise the circuit for each state assignment. The product terms from this minimisation determine the number of cubes for that assignment. Switching activity is calculated by (6). T , $T_{\text{threshold}}$ and T_0 are set to 10000, 100 and 0.1, respectively. In order to balance the required solution quality and computational time, temperature schedule coefficient α is selected to 0.95 and 0.8 in the rough annealing stage and focusing annealing stage, respectively. Based on the numerous experimental results, when γ is 0.3, it performs the best trade-off between area and power consumption.

The results are given in Table I, in which the first column shows the name of the circuit, the second column shows the number of states for the given circuit in the first column. Next three columns show comparison results among GA [8], MOGA [7] and the proposed method in terms of CPU time. The proposed method can converge quicker than results compared to GA method in [8]. MOGA minimises the logical expressions via communication with ESSPRESSO, resulting that it takes significant more time than the proposed method.

It can be seen in Table II that on average the proposed method produces results requiring 13.2% fewer product terms and 44.1% less switching activity compared to results obtained from NOVA [19]. Methods used in [20]

and [21] achieve good power reduction but paying penalty of area consumption. MOGA [7] and GA [8] perform well in both area and power. The proposed algorithm outperforms other methods in both area saving and less power consumption in the tested suite.

TABLE I.

RECENTLY PUBLISHED RESULTS IN TERMS OF CPU TIME

Circuits	States	GA[8] (s)	MOGA[7]	Ours (s)
bbara	10	0.17	0h and 8 mins	0.12
cse	16	0.59	3hs and 9 mins	1.1
donfile	24	0.77	6hs and 4 mins	1.58
keyb	19	1.61	3hs and 32 mins	0.87
modulo12	12	0.13	5hs and 56 mins	0.15
planet	48	2.75	25hs and 23 mins	2.77
s1	20	4.37	6hs and 0 min	3.94
styr	30	4.44	6hs and 5 mins	2.57
ex1	20	3.26	6hs and 7 mins	2.68
ex4	14	0.27	6hs and 1 mins	0.38
opus	10	0.21	0h and 40 mins	0.22
Total	606	18.57	69h and 5 mins	16.38

V. CONCLUSIONS

In the paper, two-stage simulated annealing based algorithm approach to the state assignment problem is proposed with the aim of minimising area and power dissipation for sequential circuits. By using designed two-stage annealing approach, the algorithm is able to find the best assignment with fast convergence, which has reduced switching activity to minimise the power dissipation and the area simultaneously. By combing the best and current solutions, it utilises the experience of past solutions, overcoming the problem that ranking system in GA selection takes significant runtime. As a result, the algorithm outperforms the existing GA-based FSM state-encoding strategies achieving 13.2% fewer product terms and 44.1% less switch compared to NOVA and is therefore more suitable for area/power-optimised realisation of FSMs.

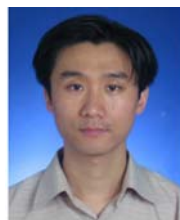
ACKNOWLEDGMENT

This work was supported by a grant (No. 11MS011) from State Key Lab of ASIC and System, China.

REFERENCES

- [1] P. J. Wang, H. Li, "Low power mapping for AND/XOR circuits and its application in searching the best mixed-polarity," *Journal of Semiconductors*, vol. 32, pp. 025007-6, 2011.
- [2] S. N. Pradhan, M. T. Kumar, S. Chattopadhyay, "Low power finite state machine synthesis using power-gating," *Integration, the VLSI Journal*, vol. 44, pp. 175-184, 2011.
- [3] P. J. Wang, J. G. Lu, X. Y. Zeng, "Searching the best polarity for low power based on WAGA," *Journal of CAD & Computer Graphics*, vol. 20, pp. 73-78, 2008.

