Design of Logic Circuits with Wired-Logic Utilizing Transduction Method

Abstract

Wired-Logic is especially useful when designing fan-in restricted logic circuits which are implemented with bipolar and MOS transistors. However, there are few papers about logic design with Wired-Logic except the paper of the authors. In this paper, a method which reduces the levels of circuits by utilizing Wired-Logic is presented. The method restricts the number of fan-ins of a gate utilizing Wired-Logic. It efficiently utilizes Wired-Logic by transforming circuits using Transduction Method which is an optimization method. For the combination of NOR gates and Wired-OR gates which is especially important, an algorithm to utilize Wired-OR gates for the fan-in restriction is also presented. By performing experiments on 3-level NOR circuits, the proposed method is compared with *Generalized Serial Duplication* which is an efficient *Serial Duplication*. The levels of most circuits are reduced by utilizing Wired-OR gates, which shows the efficiency of Wired-Logic. The proposed method can be modified to use the cost function as the connection length which is more important than the levels of circuits.

key word : logic design, Transduction Method, Wired-Logic, fan-in restriction

1 Introduction

Recently, a large-scale circuit has been able to be implemented in a chip as the VLSI technology progresses. Therefore, automated logic design systems, such as generating initial circuits and optimizing circuits, have been vitally important and studied from many angles. Among such systems, Transduction Method has attracted a great deal of attention since it can optimize circuits powerfully[4][6][8].

Transduction Method[2][5] was developed by the authors at Illinois University during 1971 to 1973. It is a design method to reduce redundancies of a circuit and transform a circuit into another circuit based on the notion called *Permissible Functions*.

It is important to reduce the numbers of gates and levels under the maximum fan-in restriction when we design VLSI circuits. Serial Duplication[1], which inserts two NOR gates, is known to be a fan-in restriction method. The results of Serial Duplication can be improved by Transduction Method. The results of Generalized Serial Duplication[7][9] are much better than those of Serial Duplication. In this paper, a fan-in restriction method which uses Wired-Logic [3] and Generalized Serial Duplication together is presented.

Wired-Logic refers to the capability of tying the outputs of two or more gates together to perform additional logic without any extra components. It can be usually used when we design circuits with gates realized by such as TTL and ECL. Since Wired-Logic gates have much shorter delay than ordinary gates, it is considered that they are useful to reduce the numbers of gates and levels of circuits. Since Wired-Logic gates are not ordinary gates physically, some restrictions are imposed on their usage[3].

In this paper, the restrictions are relaxed by using the notion of *Connectable/Disconnectable* in Transduction Method. A transformation method which utilizes Wired-OR gates is also presented. We have done some experiments of proposed methods. The results show the effectiveness of using Wired-Logic at transforming circuits.

2 Preliminaries

In this section, we present the outlines of Transduction Method in [5], properties of Wired-Logic in [3], and Generalized Serial Duplication in [9].

2.1 Transduction Method

2.1.1 Basic Definitions

In this paper, we consider a loop-free combinational circuit. Although only NOR gates and Wired-OR gates are treated in this paper, the extension to other types of gates can be made easily. Let the number of inputs of a circuit be n, $IP(v)$ be the set of gates which are connected to an input of v and $IS(v)$ be the set of gates which are connected to the output of v.

A function realized at c (a gate or a connection) is denoted by $f(c)$. $f(c)$ is represented by a 2^n-dimensional vector, $(f(c)^{(1)}, f(c)^{(2)}, \cdots, f(c)^{(2^n)})$, where $f(c)^{(j)}$ is the values of $f(c)$ in the j-th row of the truth table for $f(c)$ ($f(c)^{(j)}$ is 1 or 0).

2.1.2 Permissible Functions

Definition 1 *If no specified components of the network output functions change after replacing the function realized at an input terminal, a gate, or a connection by a function f, then function f is called a permissible function for the input terminal, the gate or the connection, respectively.*

Usually there are more than one permissible functions for an input terminal, a gate or a connection. Therefore, we can consider a set of permissible functions for an input terminal, a gate or a connection. The sets of permissible functions which can be replaced simultaneously are called $CSPFs(CompatibleSetofPermissibleFunctions)$. In this paper, CSPFs are used for circuit transformation. The CSPF of c (a gate or a connection) is denoted by $G(c)$ below. $G(c)$ is represented by a 2^n-dimensional vector whose components are 0, 1, or $*$. The vector that has $*$ in its components denotes the set of all vectors that are formed by assigning 0 or 1 to $*$ in all possible combinations. For example, if the CSPF of c is {(1,0,0,0),(1,0,0,1),(1,0,1,0),(1,0,1,1)}, $G(c) = (1, 0, *, *)$. Please refer to [2] and [5] about how to calculate CSPFs.

