Title	Optimal Time-Multiplexing in Inter-FPGA Connections for Accelerating Multi-FPGA Prototyping Systems
Authors	Masato Inagi, Yasuhiro Takashima, Yuichi Nakamura, Atsushi Takahashi
Citation	IEICE Trans. Fundamentals, Vol. E91-A, No. 12, pp. 3539-3547
Pub. date	2008, 12
URL	http://search.ieice.org/
Copyright	(c) 2008 Institute of Electronics, Information and Communication Engineers

PAPER *Special Section on VLSI Design and CAD Algorithms*

Optimal Time-Multiplexing in Inter-FPGA Connections for Accelerating Multi-FPGA Prototyping Systems

Masato INAGI[†a)], **Yasuhiro TAKASHIMA**[††], **Yuichi NAKAMURA**[†††], ***and*** **Atsushi TAKAHASHI**[††††], ***Members***

SUMMARY In multi-FPGA prototyping systems for circuit verification, serialized time-multiplexed I/O technique is used because of the limited number of I/O pins of an FPGA. The verification time depends on a selection of inter-FPGA signals to be time-multiplexed. In this paper, we propose a method that minimizes the verification time of multi-FPGA systems by finding an optimal selection of inter-FPGA signals to be time-multiplexed. In the experiments, it is shown that the estimated verification time is improved 38.2% on average compared with conventional methods.

key words: *FPGA prototyping, ILP, I/O pins constraint, verification, time-multiplexed I/O*

1. Introduction

Circuit verification is one of the essential processes of VLSI design. In circuit verification, it is often requested to use very large test benches. For example, test benches corresponding to 1-hour video streams are used in verification of video encoding/decoding chips with 100 MHz system clock. In such cases, software RTL simulations are too slow to adopt, and FPGA prototyping systems are used. Even if an FPGA prototyping system with 1 MHz system clock is used, it takes 4 days to finish verification once. The verification time tends to increase according to the increase in size and complexity of VLSIs. The desire to shorten verification time is becoming stronger.

Although the device capacity of FPGAs is becoming very large, circuits to be designed are often larger than the leading edge FPGAs. Therefore, a prototyping system for such circuits consists of multiple FPGAs. A large circuit is partitioned into several sub-circuits each of which is implemented into an FPGA ([4], [5]). Each sub-circuit has to communicate with other sub-circuits through I/O pins of the FPGA. The number of I/O pins of an FPGA is becoming larger. However, the number of inter-FPGA signals from a sub-circuit is usually larger than the number of I/O pins of an FPGA even if a circuit is partitioned so that the number of inter-FPGA signals is minimized. Moreover, there exists a signal path that is cut several times by partitioning even if its minimization is pursued.

In order to relax the I/O pins constraint, time-multiplexed I/O (TM I/O) technique, such as in [6], is used in multi-FPGA prototyping systems. When TM I/O technique is employed, an I/O pin is used as a TM I/O, which is shared by multiple signals by time-division. Multiple inter-FPGA signals are transmitted by a TM I/O in one system clock period, but an inter-FPGA signal may be requested to be transmitted several times in one system clock period. An inter-FPGA signal transmission is far slower than an intra-FPGA signal transmission, and the required time to complete inter-FPGA signal transmissions by a TM I/O is far larger than that by a normal I/O. Therefore, the system clock period is mainly bounded by inter-FPGA signal transmissions by TM I/Os.

In TM I/O model, one system clock period is divided into several time slots, and an inter-FPGA signal is transmitted in a time slot by a TM I/O. The length of a time slot should be large enough to complete an inter-FPGA signal transmission. In our model, one system clock period is divided into several stages and each stage is divided into several time slots, which are the same as in [6]. Since the system clock period is mainly bounded by inter-FPGA signal transmissions by TM I/Os, the system clock period is estimated by the product of two indices, #slot, the number of time slots in one stage, and #stage, the number of stages in one system clock period.

