A Subnetwork-based Method for Optimization of Large Scale Circuits with Transduction Method

2H−4

‡Takenori KOUDA †Youxin WANG ‡Yahiko KAMBAYASHI
†Faculty of Engineering,Kyoto University ‡Graduate School of Engineering,Kyoto University

1 Introduction

With the progress of VLSI technology and high competition between manufactures, requirements for rapid design of correct and efficient circuits are growing greater and greater. Automatic Design in this area has been researched for years.

In order to verify and maintain a given circuit efficiently, keeping the size of the circuit small is essensial. Moreover, lessening the area of the circuit may spare chip area; lessening the level of the circuit may shorten the propogation time.

As a logic optimization method, Transduction Method[1] was proposed in 1970's. Many application showed that it was powerful optimizaiton tool. Especially after the introduction of BDD technology, Transduction Method can treat relatively large circuits with relatively low cost.

However, like any other Boolean minimization mehtods, Transduction Method consume lots of memories. This restricts its usability, especially for Large Scale Circuits.

Many subnetwork-based approaches have been proposed. But optimization result was greatly affected by the quality of the subnetwork. In this paper, we propose an algorithm of extraction for subnetwork suited for Transduction Method and experiment results show that this improvement expands its usability for large scale circuits largely.

2 Transduction Method

[Definition 1] If none of the output functions of the network is changed after replacing the function realized at gate v_j or connection c_{ij} by function f, then the function f is called a *permissible function* for gate v_j or connection c_{ij}, respectively.

For any gate or connection, we can define a maxinum set of its permissible function, we call it *Maximum Set of Permissible Functions*(MSPF). To a certain group of gates or connections, *Compatible Sets of Permissible Functions*(CSPF) ensures that concurrently change of these functions will not make non-permitted outputs.

Normally, Transduction Method contains these steps: i) Calculate functions of every gate and connection; ii) Calculate CSPF of every gate and connection; iii) Transform the circuit.

Transduction Method was once very costly. The use of SBDD[2] effectively reduced the necessary memories. However, BDD can only manupulate 0 and 1, but not *don't care*. So two BDDs are used to express a *don't care* function. When the number of CSPFs which contain *don't care* increase the use of the SBDD nodes also increase. When the consumption of SBDD nodes–also the memories–reachs some extent, the computation performance drops greatly. This situation makes it diffecult to apply Transduction Method for large scale circuits. How to optimize large scale circuits under certain SBDD nodes remains a problem.

3 Optimizing Large Scale Circuits

Since CSPFs cost lots of SBDD nodes, omitting the calculation of CSPFs of some gates and connections may reduce some computation cost. But for many circuits, this also reduces the freedom of optimization and the extension of application of Transduction Method is limited.

It is obvious that it is more practical to apply Transduction Method to only some parts of the original circuit rather than to the whole.

Generally speaking, optimization results are better when the subnetwork is bigger, since Transduction Mehod is a don't-care-based optimization method. A bigger network means more *don't cares*, genarally. On the other hand, when the subnetwork grows bigger, especially when SBDD nodes are over some limit, the computing performance drops rapidly.

In some research[3], a threshold is set for SBDD nodes beforehand. Extraction is carried out from primary inputs. So extraction and optimization can only be taken arround primary inputs. Random extractions are desired.

4 Extracting Sub-Network

In the field of logic partition, some algorithms has been developed to partition a circuit. Of course, partition is desired if it can be easily reached. Unfortunately, these algorithms are always time-consuming.

Fundamentally, what we need is to extract one "good" subnetwork rather than to divide the circuit skillfully. "Good" subnetwork means its inputs and outputs are relatively few, since we know too many inputs and outputs will lessen the freedom of optimization.

Optimization process is as follows: i) Extract a subnetwork from the circuit; ii) Optimize that subnetwork; iii) Return the optimized subnetwork to the circuit; iv) Go to i) if necessary.

