Proc. Natl. Sci. Counc. ROC(A)
Vol. 20, No. 6, 1996. pp. 641-650

Low-Power Greedy State Pair Grouping Algorithm and Power Reduction Estimation for Finite State Machine

CHUA-CHIN WANG AND YANG-LIN FAN

Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan, R.O.C.

(Received February 17, 1996; Accepted August 2, 1996)

ABSTRACT

The power consumption of a finite state machine (FSM) can be reduced if the count of bit change activity during the state transition is decreased by appropriate arrangement of the states. In this work, we propose a greedy pair grouping of states for FSM and analyze the expected performance of this algorithm. The probability that a state will appear is computed by using the state transition probability matrix and the initial input signal. A greedy pairwise grouping algorithm is then presented in which the states are pairwisely grouped together by the magnitude of their individual terminal probability. The bit representation of the state assignment is carried out in the reversed order of the sequence of the grouping. The systematic characteristics of the greedy grouping results in transition reduction compared to the classical method. Furthermore, an upper bound of the expected reduction ratio of our method is discovered. Results using this greedy pairwise state grouping method show that the performance of the reduction of transition activity can be pre-estimated before circuits are implemented in hardware.

Key Words: state assignment, greedy pair grouping, power estimation, transition density

I. Introduction

Recently, there has been a surge of interest in low-power devices and design techniques mainly due to the boom in personal wireless communication tools. While many papers have been published describing power-saving skills for use in digital systems (Chandrakasan *et al.*, 1992; Chen *et al.*, 1994; Horowitz *et al.*, 1994), a common measure for power consumption has been widely employed, i.e., **transitions density** (Najm, 1993). This is due to the average dynamic power dissipation by a CMOS gate:

$$P_{\text{ave}} = \frac{1}{2}\sum_{\forall \text{gates}} V_{\text{dd}}^2 C_{\text{load}} f_{\text{switch}},$$

where C_{load} is the output capacitance, and f_{switch} is the switching frequency or the average number of gate transitions per unit of time. Hence, we certainly can reduce the power consumption by designing a digital circuit that lowers the transition activity but still achieves the identical logic function (Chandrakasan *et al.*, 1992; Ercegovac and Lang, 1994; Horowitz *et al.*, 1994). This transition reduction is independent of the power supply and processing technologies (Olson and Kang, 1994). We attempt to reduce the state transition activity for finite state machines (FSM) by optimizing state assignment. This optimization is based on the *transition probability* on each edge between one state and another state.

However, most prior assignment algorithms are quite complicated and converge slowly (Hachtel *et al.*, 1994; Murgai *et al.*, 1994; Olson and Kang, 1994). We propose a greedy state pair grouping algorithm to achieve reduction of transition activity (Wang and Fan, 1995). After computing the terminal probability of each state, the states are re-ordered according to their individual terminal probabilities. The basic idea is that in each grouping, scanning the state with the largest probability will enable us to find one of its next possible states with the largest probability to combine with, except in a few special situations, e.g., when the grouping causes a "cut point" for the rest of the states in the state diagram.

Although the simulation results of the greedy state grouping reveal satisfactory performance, the expected best reduction ratio still remains unsolved. Regarding this problem, we also analyze the reduction ratio of our method in contrast to the classical or random assignment method. An interesting result is that we can predict the average reduction ratio without actual realization of the FSM.

In this paper, we will first briefly introduce definitions of the terms used in our work and the derivation of the terminal probability of each state. Then, the greedy pair grouping of states for an FSM will be discussed. Some simulation results will be presented to verify the performance of our approach.

II. Greedy Pairwise Grouping Algorithm (GPGA)

The FSM can be represented by a transition matrix. Assume there are n states in an FSM of which the states are $G=\{S_0, S_1, ..., S_{n-1}\}$. The probability that state S_i will go to state S_j is $P_{ij}(x)$, where x is the input to the FSM, and

$$P_{ij}(x)=\begin{cases} 1, & \text{if } x=\text{don't care} \\ P_{ij}(0), & \text{if } x=0 \\ 1-P_{ij}(0), & \text{if } x=1 \end{cases}$$

Note that the default value of $P_{ij}(0)$ and $P_{ij}(1)=1-P_{ij}(0)$ is 0.5. Hence, the transition matrix for Ω can be written as

$$A=\begin{bmatrix} P_{00} & P_{10} & \cdots & P_{(n-1)0} \\ P_{01} & P_{11} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ P_{0(n-1)} & P_{1(n-1)} & \cdots & P_{(n-1)(n-1)} \end{bmatrix}$$

$\lim_{k\to\infty}A^k$ exits as long as any eigenvalue of A is not -1. When the initial state vector, P_{init}, is given, we can easily derive the **terminal probability** for each state:

$$P_{\text{stable}}=(\lim_{k\to\infty}A^k)\cdot P_{\text{init}}, \tag{1}$$

where each component of P_{stable} is the terminal probability of a state S_i, $\forall i=0, ..., (n-1)$.

