設 計 自 動 化 67－4
（1993．6．25）

SBDDを用いた最小STT状態割当

権 容珍　矢島 脩三

京都大学工学部情報工学教室

非同期式順序回路に対する状態割当の一つである、Single Transition Time(STT)状態割当において、命題論理を用いて最小解を得る新手法を提案する。この手法では二分割に対する状態変数による被覆条件を、新しく導入されたブール変数を用いた論理式で表現し、データの内部表現として共有二分決定図を使用することにより大きな問題を扱うことができる。また、ここで提案された手法は、最近注目を浴びている制約付き状態割当にも応用可能で、今までの制約付き状態割当のアルゴリズムはheuristicな手法によるものが大部分を占めていることとは異なり、最小解が得られる。実験結果も示す。

Minimum STT State Assignment Using SBDD

KWON, Yong-Jin and YAJIMA, Shuzo

Dept. of Information Science, Faculty of Engineering
Kyoto University, Kyoto-shi, 606-01, Japan
Email: kwon@yajima.kuis.kyoto-u.ac.jp

We propose a new method of the single transition-time(STT) assignments for asynchronous sequential circuits, in which the propositional calculus, or Boolean algebra is adopted. Exact minimum solutions of the STT assignments are obtained by our method. In order to handle huge propositional formulas, the shared binary decision diagrams(SBDD's) are used as an internal representation of the formulas which denote the STT assignment for a given normal flow table. Moreover, as an application of the minimum algorithm, a minimum algorithm for the constrained encoding problems are also proposed, for which so far only heuristic algorithms are known for solving large and practical problems. However, our method always guarantees minimum solutions to be obtained. Experimental results show that our methods are effective to obtain minimum solutions at significantly reduced computation cost.
key words: *unicode STT, state assignment, minimum, asynchronous, sequential circuits, propositional calculus, dichotomies, SBDD's*

1 Introduction

While every state assignment, which has a unique binary vector for every internal state of a synchronous sequential circuit, may yield a correct circuit realization, it is not a sufficient condition for asynchronous sequential circuit realizations. The state assignment for asynchronous sequential circuits must ensure that no critical races exist among the state variables. We are concerned with the state assignment for asynchronous sequential circuits in this paper.

As state assignments of asynchronous sequential circuits, many methods have been proposed, including a class of the single transition-time(STT) assignments. In order to obtain minimum solutions of the STT assignment problems, we have to conquest an NP-complete component of the problems, which is the covering problem, so that it is said to be difficult to calculate a minimum STT assignment for problems of large and practical size. In this paper, we propose a minimum algorithm for the STT assignments, especially the *unicode* ones. Propositional calculus is employed for expressing the unicode STT assignments with introduced Boolean variables here, and the shared binary decision diagrams(SBDD's)[1] are used as an internal representation of propositional formulas denoting the state assignments, so that minimum solutions are obtained efficiently for large and practical problems.

Besides, as an application of the conception that for obtaining the minimum unicode STT assignments, dichotomies of a given flow table are denoted by propositional formulas by means of newly introduced Boolean variables, we also propose a minimum algorithm for the constrained encoding problems[2] whose importance has grown with the recent advances in the synthesis of sequential machines, where [2] proposed a heuristic algorithm with no guarantee of its results being minimum.

We have implemented the minimum unicode STT assignment and the minimum constrained encoding, and conducted experiments to evaluate the performance. The results show that our methods are very effective to obtain minimum solutions at the significantly reduced computation time and memory. The STT assignments are said to be very practical so that our newly proposed method is useful at the design of asynchronous sequential circuits.

The rest of the paper is organized as follows. Section 2 covers the unicode STT assignments. The minimum unicode STT assignment is described in section 3. In section 4, the minimum constrained encoding is proposed, and we present in section 5 the results of our experiments. Finally, we have concluding remarks and point to directions for future work.