2.1.3 Circuit Transformation by Permissible Functions

In addition to the removal of redundancies in a circuit, Transduction Method can change connections and replace gates with other gates by using permissible functions. Here, we mention about how to change connections, which is used in this paper.

Theorem 1 *Input terminal or gate v_i is said to be connectable to v_j, if the following conditions are satisfied[2].*

1. *$f'(v_j) \in G(v_j)$, where $f'(v_j)$ is the new function of v_j after adding the connection.*
2. *There is no path from v_j to v_i.*

Theorem 2 *Input terminal or gate v_i is said to be disconnectable to v_j, if the following condition is satisfied[2].*

$f'(v_j) \in G(v_j)$, *where* $f'(v_j)$ *is the new function of* v_j *after removing the connection.*

The method called C/DC (Connectable/Disconnectable)[2] optimizes a circuit as follows. First it increases the redundancies of the circuit by adding connectable connections by the Theorem 1. Then it removes connections by the Theorem 2 to changing the circuit configuration.

2.2 Properties of Wired-Logic

Wired-Logic refers to the capability of tying the outputs of two or more gates together to perform additional logic such as AND or OR. Wired-Logic is usually allowed in a circuit which contains such as NOR and NAND. It is especially useful in LSI. The number of fan-ins of a gate can be decreased by assembling the fan-ins with a Wired-OR gate if the gate is OR or NOR, with a Wired-AND gate if the gate is AND or NAND[3].

In this paper, tying the outputs of two or more gates is treated as a virtual gate. It is called a Wired-Logic gate. To treat a Wired-Logic gate as an ordinary gate at circuit transformation, some restrictions are imposed on their usage as follows.

Definition 2 *There are the following two restrictions to handle a Wired-Logic gate in the same manner as an ordinary gate.*

1. *Any gate which is connected to a Wired-Logic gate cannot be connected to other gates, other Wired-Logic gates or output terminals.*
2. *A Wired-Logic gate cannot be connected to any other Wired-Logic gates, input terminals or output terminals.*

For example, if gate v_4 is connected to gate v_1 and a Wired-Logic gate as shown in Fig. 1 (a), the circuit is equivalent to the circuit shown in Fig. 1 (b) because the Wired-Logic gate realized by connecting v_4 and v_5 is electronically connected to the input of v_1. If we handle a Wired-Logic gate in the same manner as an ordinary gate, the above confusion occurs. To avoid such confusion, we will use Fig. 1 (b) where all gates (v_4 and v_5 in the above example) which are connected to a Wired-Logic gate are not connected to any other gate. This is the first restriction on a Wired-Logic gate. Since a Wired-Logic gate realized by only connecting the outputs of gates, if a Wired-Logic gate is connected to other Wired-Logic gates, input terminals or output terminals, the logic functions of such gates may change. The second restriction avoids such cases.

The number of circuit levels is calculated as follows.

Definition 3 *Let the delay of an ordinary gate be* 1, *the delay of a Wired-Logic gate be* $d (d << 1)$, *and the number of levels of* v_i *from the outputs be* $L(v_i)$. $L(v_i)$ *is recursively defined as follows.*

1. *If* v_i *is an output terminal,* $L(v_i) = 0$.
2. *If* v_i *is a Wired-Logic gate,* $L(v_i) = \max_{v \in IS(v_i)}\{L(v)\} + d$.
3. *If* v_i *is an ordinary gate,* $L(v_i) = \max_{v \in IS(v_i)}\{L(v)\} + 1$.

Then the number of the circuit levels is given by $\max_{v \in V}\{L(v)\}$.

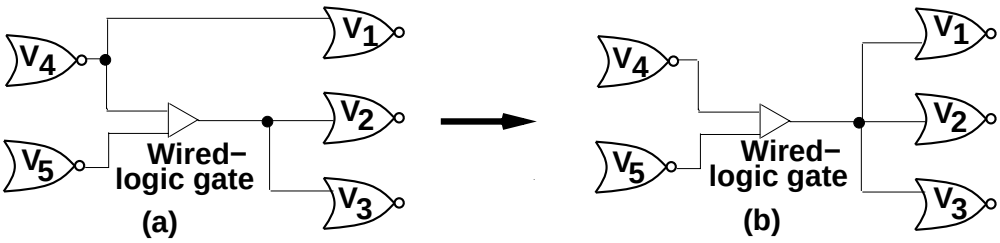

Figure 1: A Restriction on a Wired-Logic Gate.