Hence, the **transition probability** for each transition edge (te) from state i to state j is defined as

$$P_{\text{te}}(S_i, S_j)=P_{\text{stable}}(S_i)\bullet P_{ij}(x)+P_{\text{stable}}(S_j)\bullet P_{ji}(y), \tag{2}$$

where $x, y\in\{0, 1, \text{don't care}\}$, $i\neq j$.

1. Greedy Pairwise Grouping by Terminal Probability

(1) Find the edge with the largest transition probability, say $P_{\text{te}}(S_i, S_j)$, $i\neq j$. Group S_i, S_j into a new state, α_0. The terminal probability of this new state is $P_{\text{stable}}(\alpha_0)=P_{\text{stable}}(S_i)+P_{\text{stable}}(S_j)$.

(2) Temporarily exclude those states which are grouped already. Repeat Steps (1) and (2) until all states are traversed once. If the number of states is odd, then the last one is grouped by itself.

(3) Thus, the original state diagram, G, is transformed into a new state graph, $G'=\{\alpha_0, \alpha_1, ..., \alpha_{n'-1}\}$. If $n'=2$, go to the bit assignment procedure. If $n'=\lceil\frac{n}{2}\rceil$, the grouping at this stage is successful. Go to Step (1) to start a new stage of state grouping. Otherwise, there is at least one **cut point** in the graph G'. Restart at Step (1) by choosing the edge with the second largest probability.

(4) Compute the transition probability for each edge in the new graph, which will be utilized for the next stage of grouping. According to the definition shown in Fig. 1, the probability of the transition edge of β_0 and β_1, $P_{\text{te}}(\beta_0, \beta_1)$, is computed as

$$P_{\text{te}}(\beta_0, \beta_1)=\sum_{\alpha_i\in\beta_0}\sum_{\alpha_j\in\beta_1}P_{\text{stable}}(\alpha_i)\bullet P_{\text{te}}(\alpha_i, \alpha_j)$$

$$+\sum_{\alpha_j\in\beta_1}\sum_{\alpha_i\in\beta_0}P_{\text{stable}}(\alpha_j)\bullet P_{\text{te}}(\alpha_j, \alpha_i), \tag{3}$$

where β_i's are the nodes after the grouping while α_i's are the current stage nodes.

Note that the necessary condition for an optimal bit assignment is that $n'\leq\lceil\frac{n}{2}\rceil$. The cut point is defined as a state node which is an articulation point of the graph. The cut point basically divides the states into at least two sets. If one state in one set wishes to travel to another state in another set, it has to go through this cut point. When the grouping causes the occurrence of any cut point, this is very disadvantageous to the

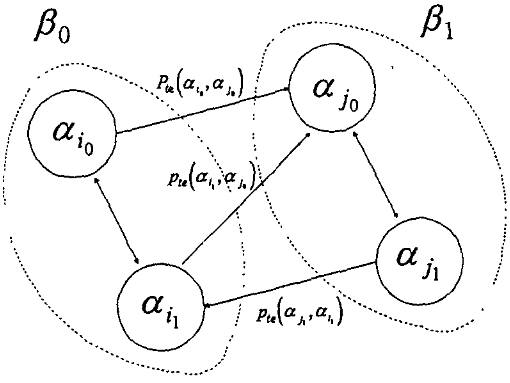
β_0
β_1
α_{i_0}
α_{j_0}
α_{i_1}
α_{j_1}
$P_{te}(\alpha_{i_0}, \alpha_{j_0})$
$p_{te}(\alpha_{i_1}, \alpha_{j_0})$
$p_{te}(\alpha_{j_1}, \alpha_{i_1})$

Fig. 1. The transition edge probability of β_0 and β_1, $P_{\text{te}}(\beta_0, \beta_1)$.

optimal state assignment.

2. Bit Assignment

By using the previous procedure, a 2-node graph will finally be generated. Clearly, the greedy pairwise grouping forms a full binary tree. That is, a node in the current graph contains one or two nodes from the previous graph. We then use the following bit assignment at each stage.

(1) Start from the last 2-node graph, and backtrack to the original state diagram. Assume the last 2-node graph consists of two node, Y and Z. If $P_{\text{stable}}(Y) \geq P_{\text{stable}}(Z)$, then $S_Y=0$ and $S_Z=1$, and vice versa.

(2) Each node at the current stage contains one or two nodes from the previous stage. That is, the parent node has either one child or two children. If there is only one child, add one "0" to the bit representation. If there are two children, add "0" to the bit representation of the node with larger terminal probability, and add "1" to the other node.

(3) Repeat Step (2) until the original states are all assigned.