2 Unicode STT assignments

We consider a class of single transition-time(STT) assignments[3]. State assignments for which a single transition time is always sufficient for any transition are called STT assignments in which transitions may occur by means of noncritical races among all of the state variables distinguishing the internal and final states. Speed of operation is often an important consideration in the design of switching circuits, and it is not unusual to pay for it with significant increases in the number of components used. A key factor influencing the speed of operation of asynchronous sequential circuits is the maximum number of

transition times required for an interstate transition. Only normal flow tables are treated here, and the assignment are restricted to those with one binary vector composed of state variables per state of machines, which refer to as unicode STT assignments.

The fundamental conception of the STT assignments is the one of dichotomies. Letting U and V be disjoint subsets of states of flow tables, we define a *dichotomy* as the unordered pair $(U; V)$. Given a pair of transitions $i \rightarrow j$ and $k \rightarrow m$, where i and j are each different from k and m, we say that the dichotomy is $(U; V)$, where $U = \{i, j\}$ and $V = \{k, m\}$. This dichotomy is generally written as $(ij; km)$. In cases in which one of the transitions is degenerate (say $i = j$), we obtain a dichotomy such as $(i; km)$, and if both transitions are degenerate, the dichotomy reduces to $(i; k)$. A state variable y_i in a particular state assignment is said to *cover* a dichotomy $(U; V)$ if $y_i = 0$ for every state in U and $y_i = 1$ for every state of V (or vice versa). Using the above notion, we can now state a basic theorem for the STT assignments as follows:

> **THEOREM**[3] *A state assignment for a normal flow table is a valid unicode STT assignment iff, for every pair of transitions $i \rightarrow j$ and $k \rightarrow m$ that appear in the same column and such that $j \neq m$, the dichotomy $(ij; km)$ is covered by at least one state variable of the state assignment.* □

The procedure finding minimal unicode STT assignments for a normal flow table is simply summarized as follows.

1. For each column, form a dichotomy $(ij; km)$ for every pair of transitions $i \rightarrow j$ and $k \rightarrow m$ where $j \neq m$.

2. Find maximal compatibles corresponding to the dichotomies.

3. Find a minimal set of the maximal compatibles that covers every dichotomy.

The procedure contains some components so that it is said to be difficult to obtain minimum solutions for large and practical problems. The components are that the number of maximal compatibles may be exponential in the number of states, and the covering problem is NP-complete. Thus efficient solving methods have been expected and have been proposed so far. Here, we propose a minimum and effective algorithm for the unicode STT assignments.

3 Minimum unicode STT assignment using SBDD

We propose a minimum unicode STT assignment algorithm in this chapter. we represent the unicode STT assignment as propositional formulas which are specified with SBDD's as internal representations. Minimum solutions are obtained by searching the SBDD's in time linear in the number of Boolean variables appeared in the SBDD's. We call this method a minimum unicode STT assignment using SBDD.

3.1 Introduction of Boolean variables

We introduce some Boolean variables which are used for denoting the unicode STT assignment problem with the propositional calculus.

We will let:

$$S = \text{a set of } n \text{ states}$$
$$B = \text{a set of binary vectors } (b_0 \cdots b_i \cdots b_{k-1}), b_i \in \{0,1\}$$

When a codeword $b(b_0 \cdots b_i \cdots b_{k-1}) \in B)$ is assigned to a state $X \in S$, we introduce a Boolean variable $D^X_{b_i} \in \{0,1\}$ in the following:

$$D^X_{b_i} = \begin{cases} 1 & \text{if the codeword } b_0 \cdots b_i \cdots b_{k-1} \text{ is assigned to the state } X \in S \\ & \text{and } b_i = 1 \\ 0 & \text{otherwise.} \end{cases}$$

For example, if a codeword $101(b_0b_1b_2)$ is assigned to a state s_i, then b_0 is 1, b_1 is 0 and b_2 is 1 so that $D^{s_i}_{b_0}$ is 1, $D^{s_i}_{b_0}$ is 0 and $D^{s_i}_{b_2}$ is 1. Using these Boolean variables, we can denote that a dichotomy is covered by a state variable. Namely, if we have a propositional formula $\mathcal{Q}$ given as follows, then $\mathcal{Q} = 1$ denotes that " A dichotomy $(ij; km)$ is covered by a state variable b_0".

$$\mathcal{Q} \equiv D^i_{b_0} D^j_{b_0} \overline{D^k_{b_0}}\,\overline{D^m_{b_0}} + \overline{D^i_{b_0}}\,\overline{D^j_{b_0}} D^k_{b_0} D^m_{b_0}.$$

3.2 Propositional representations of the unicode STT assignment

We represent a method called propositional representation of the unicode STT assignments, which denotes the assignment with propositional formulas based on the Boolean variables introduced above.