部分回路抽出によるトランスダクション法の大規模回路への応用
‡幸田武範 †王又新 ‡上林弥彦
†京都大学工学部 ‡京都大学大学院工学研究科

Table 1: Experiment Result

	Inital			Result			
Circuits	♯Gates	♯Conn	♯Level	♯Gates	♯Conn	♯Level	Time(sec)
C888	394	740	22	328	721	22	352
C1355	619	1137	27	493	1026	23	409
C1708	718	1335	37	444	980	27	455
alu4	720	1441	42	581	1505	36	722
too_large	748	1743	24	584	1359	24	414
C2670	1017	1893	26	847	1628	25	951
C3540	1161	2426	41	1044	2426	38	815
frg2	1133	2613	15	854	2076	12	1013
dalu	1908	38899	38	1386	3086	37	1319
C5315	1929	4006	46	1665	3707	39	2443
t481	3393	8144	20	2559	5925	20	5357

Extracting algorithm is shown as follows:

[Extracting Algorithm:]

S:**set of gates in the subnetwork**

O:**other gates**

$E(g)$:**evaluation of gate** g

1. $S = \phi$
   take one gate randomly into S
2. **for** $\forall g \in O$ **and** g **connects to** S
   $S = S + g$
   until $\sharp S = N_1$
3. **compute** $E(g)$ **for** $\forall g \in O$
4. **loop**
   $g_M : E(g_M) \geq E(g)$ **for** $\forall g \in O$
   $S = S + g_M$
   recompute changed $E(g)$
   until $\sharp S = N_2$

Here, N_1,N_2 are parameters, N_1 is the size from which subnetwork grows, N_2 determines the size of the subnetwork. Evaluation functions evaluate the cost of a gate if subnetwork take it in. It contains input cost($E_I(g)$) and output cost($E_O(g)$). α and β are parameters.

$$E(g) = \alpha E_I(g) + \beta E_O(g) \quad (1)$$

$$E_I(g) = \sum_{p \in (IP(g) \cap O)} \frac{1}{\sharp(IS(p) \bigcap O)} \quad (2)$$

Here, O means set of gates not in subnetwork, IP refers to immediate predecessor, IS refers to immediate successor. Similarily,

$$E_O(g) = \sum_{s \in (IS(g) \cap O)} \frac{1}{\sharp(IP(p) \bigcap O)} \quad (3)$$

5 Experiments

We implemented the above algorithm in C++ language. The experiment was carried on on SUN-Ultra1(167MHz). Circuits are from LGSynth'91 multi-level benchmark circuits mapped with NOR. We used Transduction Method Release2.1 package. This package uses SBDD package from Yajima Lab,Kyoto University, which has a threshold for SBDD nodes, about 1,000,000. Under this restriction, normally we can optimize circuits under 400 gates with fast speed. N_1 and N_2 in extrancting algorithm are set to 70 and 250 respectively. Parameters α and β in Equa 1 are set both to 1. Optimization are carried on repeatedly until no progress is made between two consecutive repetition.

Experiment results are showed above in Table 1. As we can see from Table 1, the reduction of gate numbers are more than 10% in average. We optimized circuits with more than 1000 gates.

6 Conclusion

In this paper, we tried to optimize large scale circuits with a subnetwork-based Transduction Method. An extracting algorithm is proposed. And experiment result showed than with this algorithm we can treat circuits with more than 1000 gates with relatively low cost.

Theoretically, we can treat much greater circuits, but when size the circuits grows, the cost also grows. How to select subnetworks to optimize is what we have to research in the future.

Gererally, extracted subnetworks has more inputs and outputs than usual circuits, but still, Transduction Method optimized them satisfiably.

This reserch is supported in part by NEDO research grant.

References

[1] S.Muroga, Y.Kambayashi,H.C.Lai,J.N.Culliney: *Transduction Method-Design of Logic Networks Based on Permissible Function* IEEE Transactions on Computers, Vol.38,No.10, pp.1404-1424(1989).

[2] S.Minato,N.Ishiura,S.Yajima: *Shared Binary Decision Diagram with Attributed Edges for Efficient Boolean Function Manipulation*,Proc.27th Design Automat. Conf., pp.52-57,1990

[3] H.Nagai,Y.Kambayashi: *Extraction of Subcircuits from Large Networks for Transduction Method*, Proc. of the 51th Annual National Convertion of IPS Japan,7P-3,1995