3. Estimation of the Reduction Ratio

$code(S_i)=b_{m-1}b_{m-2}\ldots b_0$ indicates the bit representation of state S_i, where b_j is either 0 or 1. $TBG(S_i, S_j)$ is for the number of changed bits per clock cycle of the greedy grouping method, $TBN(S_i, S_j)$ is for the classical method, and $TBW(S_i, S_j)$ is for the worst case of the classical method. In addition to the above definitions, the probability that any state bit will be changed is assumed to be p. The default value of p is 0.5.

greedy pair grouping: Referring to Fig. 2 which shows the encoding result of GPGA for any FSM with $n=3$, the leaf nodes of the binary tree contain the codes for the states. Note that the X in the tree indicates the don't care. Since 0 and 1 are dual in Boolean algebra, we can assume robustly that the state with the largest terminal probability has the code 00...0. The optimal result occurs when $P_{\text{stable}}(S_i) \geq P_{\text{stable}}(S_j)$, $i<j$. That is, there exists no cut point in the graph of the FSM. Then, the optimal result of GPGA must be as follows:

$$code(S_0)=0=(000)_2$$

$$code(S_1)=1=(001)_2$$

$$\vdots$$

$$code(S_7)=7=(111)_2.$$

classical method: However, the conventional method for state assignment of an FSM results in a different binary tree, as shown in Fig. 3, with $n=3$. Thus, the worst case of the state assignment should be generated following a guideline; i.e., the codes of two states which reside at the ends of the edge with largest transition probability are complementary. The probability of the transition from S_i to S_j is computed by

$$P'_{\text{te}}(S_i, S_j) = P_{\text{stable}}(S_i) \times P_{ij}(I), \tag{4}$$

where I is the binary input vector. For example, $n=3$, and $P_{\text{te}}(S_0, S_1)$ is the largest. Then, $code(S_0)=0=000$ while $code(S_1)=7=111$.

4. Expectation of Transition Activity

GPGA: Assume that the number of bits for each state

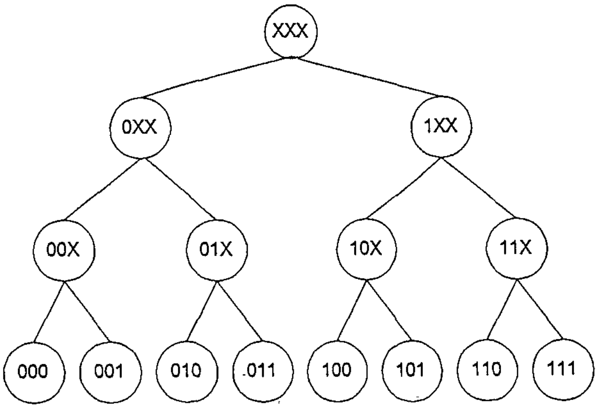

Fig. 2. The encoding result of GPGA for any FSM with $n=3$.

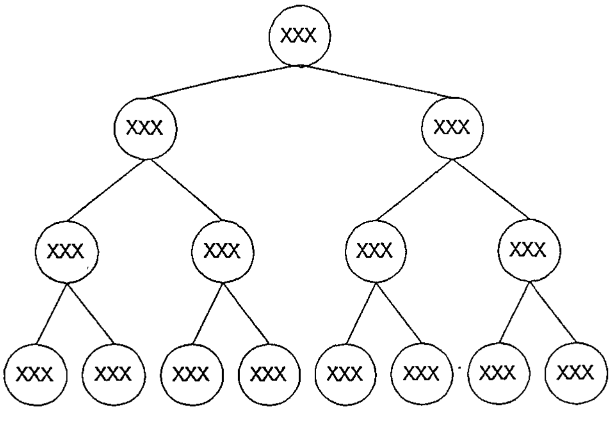

Fig. 3. The possible assignment of the classical method for any FSM with $n=3$.

is n while the number of states is m. The expectation for the number of transitions for a state S_i can be described by the following equation:

$$E_{S_i} = \sum_{j \neq i} P'_{\text{te}}(S_i, S_j) \cdot TBG(S_i, S_j), \quad 0 \le j \le m-1. \tag{5}$$

Thus, the expectation value of the overall transition activity is a summation of the above expectations:

$$E_{\text{greedy}} = \sum_{i=0}^{m-1} E_{S_i} = \sum_{i=0}^{m-1} \sum_{j \neq i} P'_{\text{te}}(S_i, S_j) \cdot TBG(S_i, S_j), \quad 0 \le j \le m-1. \tag{6}$$

Hence, given any FSM to which is assigned states according the GPGA, the expectation value of the transitions can be predicted based on Eq. (6).

Classical approach: In the traditional classical approach for state assignment, the number of transitions for state S_i to state S_j can be 0, 1, ..., or n. Hence, it is easy to derive that the expectation value for the classical approach is

$$TBN(S_i, S_j) = \sum_{k=0}^{n} k \cdot C^n_k p^k (1-p)^{n-k} = np. \tag{7}$$

Thus, the expectation value of the overall transitions is derived as

$$E_{\text{classical}} = \sum_{i=0}^{m-1} \sum_{j \neq i} P'_{\text{te}}(S_i, S_j) \cdot TBG(S_i, S_j), \quad 0 \le j \le m-1. \tag{8}$$

As for the worst case assignment of the classical method, the adjacent states at leaf nodes in Fig. 3 have to be mutually complementary. That is, $code(S_i) = \overline{code(S_{i+1})}$, i=0, 2, 4, ... Then, $TBW(S_i, S_{i+1}) = n$ while $TBW(S_i, S_j)$, $j \neq i+1$ can be assumed to be less than or equal to $n-1$. Finally, we conclude that the expectation value of the worst case assignment of the classical method is

$$E_{\text{worst}} = \sum_{i=0}^{m-1} \sum_{j \neq i} P'_{\text{te}}(S_i, S_j) \cdot TBW(S_i, S_j), \quad 0 \le j \le m-1. \tag{9}$$

Upper bounds of reduction ratio: Based upon the above analysis, several upper bounds for transition reduction for the greedy pair grouping algorithm can be found.