When TM I/O technique was first introduced in a multi-FPGA prototyping system, all inter-FPGA signals were time-multiplexed and all I/O pins were used as TM I/Os (i.e. [6]) since the number of I/O pins is far smaller than the number of inter-FPGA signals. In such cases, #stage is determined by a given partitioning and #slot is minimized under the #stage. But, the system clock period is not reduced enough since #stage tends to be large in practical cases.

Recently, the number of I/O pins of an FPGA has been increased in some degree. It is becoming possible to use I/O pins as not only TM I/Os but also normal I/Os. In such case, #stage and #slot depend on a selection of inter-FPGA signals to be time-multiplexed, for a given circuit partitioning and I/O constraint. Thus, we propose an ILP-based method for minimizing the minimum system clock period by optimally selecting inter-FPGA signals to be time-multiplexed. Our method finds an optimal inter-FPGA signal selection from the solutions of several ILP problems which can efficiently be solved by integer relaxation. This ILP-based scheme is

Manuscript received March 24, 2008.
Manuscript revised June 24, 2008.

[†] The author is with the Graduate School of Information Sciences, Hiroshima City University, Hiroshima-shi, 731-3194 Japan.

[††] The author is with the Faculty of Environmental Engineering, the University of Kitakyushu, Kitakyushu-shi, 808-0135 Japan.

[†††] The author is with the System IP Core Research Laboratories, NEC, Kawasaki-shi, 211-8666 Japan.

[††††] The author is with the Department of Communications and Integrated Systems, Tokyo Institute of Technology, Tokyo, 152-8550 Japan.

a) E-mail: inagi@hiroshima-cu.ac.jp

DOI: 10.1093/ietfec/e91–a.12.3539

applicable to FPGA systems with more than two FPGAs by formulating an ILP problem for each connected pair of FPGAs and merging the problems, but in this paper we deal with dual-FPGA systems for simplicity. In the experiments, the estimated verification time for ISCAS89 benchmark circuits and industrial circuits are improved 38.2% on average compared with the case that all I/O pins are used as TM I/Os.

The rest of this paper is organized as follows. In Sect. 2, we give some definitions. In Sect. 3, our time-multiplexed I/O model is introduced. Our inter-FPGA signal selection method is proposed in Sect. 4. Experimental results are shown in Sect. 5. Finally, we give some conclusions.

2. Preliminaries

A circuit is modeled as a hyper-graph $G(V, E)$, called a circuit graph, which represents a circuit netlist. V is the set of vertices corresponding to cells (e.g. gates, flipflops) including primary-I/Os. Hereinafter, a vertex $v \in V$ and the cell corresponding to v are used interchangeably. E is the set of hyper-edges corresponding to nets which electrically connect cells. Each element $e \in E$ is the set of vertices to which the net corresponding to e connects. Hereinafter, a hyper-edge $e \in E$ and the net corresponding to e are used interchangeably.

Decomposition of a circuit into sub-circuits is called circuit partitioning. Especially circuit partitioning into two parts is called bipartitioning. Bipartitioning of a circuit graph $G(V, E)$ is to give the pair of vertex set V_1 and V_2 such that $V_1 \neq \emptyset, V_2 \neq \emptyset$, $V_1 \cup V_2 = V$, and $V_1 \cap V_2 = \emptyset$. V_1 and V_2 are called blocks. A pair $\Phi = (V_1, V_2)$ of vertex sets is called a partition. $e \in E$ is called a *cutnet* if $e \cap V_1 \neq \emptyset \wedge e \cap V_2 \neq \emptyset$. Bipartitioning whose objective is to minimize the number of cutnets is called min-cut partitioning. In partitioning for multi-FPGAs, the size (area) constraint, which limits the size of each block, is given to fit each block to an FPGA. In this paper, we deal with delay minimization problems for bipartitioned circuits with TM I/O for simplicity.