Here, V denotes the set of all gates in the circuit. Practically the value of d is changed according to the summation of the numbers of inputs and outputs of a Wired-Logic gate. In this paper, we also restrict the number of fan-ins of a Wired-Logic gate and treat d as 0 for simplicity since $d << 1$.

The number of connections in a circuit which includes Wired-Logic gates is calculated by the following definition.

Definition 4 *A Wired-Logic gate with k_1 inputs and k_2 outputs can be realized by $(k_1 + k_2 - 1)$ connections[3].*

2.3 Generalized Serial Duplication

We can restrict the number of fan-ins of a NOR gate by Serial Duplication[1]. This method works as follows. For gate v, $IP(v)$ is divided into some groups at first. Then for each group, a pair of serially connected NOR gates are created and the outputs of all the gates in the group are connected to the former gate of the pair. Finally all the outputs of the pairs of serially connected NOR gates are connected to v. Generalized Serial Duplication[9], mentioned below, is more efficient than Serial Duplication.

If all the gates in a group which is obtained by the division of $IP(v_e)$ have the same input from v, the output of v_e does not change when v is connected to the latter gate of the pair of serially connected NOR gates for the group. In such a case, the output of v_e does not change even if v is removed from the fan-ins of all the gates which have one fan-out in the group. Generalized Serial Duplication makes good use of the above transformation not to increase the number of gates and the circuit levels when Serial Duplication is done.

For example, if all the gates in a group at Serial Duplication have the same input x from circuit A as shown in Fig. 2 (a), we can connect x to the latter gate of the pair of serially connected NOR gates for the group as shown in Fig. 2 (b) without changing the output of the circuit. Furthermore we can remove x from the fan-ins of the gates which have one fan-out in the group as shown in Fig. 2 (c) without changing the output.

3 Transformation with Wired-Logic

In this section, we propose the restrictions which are imposed on assembling the fan-ins of a gate with Wired-Logic gates. We also present methods to assembles the fan-ins with Wired-OR gates.

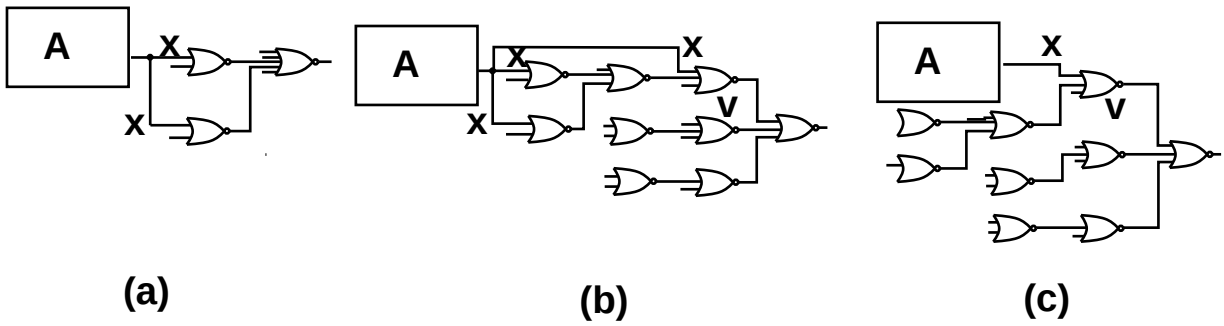

Figure 2: An Example of Generalized Serial Duplication.

3.1 Restrictions on Assembling with a Wired-Logic Gate

Some definitions are introduced here.

Definition 5 un-com-fanout
Un-com-fanout(v_i, v_j) denotes the set of gates that are included in $IS(v_i)$ and not included in $IS(v_j)$.

Definition 6 com-fan-out
com-fan-out(v_i, v_j) denotes the set of gates that are included in both $IS(v_i)$ and $IS(v_j)$.

The following theorem is obtained from Definition 2 and 5.

Theorem 3 *If the following condition is satisfied, the outputs of v_i and v_j can be assembled with a Wired-Logic gate, and they are said to be* **assemblable**.