(1) Average reduction ratio:

$$R_{\text{ave}} = \frac{E_{\text{classical}} - E_{\text{greedy}}}{E_{\text{classical}}} \cdot 100\%.$$

(2) Upper bound of reduction ratio:

$$R_{\text{upper}} = \frac{E_{\text{worst}} - E_{\text{greedy}}}{E_{\text{worst}}} \cdot 100\%.$$

(3) Absolute upper bound:

$$R_{\text{abs}} = \frac{n-1}{n} \cdot 100\%.$$

Note that R_{abs} occurs in an ideal situation. That is, every two states in an FSM are only one bit away from each other.

III. Simulation and Analysis

Example 1: In order to verify the reduction of transition activities which is possible using our greedy pairwise state grouping algorithm, we will utilize an FSM to illustrate how the algorithm works and what the result is. Referring to Fig. 4, which shows a 9-state FSM, we derived its terminal probability for each state by the given initial vector, [1 0 0 0 0 0 0 0 0], and the transition matrix formed from the state diagram. The terminal probability for each state is

P_A=0.0361 P_B=0.0181 P_C=0.3795

P_D=0.0241 P_E=0.0241 P_F=0.2892

P_G=0.0120 P_H=0.1440 P_I=0.0723.

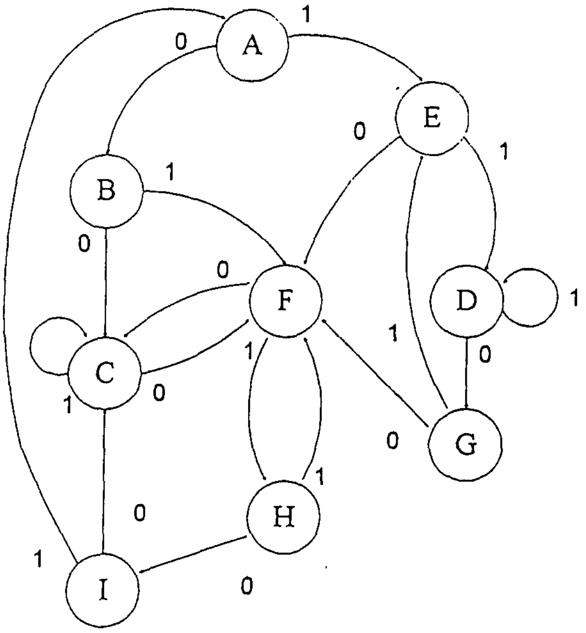

Fig. 4. 9-state FSM.

Since $P_C > P_F > P_H > P_I > P_A > P_D = P_E > P_B > P_G$, we can then start to group the states pairwisely. At the first stage, we can get the 5-node graph shown in Fig. 5. Then, we can further group the nodes into a 3-node graph as shown in Fig. 6. Next, we will get a 2-node graph as shown in Fig. 7. Now, we backtrack through the grouping stages and process the bit assignment for each state using our algorithm to get the final state assignment of the FSM:

$$\gamma_0=\begin{cases}\beta_0=\begin{cases}\alpha_0=\begin{cases}C:0000\\F:0001\end{cases}\\ \alpha_1=\begin{cases}H:0010\\I\ :0011\end{cases}\end{cases}\\ \beta_1=\begin{cases}\alpha_2=\begin{cases}A:0100\\E:0101\end{cases}\\ \alpha_3=\begin{cases}D:0110\\G:0111\end{cases}\end{cases}\end{cases}$$

$\gamma_1=\beta_2=\alpha_4=B$: 1000.

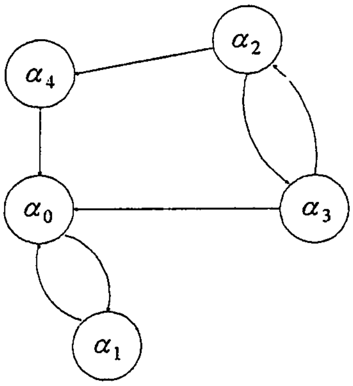

Fig. 5. Grouping 9 nodes into 5 nodes.

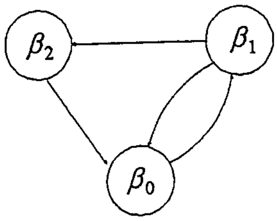

Fig. 6. Grouping 5 nodes into 3 nodes.

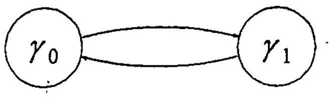

Fig. 7. Grouping 3 nodes into 2 nodes.