In order that the unicode STT assignment of a given flow table is expressed by the propositional calculus, we have to calculate necessary dichotomies for the assignment of the given flow table. For a normal flow table we can derive a set of dichotomies with ease. we show an example in the following. we assume that a normal flow table is given and three state variables y_0, y_1 and y_2 are used for the assignment of the flow table, where each variable is binary. For building propositional formulas denoting the unicode STT assignment of the flow table, Suppose that we obtain the following dichotomies from the flow table.

$$(S_0S_1; S_4S_5), (S_4S_5; S_3), (S_1S_2; S_3S_4), (S_2S_4; S_1S_3), (S_2S_4; S_3S_5)$$

Then, a proposition that " The dichotomy $(S_0S_1; S_4S_5)$ is covered by at least one state variable among y_0, y_1 and y_2" is denoted by $\mathcal{D}_{(01;45)} = 1$, where the propositional formula $\mathcal{D}_{(01;45)}$ is given as follows:

$$\mathcal{D}_{(01;45)} \equiv D^{S_0}_{b_0} D^{S_1}_{b_0} \overline{D^{S_4}_{b_0}}\,\overline{D^{S_5}_{b_0}} + \overline{D^{S_0}_{b_0}}\,\overline{D^{S_1}_{b_0}} D^{S_4}_{b_0} D^{S_5}_{b_0} + D^{S_0}_{b_1} D^{S_1}_{b_1} \overline{D^{S_4}_{b_1}}\,\overline{D^{S_5}_{b_1}}$$
$$+ \overline{D^{S_0}_{b_1}}\,\overline{D^{S_1}_{b_1}} D^{S_4}_{b_1} D^{S_5}_{b_1} + D^{S_0}_{b_2} D^{S_1}_{b_2} \overline{D^{S_4}_{b_2}}\,\overline{D^{S_5}_{b_2}} + \overline{D^{S_0}_{b_2}}\,\overline{D^{S_1}_{b_2}} D^{S_4}_{b_2} D^{S_5}_{b_2}$$

Similarly, for the dichotomy $(S_4S_5; S_3)$ we have the following formulas $\mathcal{D}_{(45;3)}$:

$$\mathcal{D}_{(45;3)} \equiv D^{S_4}_{b_0} D^{S_5}_{b_0} \overline{D^{S_3}_{b_0}} + \overline{D^{S_4}_{b_0}}\,\overline{D^{S_5}_{b_0}} D^{S_3}_{b_0} + D^{S_4}_{b_1} D^{S_5}_{b_1} \overline{D^{S_3}_{b_1}}$$
$$+ \overline{D^{S_4}_{b_1}}\,\overline{D^{S_5}_{b_1}} D^{S_3}_{b_1} + D^{S_4}_{b_2} D^{S_5}_{b_2} \overline{D^{S_3}_{b_2}} + \overline{D^{S_4}_{b_2}}\,\overline{D^{S_5}_{b_2}} D^{S_3}_{b_2}$$

For all the dichotomies $(U;V)$, we make propositional formulas $\mathcal{D}_{(U;V)}$ like above, and let $\mathcal{D}$ be the product of all the formulas $\mathcal{D}_{(U;V)}$:

$$\mathcal{D} \equiv \prod_{all(U;V)} \mathcal{D}_{(U;V)}$$

Then, the unicode STT state assignment is denoted by $\mathcal{D}$. A minimum solution having the minimal number of state variables is obtained when the $\mathcal{D}$ that is not contradiction is acquired for the first time, while the number of state variables is increased one by one each step, and the minimum solution is associated with a product term which is involved in the $\mathcal{D}$ when the $\mathcal{D} = 1$.