- [4] T. Villa, T. Kam, R. Brayton, and A. Sangiovanni-Vincentelli, *Synthesis of FSMs: Logic Optimisation* Kluwer Academic Publishers, 1997.
- [5] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
- [6] A. E. A. Almaini, J. F. Miller, P. Thomson, and S. Billina, "State assignment of finite state machines using a genetic algorithm," *IEE Proc. Comput. Digit. Tech.*, vol. 142, pp. 279-286, 1995.
- [7] B. A. Al Jassani, N. Urquhart, A. E. A. Almaini, "State assignment for sequential circuits using multi-objective genetic algorithm," *IET Proc Comput. Digit. Tech.*, vol. 5, pp. 296-305, 2011.
- [8] Y. Xia, and A. E. A. Almaini, "Genetic algorithm based state assignment for power and area optimisation," *IET Proc Comput. Digit. Tech.*, vol. 149, pp. 128-133, 2002.
- [9] S. Chattopadhyay and P. Reddy, "Finite state machine state assignment targeting low power consumption," *IET Proc Comput. Digit. Tech.*, vol. 151, pp. 61-70, 2004.
- [10] S. Chattopadhyay, "Area conscious state assignment with flip flop and output polarity selection for finite state machine synthesis genetic algorithm approach," *Comput. J.*, vol. 48, pp. 443-450, 2005.
- [11] W. T. Shiue, "Novel state minimization and state assignment in finite state machine design for low power portable devices," *Integration. VLSI J.*, vol. 38, pp. 549-570, 2005.
- [12] S. Goren and F. J. Ferguson, "On state reduction of incompletely specified finite state machines," *Computer Electr. Eng.*, vol. 33, pp. 58-69, 2007.
- [13] W. M. Aly, "Solving the state assignment problem using stochastic search aided with simulated annealing," *America J. Eng. Appl. Sci.*, vol. 2, pp. 710-714, 2009.
- [14] T. A. Dolotta, and E.J. McCluskey, "The coding of internal states of sequential circuits," *IEEE Trans. Electron. Comput.*, vol. EC-13, pp. 549-562, 1964.
- [15] S. Chattopadhyay, and P. N. Reddy, "Finite state machine state assignment targeting low power consumption," *IET Proc Comput. Digit. Tech.*, vol. 151, pp. 61-70, 2003.
- [16] Y. Q. Sheng, A. Takahashi, S. Ueno, "2-Stage Simulated Annealing with Crossover Operator for 3D-Packing Volume Minimization," *Proceedings of the 17th Workshop on Synthesis And System Integration of Mixed Information Technologies*, pp. 227-232, 2012.
- [17] H. T. Wang; Z. J. Wang; J. Luo, "A simulated annealing algorithm for single container loading problem," *Proceedings of the 9th Intl. Conf. on Service Systems and Service Management*, pp. 551-556, 2012.
- [18] A. E. A. Almaini, N. Zhuang, and F. Bourset, "Minimisation of multioutput Reed-Muller binary decision diagrams using hybrid genetic algorithm," *IEE Electron. Lett.*, vol. 31, pp. 1722-1723, 1995.
- [19] T. Villa, and A. Sangiovanni-Vincentelli, "NOVA: state assignment of finite state machine for optimal two-level logic implementation," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 9, pp. 905-924, 1990.
- [20] S. K. Hong, I. C. Park, S. H. Hwang, and C. M. Kyung, "State assignment in finite state machines for minimal switching power consumption," *IEE Electron. Lett.*, vol. 30, pp. 627-629, 1994.
- [21] S. J. Wang, and M. D. Horng, "State assignment of finite state machines for low power applications," *IEE Electron. Lett.*, vol. 32, pp.2323-2324, 1996.



Meng Yang received Bachelor of Engineering (Honor) degree in Electrical Engineering from Shanghai University, Shanghai, China, in 1999. He received Master of Science with distinction in Electronics and Communication Engineering and Ph.D. in Electronics from School of Engineering Edinburgh Napier University, Edinburgh, UK, in 2002 and 2006, respectively.

Currently he is a lecturer of Department of Microelectronics, School of Information Science and Technology, Fudan University, Shanghai, China. His research interests include algorithms in FPGA design automation, logic synthesis, and dynamic reconfigurable FPGA automation design. He has published more than 30 research papers.

Dr. Yang is a member of IET and Chairman of Young Member Section of IET Shanghai Branch.

TABLE II.
EXPERIMENTAL RESULTS SHOWING POWER AND AREA IN COMPARISON

Circuits	NOVA[19]		MOGA[7]		GA[8]		Hong[20]		Wang[21]		This paper	
	area	Power	area	power	area	Power	Area	power	area	power	area	Power
bbara	24	0.495	22	0.49	22	0.317	26	0.295	26	0.279	22	0.277
bbsse	30	1.5	28	0.663	27	0.783	31	0.856	31	0.776	27	0.758
cse	46	0.604	43	0.39	43	0.355	50	0.292	48	0.239	43	0.250
donfile	28	1.75	22	1.375	36	1.6	40	1.083	45	1.125	22	1.458
keyb	48	1.466	46	0.98	46	0.674	52	0.647	58	0.556	46	0.533
modulo12	12	1	10	0.75	12	0.583	12	0.583	12	0.5	10	0.568
planet	87	2.831	81	2.49	86	2.424	101	1.153	103	0.984	81	1.680
s1	80	1.698	43	1.37	66	1.48	85	1.131	91	1.175	43	1.188
sand	97	1.085	94	0.585	89	0.765	110	0.604	109	0.61	92	0.666
styr	94	1.278	78	1.1	88	0.943	101	0.578	99	0.553	78	0.578
ex1	44	1.358	48	0.78	52	0.842	49	0.755	47	1.135	48	0.754
ex4	19	1.316	13	0.568	14	0.421	16	0.495	18	0.957	13	0.476
opus	16	0.812	15	0.49	15	0.556	17	0.524	17	0.712	15	0.471
train11	9	0.619	10	0.414	10	0.339	9	0.36	10	0.714	10	0.302
Total	634	17.812	553	12.445	606	12.082	699	9.356	714	10.315	550	9.959
Improved%	0%	0%	-13%	-30%	-4%	-32%	10%	-47%	13%	-42%	-13.2%	-44.1%