Note that the bit length for each state is the same as that of the classical logic design method. We can use CADENCE and HSPICE to simulate these two logic circuits for the same finite state machine. Figure 8 shows the logic circuit of our method, and Fig. 9 is the logic circuit of the classical method. Thus, in Fig. 10, given a series of input signal, we find that the total number of transitions of the greedy pairwise state grouping method is 27. Using the same input sequence, the total number of transitions of the classical method is 44, as shown in Fig. 11. A comparison of the power

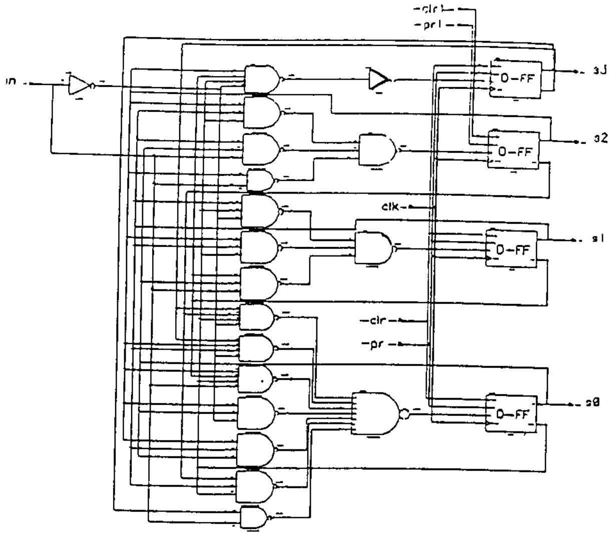

Fig. 8. The logic circuit of the greedy method.

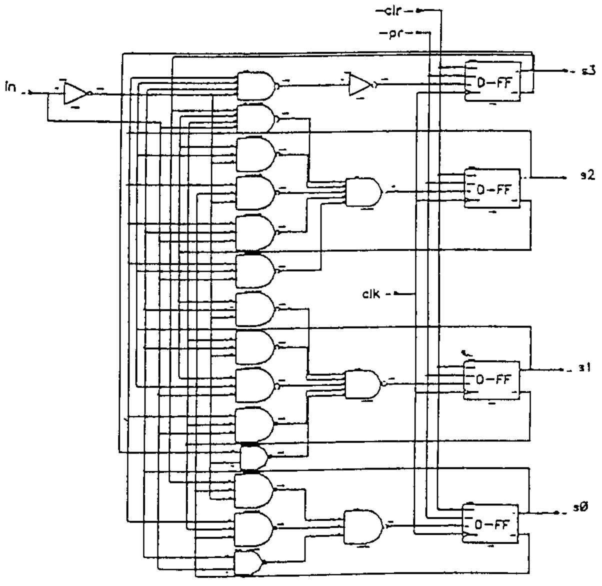

Fig. 9. The logic circuit of the classical method.

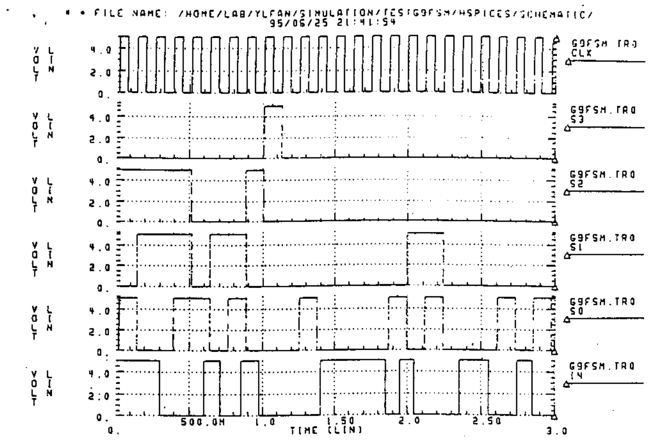

Fig. 10. The simulation results obtained by the greedy method.

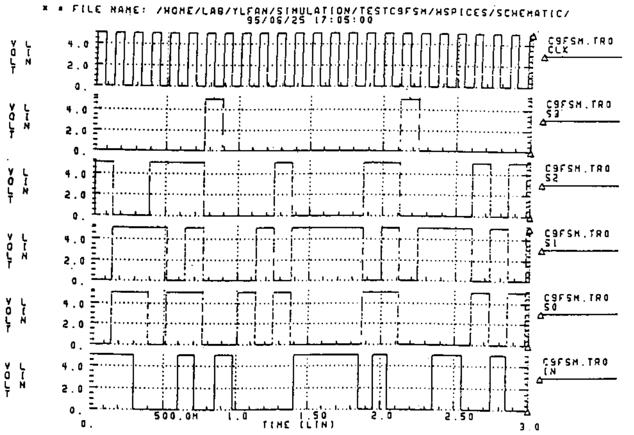

Fig. 11. The simulation results obtained by the classical method.