> *All the gates in un-com-fanout(v_i, v_j) are either disconnectable to v_i or connectable to v_j and all the gates in un-com-fanout(v_j, v_i) are either disconnectable to v_j or connectable to v_i.*

For example, the outputs of v_i and v_j in the circuit shown in Fig. 3(a) cannot be assembled with a Wired-OR gate by Definition 2. However, since un-com-fanout(v_i, v_j) = $\{v_1\}$ and un-com-fanout(v_j, v_i) = $\{v_4\}$, we can assemble in the following cases by Theorem 3.

1. When v_j is connectable to v_1 and v_i is connectable to v_4.
2. When v_i is disconnectable to v_1 and v_j is disconnectable to v_4.
3. When v_j is connectable to v_1 and v_j is disconnectable to v_4.
4. When v_i is disconnectable to v_1 and v_i is connectable to v_4.

For example, the outputs of v_i and v_j can be assembled with a Wired-OR gate as Fig. 3(b) in case 1, as Fig. 3 (c) in case 2.

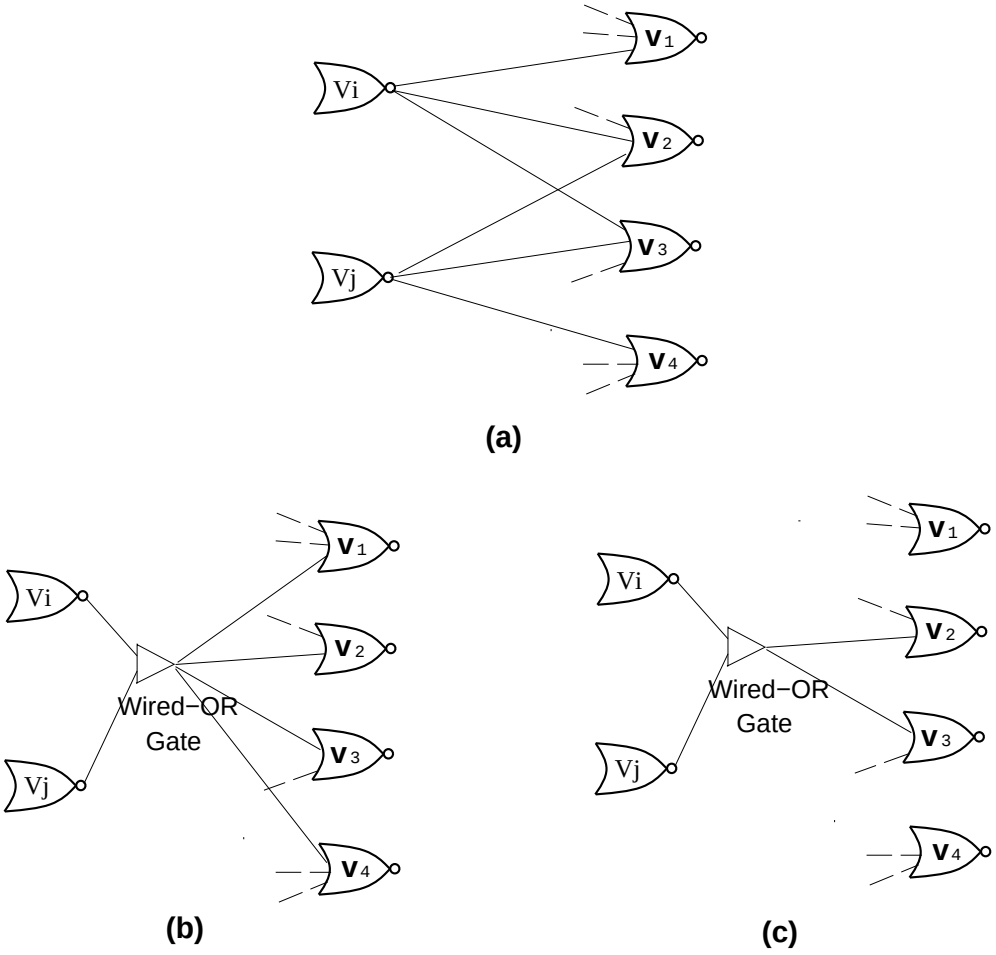

Figure 3: An Example of Assemblable.

3.2 Assembling Fan-ins with a Wired-OR Gate

Below we present a summary of a procedure "Assemble" that assembles the outputs of v_i and v_j with a Wired-OR gate. This procedure satisfies the conditions of Theorem 2. It is supposed that v_i and v_j are either NOR gates or Wired-OR gates and their outputs are assemblable. All the disconnectable connections in the outputs of v_i and v_j are supposed to be removed before this procedure.