3. Time-Multiplexed I/O Model

In Fig. 1, the concept of our implementation is depicted. In our TM I/O implementation, one system clock period is divided into several stages, and each stage is divided into several time slots. An inter-FPGA signal is transmitted through inter-FPGA connection within a time slot. A transmission in a time slot is controlled by I/O clock. In each stage, the inter-FPGA signals assigned to a TM I/O are serially transmitted once. The inter-FPGA signals assigned to a TM I/O are regarded as parallel signals but are transmitted serially from a sending FPGA by using a parallel/serial converter which consists of a counter and flipflops in the sending FPGA. In a receiving FPGA, serially transmitted inter-FPGA signals are converted into parallel signals by using a serial/parallel converter and are kept until the end of the next stage. All our TM I/Os work synchronously by using an I/O clock. The number of stages in one system clock period and the number of time slots in one stage are the same for all TM I/Os. In our model, the consecutive stages are separated in time axis. The separation of stages is used to await the convergence of signal states after a stage, considering also the pre-stage and post-stage calculation between an FF and a TM I/O.

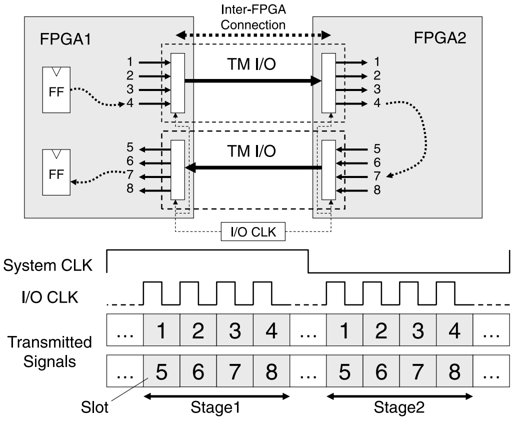

Fig. 1 Our time-multiplexed I/O (#slot=4, #stage=2).

In the following, the number of time slots in one stage is referred by #slot, and the number of stages in one system clock period is referred by #stage. That is, at most #slot signals are transmitted #stage times by a TM I/O in one system clock period. Thus, #slot and #stage are key indices of system speed. There may be several TM I/Os on a signal path between FFs. #slot and #stage need to be set such that the signal transmission along such a path are realized to simulate the circuit behavior.

There are several approaches to use time-multiplexed I/O technique. Let us consider a sophisticated approach for comparison with ours. In the approach, each inter-FPGA signal is transmitted by a TM I/O, and a TM I/O transmits different inter-FPGA signals in different slots in one system clock period. Each inter-FPGA signal is transmitted only once in one system clock period. The time slot in which an inter-FPGA signal is transmitted is decided by estimating the time when the signal is valid and by assigning an appropriate time slot of an appropriate TM I/O. The granularity of the time slot assignment depends on implementation methods. For example, in [6], all the slots in one stage are equivalent in the assignment. In this sophisticated approach, the search space of the assignment is huge, and it is not easy to find a feasible assignment. The estimation is not easy task. Even if the estimation is possible, the conflicts between inter-FPGA signals should be resolved by scheduling and allocation. The actual time would vary when the assigned TM I/O is different. There is no guarantee of convergence. Also, errors of an inter-FPGA signal before or after the time slot to which the signal is assigned may not be detected because of estimation errors. It is not easy to distinguish the circuit errors from implementation errors. Even though such a sophisticated approach may realize faster verification speed, they decreases the reliability of circuit verification, which is one of the most critical issues of circuit verification. Since we emphasize the reliability of verification, we adopt a simple TM I/O design to maintain the reliability. Our objective is to maximize verification speed without loss of reliability.

In our approach, only selected inter-FPGA signals are transmitted by TM I/Os, and the others are done by normal I/Os. A selected signal is assigned to a TM I/O and transmitted #stage times in one system clock period. Because of the simple action of our TM I/O, the amount of additional circuits in our approach is less than those in sophisticated approaches. This is desirable for circuit verification. A feasible implementation is easily obtained by setting the separation of stages, #stage, and #slot appropriately. System speed is improved by reducing #stage, considering also #slot. It is realized by using both of TM I/Os and normal I/Os and reducing the maximum number of TM I/Os on a path between FFs. Our proposed method obtains an optimal selection of inter-FPGA signals to be time-multiplexed which maximizes estimated system speed.