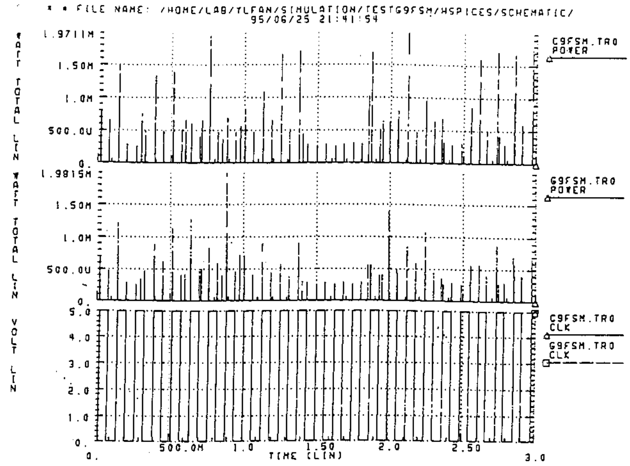

Fig. 12. A comparison of the power dissipation between the greedy method and classical method.

consumption between these two circuits is shown in Fig. 12. Hence, the reduction ratio of transition activity is 38.6%. As for total power consumption, we have conducted a series of simulations to illustrate the impact to the total power saving. Referring to Table 1, the average power reduction is about 17% while the reduction of transition activity is on average 33%. An interesting phenomena is that the higher the clock frequency is, the greater the power savings are.

Example 2: In order to verify our analysis of the transition reduction obtained by GPGA, we have conducted another series of simulations. Referring to Fig. 13, which is another 7-state FSM with only one input signal, the final state assignment is as follows:

$$\begin{aligned}
&\text{code}(S_G)=000\\
&\text{code}(S_C)=001\\
&\text{code}(S_B)=010\\
&\text{code}(S_E)=011\\
&\text{code}(S_F)=100\\
&\text{code}(S_D)=101\\
&\text{code}(S_A)=110.
\end{aligned}$$

The theoretical results based on the conclusion of our analysis are $E_{\text{greedy}}=1.199900$ and $E_{\text{classical}}=1.499850$. The reduction ratio is 20%. The upper

Table 1. Simulation Results Obtained by CADENCE and HSPICE

I/P Sequence		Classical Method		Greedy Method		Reduction Ratio	
input data	clock frequency (Hz)	power dissipation (nW)	# of transitions	power dissipation (nW)	# of transitions	power dissipation (%)	transition activity (%)
Seq-1	8.13	95.699	44	84.196	27	11.60	38.60
Seq-2	8.13	92.424	83	88.794	51	8.90	38.55
Seq-3	8.26	88.961	42	68.140	36	23.40	14.29
Seq-4	10.30	107.730	109	88.794	65	17.00	40.36
Seq-5	10.30	104.020	91	81.504	63	22.50	30.70

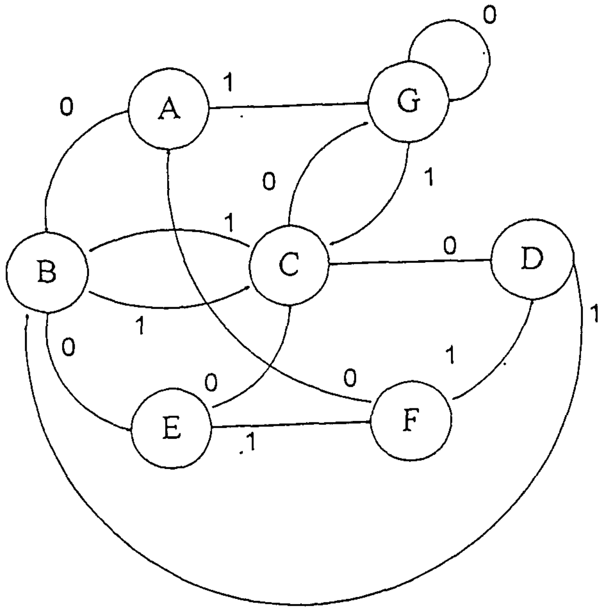

Fig. 13. 7-state FSM with only one input signal.

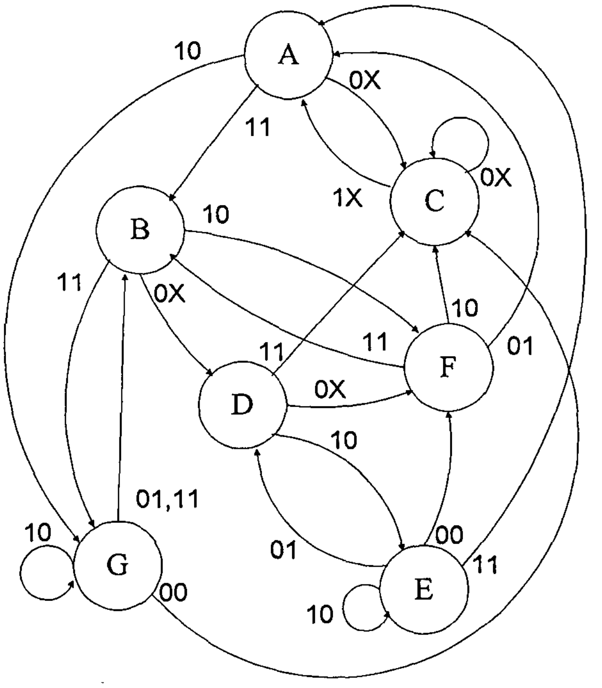

Fig. 14. 7-state FSM with 2-input signals.

bound of the reduction ratio is 25.71% given E_{worst}=1.866500. Five different test input signal sequences were utilized to test the performance. The results are shown in Table 2, in which the average transition is 1.20310, which is close to E_{greedy}= 1.199900.