Procedure Assemble

If both v_i and v_j are NOR gates

step 1 Add a new Wired-OR gate w in the circuit.

step 2 Connect the output of w to all the gates that are included in any one of com-fanout(v_i, v_j), un-com-fanout(v_i, v_j) and un-com-fanout(v_j, v_i).

step 3 Connect the outputs of v_i and v_j to w.

step 4 Disconnect all the outputs of v_i and v_j that are not connected to w, and halt.

If either v_i or v_j is a Wired-OR gate

step 1 From v_i and v_j, choose a Wired-OR gate as w and the other as v.

step 2 Connect the output of w to all the gates that are included in un-com-fanout(v, w).

step 3 Connect the output of v to w.

step 4 Disconnect all the outputs of v that are not connected to w, and halt.

If both v_i and v_j are Wired-OR gates

step 1 Connect the output of v_j to all the gates that are included in un-com-fanout(v_i, v_j).

step 2 Connect the outputs of all the gates that are connected to v_i and not connected to v_j, to v_j.

step 3 Remove v_i and halt.

If both v_i and v_j are NOR gates, the procedure creates a new Wired-OR gate which has two fan-ins. The number of fan-ins of this Wired-OR gate, however, is increased by assembling the outputs of the Wired-OR gate and any other gates. In such cases, it is necessary to restrict the total numbers of fan-ins and fan-outs of a Wired-OR gate because the delay of the Wired-OR gate becomes larger as the total numbers of its fan-ins and fan-outs increases.

The following is a summary of a procedure "Fanin-rest" that satisfies the fan-in restriction of gate v by using a Wired-OR gate as far as possible.

Procedure Fanin-rest

step 1 If the fan-in restriction of v is satisfied, halt. Otherwise, go to step 2.

step 2 Select a pair of assemblable connections from the inputs of v, assemble the pair of connections by Procedure Assemble and go to step 1. If there is no such pair, halt.

The results may change as the order of selecting a pair of assemblable connections at step 2 if there are many candidates. In this paper, the order is not considered.

3.3 Assembling Unassemblable Fan-ins with a Wired-OR Gate

There are some cases when the above procedure "Fanin-rest" cannot restrict the number of fan-ins of the gate whose fan-ins are not assemblable. Though the outputs of v_i and v_j are not assemblable, the outputs of v_i and v_j can be assembled by adding new gates and a Wired-OR gate if both v_i and v_j are not neither input terminals nor output terminals. Although extra gates must be added, this method can restrict the number of fan-ins without increasing the circuit levels.

We show an example as follows. In Fig. 4 (a), v_3 violates the fan-in restriction, but its fan-ins, the outputs of v_1 and v_2, are not assemblable. However, we can assemble the inputs of v_3 with a Wired-OR gate as shown in Fig. 4 (b) by adding v_1' which has the same inputs as v_1 and v_2' which has the same inputs as v_2. This transformation is very efficient when v_3 violates the fan-in restriction and the longest path in the circuit appears if the fan-in restriction of v_3 is done.

Below we present a summary of a procedure "Assemble-with-copy" that assembles the outputs of v_i and v_j by adding new gates even if they are not assemblable.

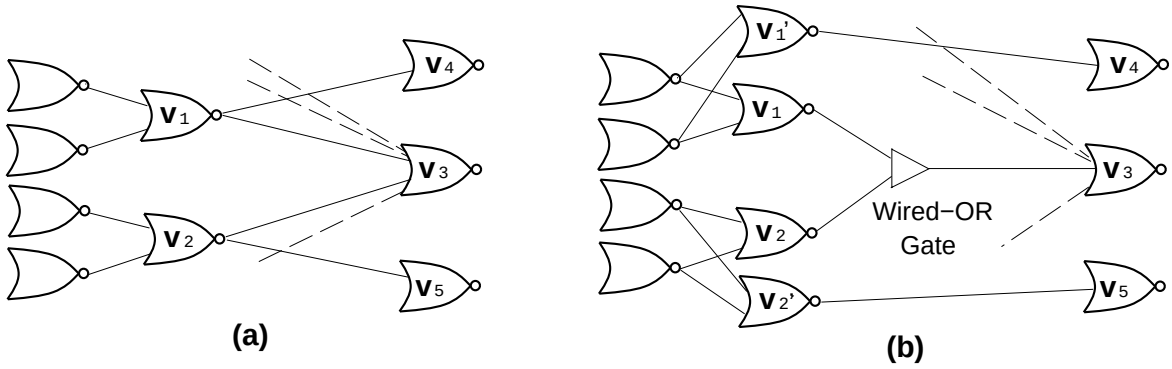

Figure 4: An Example of Assemble with Wired-OR in case of Unassemblable.