Assume that a circuit is bipartitioned and each subcircuit is implemented into a FPGA. In addition, assume that a selection of inter-FPGA signals to be time-multiplexed is decided. Let P be the number of I/O pins of a target FPGA. Let m be the number of inter-FPGA signals. Let TM-count be the number of time-multiplexed inter-FPGA signals. Let TM-depth of a path be the number of time-multiplexed inter-FPGA signals along the path. Let TM-depth of a circuit be the maximum TM-depth among signal paths between FFs. When we simply write TM-depth, we means the TM-depth of a circuit. In our model, separation is necessary between stages in order to await the completion of the logic calculation between TM I/Os. Considering also the pre-stage and post-stage calculation between an FF and a TM I/O, the separation is set to be greater than the maximum signal delay from an FF to the input of a TM I/O, from the output of a TM I/O to the input of a TM I/O, and from the output of a TM I/O to an FF. #stage must be greater than TM-depth to propagate signals from FFs to FFs, and should be as small as possible. Therefore, we set

$$\#\text{stage} = \text{TM-depth}.$$

#slot is set to the ceiling of the number of time-multiplexed inter-FPGA signals over the number of TM I/Os. Since the numbers of I/Os used as normal I/Os is m – TM-count, I/Os which can be used as TM I/Os are $P - m +$ TM-count. Thus,

$$\#\text{slot} = \left\lceil \frac{\text{TM-count}}{P - m + \text{TM-count}} \right\rceil.$$

In our assumption, $m - P > 0$. Since, $m - P$ signals cannot be transmitted without TM I/O, TM-count should be larger than $m - P$ to accommodate all inter-FPGA signals. That is,

$$\text{TM-count} > m - P.$$

Let C_{sys} and C_{io} be the system clock period and the I/O clock period, respectively. Let I_{s} be the separation between stages. Each TM I/O transmits #slot signals #stage times. Logic calculations between TM I/Os are performed #stage−1 times. Logic calculations from FFs to TM I/Os, and from TM I/Os to FFs are performed once for each. Thus,

$$C_{\text{sys}} \geq \#\text{stage} \cdot \#\text{slot} \cdot C_{\text{io}} + (\#\text{stage} + 1) \cdot I_{\text{s}}.$$

Since it is difficult to estimate I_{s} accurately immediately after partitioning, I_{s} is set with large margin. However, it is small compared with #slot $\cdot\, C_{\text{io}}$. Therefore, we simply evaluate C_{sys} by #stage $\cdot$ #slot $\cdot\, C_{\text{io}}$. Since C_{io} is considered as invariant, in order to shorten the system clock period, small #slot and small #stage are desirable. Since $P - m < 0$ in our assumption, #slot is reduced if TM-count is increased. However, if TM-count increases, #stage increases since TM-depth tends to increase. There is a trade-off between #slot and #stage.

4. Optimal Selection of Signals to Be Time-Multiplexed

In this section, we propose a method to shorten verification time by a multi-FPGA prototyping system with TM I/O technique. When a circuit partition is given, the verification time is roughly estimated by #stage times #slot, both of which are decided by a selection of inter-FPGA signals to be time-multiplexed, and I/O pins constraint. To shorten the verification time, we find a selection of inter-FPGA signals to be time-multiplexed such that the verification time is minimized satisfying I/O pins constraint. Inter-FPGA signals and TM I/Os have directions, and directions must match. We utilize optimal inter-FPGA signal selections ignoring the I/O direction to efficiently obtain an optimal inter-FPGA signal selection considering the I/O direction.