For FSM with multiple input signals, we have built a 7-state 2-input FSM for illustration purposes, as shown in Fig. 14. The final state assignmnet by GPGA is

code(S_C)=000
code(S_A)=001
code(S_B)=010
code(S_G)=011
code(S_F)=100
code(S_D)=101
code(S_E)=110.

The theoretical results based on the conclusion of our analysis are E_{greedy}=1.117950 and $E_{\mathrm{classical}}$ =1.500000. The reduction ratio is 21.37%. The upper bound of the reduction ratio is 35.43% given E_{worst}=1.826800. Another five different test input signal sequences were also used to test the performance. The results are shown in Table 3, in which the average transition is 1.17544, which is close to E_{greedy}=1.11790.

Example 3: In order to verify the performance of GPGA, we have encoded the states of several MCNC benchmark FSMs. The results are listed in Table 4. The average reduction ratio of power relative to the classical method is 12.60%. For each entry in the last column of Table 4, we conducted a series of simulations in which five different input vector sequences were fed into the benchmark circuit to evaluate the performance.

Table 2. The Simulation Results for the Expectation Value for a 7-Satte FSM with 1-Input Signal

1-bit input data	steps	the number of transitions				Expectation value of the number of transitions
		b2	b1	b0	Total	(E_{greedy})
seq-1	196	23	66	148	237	1.2092
seq-2	196	23	75	136	234	1.1939
seq-3	196	25	66	143	234	1.1939
seq-4	196	23	65	143	231	1.1785
seq-5	196	24	73	146	243	1.2398
average	–	–	–	–	–	1.2031

Table 3. The Simulation Results for the Expectation Value for a 7-State FSM with 2-Input Signal

2-bit input data	steps	the number of transitions				Expectation value of the number of transitions
		b2	b1	b0	Total	(E_{greedy})
seq-1	196	39	56	138	233	1.1887
seq-2	196	39	55	130	224	1.1429
seq-3	196	40	57	130	227	1.1582
seq-4	196	41	63	132	236	1.2040
seq-5	196	35	58	139	232	1.1836
average	–	–	–	–	–	1.1754

Table 4. Benchmark Simulations

Benchmark	Classical Method			Greedy Method			Reduction Ratio	
	# of transistors	average # of transitions	average power (uW)	# of transistors	average # of transitions	average power (uW)	transitions (%)	power (%)
bbtas.kiss2	200	36.00	49.83	208	29.40	38.95	18.33	21.83
dk17.kiss2	262	48.80	84.19	268	33.80	49.17	30.74	41.56
dk27.kiss2	196	81.00	61.16	174	55.80	49.28	31.11	19.42
dk512.kiss2	394	54.00	72.86	334	46.20	68.04	14.40	6.62
s27.kiss2	304	30.00	43.62	266	29.00	41.47	3.33	4.92
s386.kiss2	464	44.00	75.55	480	42.20	79.73	4.09	-5.53
donfile.kiss2	308	39.00	92.45	348	36.40	93.21	6.67	-0.82

Table 5. A Benchmark Circuit with Different Input Sequences

Benchmark : dk512.kiss2
States: 15 Input: 1 D-ff: 4

	Classical Method (394)		Greedy Method (334)		Reduction Ratio	
*.spice	# of transitions	power(uW)	# of transitions	power (uW)	transitions (%)	power (%)
dk512*v10	22+12+11+14=59	87.98	15+7+5+22=49	76.48	16.95	13.07
dk512*v11	16+14+15+8=53	66.30	15+14+1+16=46	75.22	13.21	-11.86
dk512*v12	15+14+14+9=52	73.25	14+14+1+14=43	65.07	17.31	11.17
dk512*v13	16+16+16+5=53	59.22	16+16+1+16=49	68.95	7.55	-16.43
dk512*v14	16+14+15+8=53	77.57	14+14+1+15=44	54.48	16.98	29.43
average	54.00	72.86	46.20	68.04	14.40	6.62

( ): the number of transistors

Table 6. Simulation Results of Estimated Transition Reduction

Expection Value	classical Average		Worst Case		Greedy Average		Average Reduction Ratio (%)		
	Est.	Sim.	Est.	Sim.	Est.	Sim.	Est.	Sim.	Error
bbtas.kiss2	1.2872	1.4400	1.5000	–	1.0601	1.1760	17.6800	18.3300	3.55
dk17.kiss2	1.5840	1.9200	1.6971	–	1.2988	1.3520	18.0100	30.7400	41.41
dk27.kiss2	1.4964	3.2400	1.5209	–	1.0698	2.2320	28.5100	31.1100	8.36
dk512.kiss2	1.9998	2.1600	2.1508	–	1.7111	1.8480	14.4339	14.4000	0.24
s27.kiss2	1.3087	1.2000	1.5104	–	1.2777	1.1600	2.3700	4.0900	42.05

The result for *dk*512 is shown in Table 5, which gives an average reduction ratio of 6.62% by HSPICE. In addition, Table 6 shows the prediction of our estimation analysis by simulation of the MCNC benchmark FSM circuits. The average error between the estimated transition reduction and the predicted transition reduc-

Table 7. The Results of Comparison between GPGA and Reencoded

Benchmark	Re-encoded		GPGA	
	transitions (%)	power (%)	transitions (%)	power (%)
s27.kiss2	1.11	-47.96	3.33	4.92
s386.kiss2	2.9	-2.94	4.09	-5.53
donfile.kiss2	5.12	-11.6	6.67	-0.82

tion is 19.12%, which is slightly better than the 20.94% of Najm's method (Najm, 1993). The result is very appealing and shows the correctness of our prediction of the reduction ratio.