Procedure Assemble-with-copy

step 1 From un-com-fanout(v_i, v_j), select gates that are not disconnectable to v_i and not connectable to v_j. Let this set of gates be obstacle(v_i).

step 2 From un-com-fanout(v_j, v_i), select gates that are not disconnectable to v_j and not connectable to v_i. Let this set of gates be obstacle(v_j).

step 3 If obstacle(v_i) is not empty, add a new gate v_i' which has the same inputs as v_i and transform the circuit as follows.

- Connect the outputs of v_i' to all the gates in obstacle(v_i).
- Disconnect the outputs of v_i to all the gate in obstacle(v_i).

step 4 If obstacle(v_j) is not empty, add a new gate v_j' which has the same inputs as v_j and transform the circuit as follows.

- Connect the outputs of v_j' to all the gates in obstacle(v_j).
- Disconnect the outputs of v_j to all the gate in obstacle(v_j).

step 5 Apply Procedure Assemble to v_i and v_j.

In the above procedure, obstacle(v_i) denotes the set of gates that are connected to the output of v_i and obstruct to satisfy the condition of Theorem 3. obstacle(v_j) has the same meaning.

3.4 Transduction Method with Wired-OR Gates

Transduction Method optimizes a given initial circuit. Therefore, the initial circuit influences the optimization result and it is not unusual to obtain a local minimum solution. Fan-in restricted Transduction Method in consideration of the technology mapping reduces the capability of changing the circuit configuration, therefore it is expected to obtain a local minimum solution in more cases than the ordinary Transduction Method. A method to get out of such a local minimum solution was proposed in [10]. The method first performs Transduction Method

without the fan-in restriction in order to keep the capability of changing the circuit configuration, and then restricts the number of fan-ins by Generalized Serial Duplication. The results of this method are better than those of the ordinary Transduction Method in about 65% cases. An outline of the method is presented below as "Procedure Fan-in restricted Transduction Method".

Procedure Fan-in restricted Transduction Method

step 1 If this is not the first loop of this procedure and the cost of the circuit is not lower than the previous loop, halt. Otherwise go to step 2.

step 2 Apply C/DC procedure of Transduction Method to the circuit without the fan-in restriction.

step 3 Satisfy the fan-in restriction of gates which violate the fan-in restriction, and go to step 1.

In the method proposed in [10], Generalized Serial Duplication is done at the fan-in restriction in step 3. Therefore, the result becomes worse if a loss by Generalized Serial Decomposition is greater than a gain by Transduction Method without the fan-in restriction. In this paper, therefore, we propose a method that utilizes Wired-Logic at the fan-in restriction in the method of [10]. The experimental results of the proposed method is presented in Section 4.

4 Experimental Results

4.1 Various Transformation Methods

Using the methods mentioned in Section 3, we implemented the five circuit transformation methods which are mentioned bellow with C language and compared their experimental results. The SBDD package developed by Yajima Laboratory of Faculty of Engineering of Kyoto University was used for representing logic functions. The number of fan-ins of a NOR gate was restricted to 4.

Method 1 A method to perform only Generalized Serial Duplication.

Method 2 A method to perform Generalized Serial Duplication after applying Procedure Fanin-rest to the gates that do not satisfy the fan-in restriction. The number of fan-ins of a Wired-OR gate is also restricted to 4 and d (the delay of a Wired-logic gate) is treated as 0.

Method 3 A method that uses Procedure Assemble-with-copy instead of Procedure Assemble in Procedure Fanin-rest in Method 2.

Method 4 Fan-in restricted Transduction Method which restricts the number of fan-ins of a gate by Method 1.

Method 5 Fan-in restricted Transduction Method which restricts the number of fan-ins of a gate by Method 2.

Table 1: Results of Method 1, 2 and 3 (gates/connections/levels/W-ORs).