First, we define a signal graph to simplify the representation of reachability between inter-FPGA signals. Then, we define an inter-FPGA signal selection problem to minimize the verification time under the given number of I/O pins as an 0-1 integer linear programming problem, ignoring the I/O direction. Our proposing method solves the problem by decomposing the problem into a reasonable number of sub-problems. The objective of the sub-problems is to maximize the number of signals to be time-multiplexed under the given #stage. The sub-problems are efficiently solved by integer relaxation because of the total-unimodularity of their coefficient matrices. Then, we can calculate the optimal #slot for each #stage from I/O pins constraint. The optimal selections for the sub-problems include an optimal inter-FPGA signal selection for the original problem. The selection with the minimum #slot times #stage would be adopted. Next, we introduce a method to consider the I/O direction.

4.1 Inter-FPGA Signal Graph

Given a circuit graph G and its bipartition Φ, we represent the connections between inter-FPGA signals by a directed graph $G_{\text{i/o}}(V_{\text{i/o}}, E_{\text{i/o}})$ called an inter-FPGA Signal Graph, where $V_{\text{i/o}} = \{v_1, v_2, \ldots, v_m\}$ is the set of vertices corresponding to the inter-FPGA signals and $E_{\text{i/o}} = \{e_1, e_2, \ldots, e_l\}$ is the set of directed edges corresponding to the direct signal paths

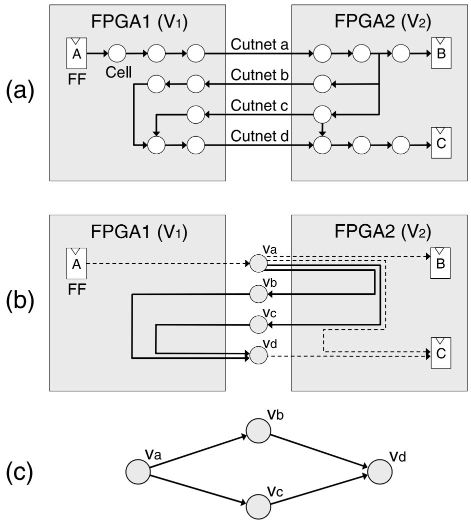

Fig. 2 (a) Circuit graph and its bipartition, (b) Relation of inter-FPGA signals, (c) inter-FPGA signal graph.

between the inter-FPGA signals (Fig. 2). Let $V_{\mathrm{FI}}(v)(v \in V_{\mathrm{i/o}})$ be the set of inter-FPGA signals (vertices in $V_{\mathrm{i/o}}$) from which there are signal paths to v through no other inter-FPGA signals. A vertex $v \in V_{\mathrm{i/o}}$ which has no incoming edges is called a source of $G_{\mathrm{i/o}}$. A vertex $v \in V_{\mathrm{i/o}}$ which has no outgoing edges is called a sink of $G_{\mathrm{i/o}}$.

4.2 Problem Formulation

Here, we define a selection problem of inter-FPGA signals to be time-multiplexed whose objective is to minimize the system clock period under the given number of I/O pins.

A circuit graph G and its bipartition $\Phi = (V_1, V_2)$ are given. Let P be the number of I/O pins of a target FPGA. Let $G_{\mathrm{i/o}}(V_{\mathrm{i/o}}, E_{\mathrm{i/o}})$ be the inter-FPGA signal graph constructed from G and Φ. Let m be the number of inter-FPGA signals, i.e. $|V_{\mathrm{i/o}}|$. #slot and #stage are calculated as follows.

Let x_v be a 0-1 integer label of $v \in V_{\mathrm{i/o}}$ which represents if v is selected to be time-multiplexed or not. $x_v = 0$ represents that v is not selected and $x_v = 1$ represents that v is selected. Clearly,

$$\text{TM-count} = \sum_{i=1}^{m} x_{v_i}. \tag{1}$$

Let y_v be an integer label of $v \in V_{\mathrm{i/o}}$ which represents the maximum TM-depth among all the paths from a source to v. y_v is recursively calculated from the maximum TM-depth $y_{v'}$ among the paths from a source to the predecessors $v' \in V_{\mathrm{FI}}(v)$, and x_v, which represents if v is time-multiplexed or not. That is, for each $v \in V_{\mathrm{i/o}}$,

$$y_v = \begin{cases} x_v & \text{if } V_{\mathrm{FI}}(v) = \emptyset \\ \max_{v' \in V_{\mathrm{FI}}(v)} y_{v'} + x_v & \text{if } V_{\mathrm{FI}}(v) \neq \emptyset \end{cases}. \tag{2}$$