Good representation of propositional formulas is a key to efficient implementation. In our implementation we use shared binary decision diagrams(SBDD's)[1] as an internal representation of the $\mathcal{D}$ in order to handle huge propositional formulas. The SBDD's, improved binary decision diagrams, are graph representations of Boolean functions, and have the following desirable properties;

- Many functions are represented compactly and simultaneously by sharing isomorphic sub-graphs.
- Logic operations between functions can be carried out much efficiently.

These two advantages are very much suited to our purpose. Since in our method, we need to represent many propositional formulas to express a set of dichotomies, the former property is very favorable. The latter one is also favorable for making an SBDD denoting the STT assignments. The propositional formulas appearing in our method are produced by logical operations. We can expect efficient implementation because SBDD's are known to have good affinity for the operations.

When the $\mathcal{D}$ is represented by an SBDD, A minimum solution of the unicode STT assignment for the flow table is associated with a path which is involved in the SBDD when the path goes to node "1"(the "1" leaf). Thus there are a lot of minimum solutions in the SBDD. Namely, by Searching out all the paths going to the "1" leaf, we can have all minimum ones. A minimum solution, or path going to the leaf is searched out in time linear in the number of Boolean variables appeared in the SBDD. The number of Boolean variables B is $B = n \times k$ if using the Boolean variables introduced in this paper, where n is the number of states and k the number of state variables allowed to the assignments. Using coded Boolean variables, it can be expected that the figure is decreased. See (4) for details.

The algorithm is summarized as follows.

Algorithm[STTBDD]
(Inputs)
A directed graph G consists of n vertices.
k state variables allowed to the unicode STT assignment.
(Output)
If a solution exists, a minimum solution.
(Algorithm)

step1 $\mathcal{D} := TRUE$;

step2 while(all dichotomies)
begin
Make a next propositional formula $D_{(U;V)}$;
$\mathcal{D} := \mathcal{D} \cdot D_{(U;V)}$;
If $\mathcal{D}$ is a contradiction, then FAIL ;
end

step 3 Search a minimum solution and output it ; □

4 An application: Constrained Encoding Problems

We propose a minimum algorithm for constrained encoding problems as an application of the minimum unicode STT assignment described in the previous chapter, in which dichotomies are expressed by propositional formulas using introduced Boolean variables.

Given a set $S = \{s_0, s_1, \cdots, s_{n-1}\}$ of n states, *dichotomy-based constrained encoding*[2] aims at finding an assignment m of S into a set $\{m(s_0), m(s_1), \cdots, m(s_{n-1})\}$ of n binary k-tuples (bit vectors), in such a way that k is minimized and a set of dichotomy constraints are satisfied. A dichotomy constraint DC on S is a pair $DC = (S^+; S^-)$ of disjoint subsets of S; to satisfy such a constraint the encoding must have at least one state variable that has the value 1 for all the states in S^+ and 0 for all the states in S^-, or vice versa.

The importance of minimum constrained encoding has grown with the recent advances in the synthesis of synchronous sequential machines. It is now accepted as a general rule that, in order to find an economic logic realization, it is desirable to assign certain groups of states to groups of neighboring vertices in the hyper cube. In particular, if a programmable logic array(PLA) is used to implement the combinational part of the circuit, such groups of states can be identified by a technique called symbolic minimization. Each group is to be embedded into a face in the hyper cube; such constraints are therefore called *face-embedding* constraints. They can also be reformulated as dichotomy constraints[2].

For the constrained encoding problem, [2] proposed a heuristic algorithm. However, the algorithm basically consists of the greedy step so that, in general, minimum solutions are not guaranteed. We propose here a minimum algorithm of the constrained encoding problem , by which for practical and large problems, minimum solutions are always calculated. We define the minimum constrained encoding problem as follows: *We assume that a set S of n states, and a set DC of d nontrivial constraints on S are given. The minimum constrained encoding problem is to find an encoding m of S so that all constraints of DC are covered by k state variables, where k is a minimum integer.*

For obtaining minimum constrained encoding, as making propositional formulas for calculated dichotomies in chapter 3, it is sufficient to build them for all given constraints. Then by representing the propositional formulas denoting all given constraints with SBDD's, we may obtain minimum solutions at significantly reduced computation time and memory cost. Other steps of the algorithm for the constrained encoding problems are similar with them of that for the unicode STT assignments.