Circuit	Method 1	Method 2	Method 3	Improvement
5xp1	126/332/7	93/338/4/16	94/341/4/13	7→4
Z9s	145/387/13	284/787/4/30	259/752/4/2	13→4
adr4	158/466/9	146/519/7/23	110/403/4/9	9→4
alu2	199/518/9	234/602/7/16	234/602/7/13	9→7
alu3	173/456/9	223/568/7/15	223/568/7/13	9→7
apla	244/568/9	237/573/7/7	233/566/7/9	9→7
con1	24/67/7	22/65/5/1	22/65/5/1	7→5
dc2	151/398/9	158/447/7/19	154/457/7/16	9→7
f51m	155/440/9	120/426/7/20	122/438/7/12	9→7
misex1	70/189/9	63/193/7/4	65/199/7/5	9→7
mlp4	536/1539/9	616/1782/9/78	507/1650/7/37	9→7
radd	138/411/9	124/419/7/18	110/403/4/9	9→4
risk	135/328/7	129/326/7/2	127/335/7/2	7→7
root	212/538/11	239/645/9/27	204/593/7/13	11→7
sao2	457/1179/11	483/1237/9/41	545/1394/7/22	11→7
sex	82/202/7	84/207/7/1	88/221/7/2	7→7
sqn	181/471/9	159/502/9/24	165/520/7/15	9→7
sqr6	130/339/7	112 /381/5/21	121/402/5/15	7→5
z4	105/314/7	81/296/5/14	69/253/4/7	7→4

4.2 Experimental Results

We performed the experiments of the transformation methods mentioned in Section 4.1 on MCNC benchmark circuits that were generated by the method in [9]. The results of Method 1, 2 and 3 are shown in Table 1. In Table 1, the numbers of gates, connections and levels are shown from the left in the column "Method 1", and the numbers of gates, connections, levels and Wired-OR gates are shown from the left in the columns "Method 2" and "Method 3". The column "Improvement" shows the level improvement of Method 2 and 3 as compared with Method 1. The results of Method 4 and 5 are shown in Table 2. In Table 2, the numbers of gates, connections and levels are shown from the left in the column "Method 4", and the numbers of gates, connections, levels and Wired-OR gates are shown from the left in the column "Method 5". The numbers of Wired-OR gates are omitted in the columns "Method 1" and "Method 4" because Wired-OR gates are not used in the methods.

4.3 Consideration

In most circuits, especially about levels, Method 2 and 3 which utilize Wired-OR gates are better than Method 1 which does not utilize Wired-OR gates. In some circuits such as sex, however, Method 1 which does not utilize Wired-OR gates can produce the results that have the least numbers of gates and connections among the three methods. The reason is thought as follows. In some cases, the capability of the circuit transformation is reduced by using Wired-OR gates, and so Generalized Serial Duplication is not able to be done efficiently, therefore the numbers of gates and connections are increased. However, it is clear that the average number of fan-ins of a NOR gate can be reduced by using Wired-OR gates even if the number of levels

Table 2: Results of Method 4 and 5 (gates/connections/levels/W-ORs).

Circuit	Method 4	Method 5
5xp1	98/269/9	80/252/7/3
Z9s	140/338/15	77/275/17/6
adr4	123/363/9	86/289/8/8
alu2	106/254/9	64/211/9/2
alu3	93/241/9	69/218/8/4
apla	118/288/9	69/239/7/3
con1	17/43/5	15/41/4/1
dc2	102/263/10	58/217/9/5
f51m	133/359/13	102/328/14/7
misex1	32/88/8	30/84/8/1
mlp4	414/1017/19	175/777/18/22
radd	105/282/9	76/270/12/8
risk	91/222/6	59/190/5/4
root	121/304/12	67/270/9/7
sao2	176/444/27	104/386/29/10
sex	49/125/5	43/119/5/2
sqn	113/291/12	73/250/10/5
sqr6	104/253/9	68/217/8/8
z4	95/269/7	62/215/8/8

is not reduced. From the results, we can observe that it is efficient to use Wired-Logic in the circuit design for especially level reduction.

Next we compare the results of Method 2 and Method 3. In most circuits, Method 3 produces better results than Method 2 especially about levels. In some cases, however, Method 3 produces the circuits that have a little more numbers of gates and connections than Method 2 because Method 3 adds redundant gates. There are some cases when Method 3 produces the circuits that do not have so much less numbers of levels than Method 2, though Method 3 is expected to restrict the numbers of fan-ins without increasing the numbers of levels. The reason is thought as follows. The initial circuits are 3-level circuits, and so there are many gates that are connected to input terminals. Therefore, Procedure Assemble-with-copy can not be used when assembling the input connections of such gates. However, Method 3 is though to be useful in many times because we can restrict the number of fan-ins of a gate without increasing the number of levels by this method if the gate is not connected to input terminals.