TM-depth is the maximum TM-depth among all the paths between FFs, and y_v represents the maximum TM-depth among all the paths from FFs to v. Thus,

$$\text{TM-depth} = \max_{v \in V_{\mathrm{i/o}}} y_v. \tag{3}$$

Therefore,

$$\#\text{stage} = \text{TM-depth} = \max_{v \in V_{\mathrm{i/o}}} y_v,$$

$$\#\text{slot} = \left\lceil \frac{\text{TM-count}}{P - m + \text{TM-count}} \right\rceil \tag{4}$$

$$= \left\lceil \frac{\sum_{i=1}^{m} x_{v_i}}{P - m + \sum_{i=1}^{m} x_{v_i}} \right\rceil. \tag{5}$$

The selection problem of inter-FPGA signals to be time-multiplexed is defined as below.

Signal Selection Problem (SS Problem)

input an inter-FPGA signal graph $G_{\mathrm{i/o}}(V_{\mathrm{i/o}}, E_{\mathrm{i/o}})$,
the number of I/O pins P
output $x_v(\forall v \in V_{\mathrm{i/o}})$
objective minimization of #stage · #slot
constraint I/O pins constraint

4.3 LP-Based System Clock Minimization

We propose a method which optimally solves the SS problem by using a linear programming problem solver (LP-solver). First, we define an integer linear programming (ILP) problem whose objective is to maximize TM-count under the given TM-depth. #slot is reduced by increasing TM-count. We show that the ILP problem is optimally solved by using a non-integer LP-solver, which effectively solves LP problems. Then, the minimum value of #slot is derived from #stage, P, and TM-count. That is, #slot for optimally minimizing #slot·#stage under given #stage is effectively obtained. Since the maximum number of gates on a path between FFs in a latest circuit is generally made less than about 50, a possible value of #stage is from 1 to about 50. Furthermore, the maximum number of inter-FPGA signals on a path between FFs is empirically less than 20. Thus, this problem is solved by applying an LP-solver only less than 20 times.

4.3.1 ILP Formulation Ignoring the I/O Direction

A maximum function is substituted by parallel inequalities in an LP formulation. Thus, Eqs. (2) and (3) can be transformed into LP inequalities. We define an N-depth inter-FPGA signal selection problem whose objective is maximization of TM-count as an ILP problem.

N-Depth Signal Selection Problem (N-SS Problem)

constants:

N: #stage;

variables:

$x_{v_1}, x_{v_2}, \ldots, x_{v_m}$: 0-1 integers;
$y_{v_1}, y_{v_2}, \ldots, y_{v_m}$: integers;

subject to:

$$0 \leq x_{v_i} \leq 1 \quad (0 \leq i \leq m)$$

$$0 \leq y_{v_i} \leq N \quad (0 \leq i \leq m)$$

$$\left.\begin{array}{ll} x_v \leq y_v & \text{if } V_{\mathrm{FI}}(v) = \emptyset \\ y_{v'} + x_v \leq y_v, \forall v' \in V_{\mathrm{FI}}(v) & \text{if } V_{\mathrm{FI}}(v) \neq \emptyset \end{array}\right\} \forall v \in V_{\mathrm{i/o}} \tag{6}$$

maximize: $\sum_{i=1}^{m} x_{v_i}$.

In the same way as in Sect. 4.2, x_v represents whether the signal v is selected to be time-multiplexed, and y_v represents an upper bound of TM-depth from the source vertices to v. Let us consider an inter-FPGA signal $v \in V_{\mathrm{i/o}}$. y_v is greater than or equal to x_v added by $y_{v'}$ where v' is a direct predecessor of v. The relation between v and v' is represented by Inequity (6). Since the relation is transitive, Inequity (6) covers the relation between inter-FPGA signal selection and upper bound of TM-depth. Since y_v has an upper bound N, TM-depth of any feasible inter-FPGA signal selection of N-SS problem is less than or equal to N. That is, #stage $=N$ is realized. The objective function represents TM-count.