5 Experimental results

We implemented the minimum one-shot state assignment algorithm on UNIX operating system in C language on the basis of the method described in the preceding sections. This implementation was linked with an SBDD manipulator[1]. We conducted experiments to evaluate its performance on SUN SPARC station 2 workstation (64MByte). We assigned several normal flow tables by giving them a number "k" which is the number of state variables allowed to the state assignment. The results are shown in Table 1. The first column shows the names of the flow tables; for example, S04D05 specifies a normal flow table which has 4 states and 5 dichotomies. The column **#m sv** shows the minimal number of state variables.

Table 1:Experimental results.

name	#m sv	#M nodes	CPU[sec]
S04D05	3	43	2.5
S05D05	3	65	2.6
S05D12	4	522	2.7
S06D16	5	2922	3.2
S07D20	4	3678	3.7

The column **#M nodes** contains the number of nodes of SBDD's used for representing the flow tables (maximum values). The last column shows CPU times for initialization of SBDD's and the state assignment per flow table in seconds. The number of nodes of SBDD's was limited to $1,000,000(2^{20})$, and the implementation (including the SBDD manipulator) used approximately 20MByte storage at most. It is clear from Table 1 that the minimum unicode STT assignment algorithm is very efficient in a point of view of computational time.

6 Conclusion and Future work

We have proposed a new method of the unicode STT assignments which leads to exactly minimum solutions. In the proposed method, dichotomies of normal flow tables are denoted by propositional formulas which are written with Boolean variables introduced here. By the use of an SBDD as an internal representation of the propositional formulas, we can reduce the storage requirement required for representing the formulas. With experiments, the new method may proves efficient. The STT assignments are said to be very practical so that our newly proposed method is useful at the design of asynchronous sequential circuits. In parallel, we showed an application of the proposed method to the minimum constrained encoding problems.

Our future work includes:

1. More sophisticated algorithm which reduces the size of nodes of SBDD's representing the minimum unicode STT assignment.

2. New method which introduces Boolean variables used for propositional formulas should be found out in order to handle more lager sequential circuits.

Besides, when we have an SBDD which is not contradiction and denotes the minimum unicode STT assignment for a flow table, there are a lot of minimum solutions in the SBDD. Though our method now output any one among them as a minimum solution, it is worth considering for example that a minimum solution which is associated with what we call the simplest state transition functions, is searched out. This is another our future work.

Acknowledgment

We would like to express our appreciation to Mr. Minato(now with NTT) for using of the SBDD manipulator. we would also like to thank the members of Yajima Laboratory at Kyoto University for their valuable discussions and comments.

References

(1) S. Minato, N. Ishiura and S. Yajima, "Shared Binary Decision Diagram with Attributed Edges for Efficient Boolean Function Manipulation", *Proc. 27th Design Automat. Conf.*, pp. 52-57,1990.
(2) C. J. Shi and J. A. Brzozowski, "Efficient Constrained Encoding for VLSI Sequential Logic Synthesis," *EUROPEAN DESIGN AUTOMATION CONFERENCE*, Germany, 1992.
(3) S. H. Unger, "Asynchronous Sequential Switching Circuits", John Wiley & Sons, Inc., 1969.
(4) Kwon, Yong-Jin and Yajima, Shuzo., "Minimal One-Shot State Assignment Using Binary Desicion Diagrams", *IEICE Techcal Report, COMP92-27.*, 1992.
(5) R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Trans. Comput.*, Vol.C-35,8, pp.677-691,1986.
(6) Kwon, Yong-Jin, "One-Shot State Assignment for Asynchronous Sequential Machines", Master Thesis, KYOTO University, 1990.