Finally, compared with the famous re-encoded method for FSM (Hachtel *et al.*, 1994), our proposed GPGA algorithm still performs better even in the three most difficult benchmark circuits, *s*27, *s*386, and *donfile*. As shown in Table 7, the proposed GPGA provides on average -0.48% power reduction while the re-encoded method provides on average -20.83% power reduction.

IV. Conclusion

According to the simulation results, our approach indeed can drastically reduce the transition activity and, consequently, the total power for an FSM. In contrast with prior state assignment methods, our approach is much more systematic. We have also presented a quick method to identify whether there is any cut point in the grouping process which might cause non-optimal bit representation of state assignment. We have also proposed a method to precisely predict the performance of GPGA even before the actual circuit is built for a given FSM. The simulation results indicates the correctness of our analysis.

Acknowledgment

This research was partially supported by National Science Council under grant NSC 85-2215-E-110-008.

References

Chandrakasan, A. P., S. Sheng, and R. W. Broderson (1992) Low-power CMOS digital design. *IEEE J. Solid-State Circuits*, **27**(4), 473-484.

Chen, Z. J. Scott, J. Burr, and J. D. Plummer (1994) CMOS technology scaling for low voltage low power applications. *1994 IEEE Symp. on Low Power Electronics*, pp. 56-57. San Diego, CA, U.S.A.

Micheli, G., R. K. Brayton, and A. Sangiovanni-Vincentelli (1985) Optimal state assignment for finite state machines, *IEEE Trans. Computer-Aided Design*, **CAD-4**(3), 269-285.

Ercegovac, M. D. and T. Lang (1994) Reducing transition counts in arithmetic circuits. *1994 IEEE Symp. on Low Power Electronics*, pp. 64-65. San Diego, CA, U.S.A.

Horowitz, M., T. Indermaur, and R. Gonzalez (1994) Low-power digital design. *1994 IEEE Symp. on Low Power Electronics*, pp. 8-11. San Diego, CA, U.S.A.

Hachtel, G. D., M. H. De La Rica, A. Pardo, M. Poncino, and F. Somenzi (1994) Re-encoding sequential circuits to reduce power dissipation. *Proc. of 1994 Inter. Workshop on Low Power Design*, pp. 69-74. Napa, CA, U.S.A.

Lin, J. Y., T. C. Liu, and W. Z. Shen (1994) A cell-based power estimation in CMOS combinational circuits. *The 5th VLSI Design/CAD Symposium*, pp. 315-320. Tau-Yuan, Taiwan, R.O.C.

Murgai, R., R. K. Brayton, and A. Sangiovanni-Vincentelli (1994) Decomposition of logic functions for minimum transition activity. *Proc. of 1994 Inter. Workshop on Low Power Design*, pp. 33-38. Napa, CA, U.S.A.

Najm F. N. (1993) Transition density: a new measure of activity in digital circuits. *IEEE Trans. Computer-Aided Design*, **12**(2), 310-323.

Olson, E. and S. M. Kang (1994) Low-power state assignment for finite state machine. *Proc. of 1994 Inter. Workshop on Low Power Design*, pp. 63-68. Napa, CA, U.S.A.

Wang, C. C. and Y. L. Fan (1995) Low-power state assignment by greedy state pair grouping. *The 6th VLSI Design/CAD Symposium*, pp. 298-301. Pu-Li, Taiwan, R.O.C.

有限狀態機以低功率貪婪式狀態配對成群演算法與功率減少估算分析

王朝欽　范揚麟

中山大學電機工程研究所

摘　要

如果將狀態指定(state assignment)做適當的安排使得在狀態遞移時狀態位元改變減少，則有限狀態機(FSM)的功率(power)消耗會減少。在這篇論文中，我們提出一個有限狀態機的貪婪式狀態配對成群(Greedy State Pair Grouping)演算法，並且分析此演算法所期望的效率。以狀態遞移機率矩陣和初始輸入訊號來計算狀態的發生機率。貪婪式配對成群演算法則是將各個狀態的終端機率(terminal probability)大小來配對成群在一起，而狀態位元表示的指定則以狀態配對成群的相反順序處理。從系統化的特性而言，貪婪式配對成群對於切換次數減少的期望值預測是相對於傳統的方法而言。因此，在我們的方法裡發現了期望切換次數減少的上界(upper bound)。使用貪婪式配對成群法的結果證明狀態遞移時，狀態位元切換次數減少的比例可以在電路以硬體實現之前被預估出來。