By solving this ILP problem for each #stage, we obtain pairs of #stage and #slot for P. The best inter-FPGA signal selection is that for #stage minimizing #stage · #slot.

Next, we show that N-SS problems are optimally, and efficiently solved by non-integer LP solver.

4.3.2 Total-Unimodularity

According to the high capability of linear programing, it has been being actively studied. As one of the important property of LP, it is commonly known that if the coefficient matrix of an ILP problem is totally unimodular, the basic solutions of the LP problem made by relaxing integer constraints of the ILP problem consists of only integers [7]. An optimal basic solution of an LP problem is efficiently obtained by simplex algorithms. Thus, an optimal solution of such an ILP problem can also be efficiently obtained.

Definition 4.1: A matrix A is called totally-unimodular if all the determinants of sub-matrices of A are 0, 1 or -1.

According to the ILP formulation, this problem belongs to the integer delay budgeting problem handled in [9]. As mentioned in [9], the coefficient matrix of any instance of the integer delay budgeting problem is totally-unimodular. Therefore, the following theorem is also formed:

Theorem 4.1: The coefficient matrix of an N-depth signal selection problem is totally unimodular.

N-depth signal selection problems are efficiently solved by simplex algorithms for LP problems.

4.4 Consideration of the Direction of I/O Signal

Each inter-FPGA signal has its direction, from block V_1 to block V_2, or from block V_2 to block V_1. A TM I/O cannot transmit two signals with different directions. Thus, signals with different direction must be transmitted by different TM I/Os. Suppose that an optimal selection without considering I/O direction is given. Let TM-countL and TM-countR be the number of selected signals with one of the directions and the number of selected signals with the other direction, respectively. If

$$\left\lceil \frac{\text{TM-count}}{\#\text{slot}} \right\rceil = \left\lceil \frac{\text{TM-countL}}{\#\text{slot}} \right\rceil + \left\lceil \frac{\text{TM-countR}}{\#\text{slot}} \right\rceil,$$

the selection is feasible even considering the direction since the number of required I/O pins does not change. Also, if and only if

$$P \leq m - \text{TM-count} + \left\lceil \frac{\text{TM-countL}}{\#\text{slot}} \right\rceil + \left\lceil \frac{\text{TM-countR}}{\#\text{slot}} \right\rceil, \tag{7}$$

the selection is feasible even considering the direction. Otherwise, the selection is not optimal when considering I/O direction. For such a case, we extends the ILP formulation of N-depth inter-FPGA signal selection problem as N-D-SS problem by considering the direction of inter-FPGA signals. The expanded SS problem considering the I/O direction is called a D-SS problem. Since the coefficient matrix of the extended formulation is not totally unimodular, and #slot must be given as a constant in the formulation, we utilize also the solutions of N-SS problems to reduce the times of solving N-D-SS problems.

4.4.1 ILP Formulation

A TM I/O transmits multiple inter-FPGA signals from the same block to the other block. Thus, inter-FPGA signals assigned to the same TM I/O must have the same direction. We say a TM I/O which transmits signals from block V_1 to block V_2, an L-TM I/O (left-to-right TM I/O). An R-TM I/O (right-to-left TM I/O) is defined in the same way. The number of necessary TM I/Os depends on the number of inter-FPGA signals to L-TM I/Os (TM-countL), the number of inter-FPGA signals to R-TM I/Os (TM-countR), and #slot. That is,

$$\#\text{TMIO} = \left\lceil \frac{\text{TM-countL}}{\#\text{slot}} \right\rceil + \left\lceil \frac{\text{TM-countR}}{\#\text{slot}} \right\rceil,$$

where #TMIO is the number of necessary TM I/Os. Since the equation is not linear, we give #slot as a constant to formulate the directed N-SS problem as an ILP problem.