Finally, a comparison between Method 4 and Method 5 is given. In most circuits, Method 5 which utilize Wired-OR gates are better than Method 4. In some circuits, however, Method 4 can produce the results that have less numbers of levels than Method 5. The reason is thought in the same way as the comparison between Method 1 and Method 2 as follows: the capability of the circuit transformation is reduced by using Wired-OR gates, and the final result becomes worse. Finding the condition that the result becomes worse if a Wired-OR gate is used is a future work.

5 Conclusion and Future Work

In this paper, we have proposed the relaxed restrictions on a Wired-Logic gate by using the notion of *Connectable/Disconnectable* in Transduction Method. We have also presented the circuit transformation methods using Wire-OR gates. Since Wired-Logic gates has much shorter delay than ordinary gates, it is considered that they are useful for the circuit transformation. Although only NOR gates and Wired-OR gates have been treated in this paper, the extension to other types of gates can be made easily. We have done some experiments of proposed methods. The results show the effectiveness of using Wired-Logic at transforming circuits for especially level reduction. We have also showed that Fan-in restricted Transduction Method with Wired-Logic can perform the efficient circuit transformation.

In this paper, an integration of using Wired-OR gates with C/DC(Connectable/Disconnectable) in Transduction Method were experimented. An integration of using Wired-OR gates with another method of Transduction Method is a future work.

We think that the experiments presented in this paper are not general since all the initial circuits are 3-level NOR circuits. A major reason why the results are not so good as expected is thought to be that there are many gates that are connected to input terminals. It is a future work to experiment on more general circuits, i.e., circuits that has more numbers of levels.

Acknowledgements

The authors would like to thank Mr. Sunao Sawada in Kyushu University and Mr. Hideyuki Takada in Mitsubishi Electric Co. for their advice. The authors also thanks Professor Shuzo Yajima and the members of Yajima Laboratory for permitting to use the SBDD package developed by his group. This research was supported by International Research Program : Joint Research from the Ministry of Education, Science and Culture in Japan.

References

[1] H.Lee and E.S.Davidson, "A transform for NAND network design," *IEEE Trans. Comput.,* Vol.c-21, No.1, pp.12-20 (1972).

[2] Y.Kambayashi and S.Muroga, "Simplification of circuits based on permissible functions (in Japanese)," *Technical Report of IEICE*, AL73-13 (1973).

[3] Y.Kambayashi and S.Muroga, "Properties of wired logic," *IEEE Trans. Comput.,* Vol.c-35, No 6,pp.550-563 (1986).

[4] Y.Matsunaga and M.Fujita, "Multi-level logic optimization using binary decision diagrams," *Proc. of International Conference of Computer Aided Design*, pp.556-559 (1989).

[5] S. Muroga, Y. Kambayashi, H. C. Lai and J. N. Culliney, " The transduction method - design of logic networks based on permissible functions," *IEEE Trans. Comput.*, Vol.38, No.10, pp.1404-1424 (1989).

[6] H.Sato, Y.Yasue, Y.Matsunaga and M.Fujita, "Boolean resubstitution with permissible functions and binary decision diagrams," *Proc. of 27th Design Automatation Conference*, pp.284-289 (1990).

[7] Y.Kambayashi, H.C.Lai and S.Muroga, "Pattern-oriented transformations of NOR networks," UIUCDCS-R-90-1573, Dep. Comput. Sci., Univ.of Illinois (1990).

[8] T.Fujimoto, H.Honjo and T.Kambe, "Large network optimization using maximal CSPF," *Report of Technical Group on Design Automation IPS Japan*, 91-DA-60 ,pp.123-130 (1991).

[9] S.Sawada, Y.Kambayashi and S.Muroga, "Generation of fan-in restricted initial networks for transduction method," *Proc. the Synthesis and Simulation Meeting and International Interchange, SASIMI '92*, pp. 36-45 (1992).

[10] H.Takada, H.Ishigaki, Y.Kambayashi, "A method for realization of fan-in restricted transduction method based on efficient serial duplication (in Japanese)," *Proc. of the 46th Annual National Convention of IPS Japan*, 8M-4 (1993).