N-Depth Directed Signal Selection Problem (N-D-SS Problem) constants:

M: #slot;
N: #stage;
$V_{\mathrm{l}}(\subset V_{\mathrm{i/o}})$: inter-FPGA signals from V_1 to V_2 in G;
$V_{\mathrm{r}}(\subset V_{\mathrm{i/o}})$: inter-FPGA signals from V_2 to V_1 in G;

variables:

$x_{v_1}, x_{v_2}, \ldots, x_{v_m}$: 0-1 integers;
$y_{v_1}, y_{v_2}, \ldots, y_{v_m}$: integers;
r and l: integers

subject to:

$$0 \le x_{v_i} \le 1 \quad (0 \le i \le m)$$

$$0 \le y_{v_i} \le N \quad (0 \le i \le m)$$

$$\sum_{v_i \in V_{\mathrm{l}}} x_{v_i} \le M \cdot l \tag{8}$$

$$\sum_{v_i \in V_{\mathrm{r}}} x_{v_i} \le M \cdot r \tag{9}$$

$$\left.\begin{array}{ll} x_v \le y_v & \text{if } V_{\mathrm{FI}}(v) = \emptyset \\ y_{v'} + x_v \le y_v, \forall v' \in V_{\mathrm{FI}}(v) & \text{if } V_{\mathrm{FI}}(v) \ne \emptyset \end{array}\right\} \forall v \in V_{\mathrm{i/o}}$$

maximize: $\sum_{i=1}^{m} x_{v_i} - r - l$.

The variables l and r represent the number of TM I/Os which transmit signals from V_1 to V_2, and from V_2 to V_1, respectively. Inequality (8) makes TM-countL less than or equal to the acceptable number of inter-FPGA signals by l L-TM I/Os. Inequality (9) works in the same way. The objective function is the number of time-multiplexed inter-FPGA signals subtracted by the number of L-TM I/Os and R-TM I/Os. l L-TM I/Os and r R-TM I/Os are necessary to transmit $\sum_{i=1}^{m} x_{v_i}$ multiplexed inter-FPGA signals, which originally needed $\sum_{i=1}^{m} x_{v_i}$ I/O pins before multiplexing. Thus, the objective function represents the reduction of the number of required I/O pins.

4.4.2 Reduction of Times of Solving ILP Problems

To solve the ILP problem N-D-SS problem, #slot M and #stage N need to be fixed. As a straightforward approach, it is possible to solve the N-D-SS problems for all the possible pair of #slot and #stage. However, solving $M_{max} \cdot N_{max}$ ILP problems each of which has no guarantee to be solved in short time is not desirable, where M_{max} and N_{max} are the maximum possible value of #slot and #stage, respectively. Thus, we reduce the number of the ILP problems which needs to be solved, utilizing the solution of N-SS problems and a fact below.

Theorem 4.2: The number of required I/O pins considering the I/O direction is equal to or 1 greater than that ignoring the I/O direction for the same #slot and #stage.

Let N-D-SS(i,j) be an N-D-SS problem for #stage = i, #slot = j. Let N-SS(i) be an N-SS problem for #stage = i. Let SS(k) and D-SS(k) be an SS problem and a D-SS problem for $P = k$, respectively.

A lower bound of j for N-D-SS(i,j) whose solution is optimal for D-SS(k) can be obtained by solving the corresponding N-SS(i) and $P = k$. Let $M_{low}(i, k)$ be the lower bound of j for N-D-SS(i,j) and $P = k$. That is, the solutions of N-D-SS(i,j) such that $j \ge M_{low}(i, k)$ includes optimal solutions of D-SS(k).

An upper bound of the minimum #slot for D-SS(k) is $M_{up}(i, k) = M_{low}(i, k - 1)$, since feasible solutions of the SS($k - 1$) also satisfy the constraints of D-SS(k) problem. That is, the solutions of N-D-SS(i,j) such that $j \le M_{up}(i, k)$ include optimal solutions of D-SS(k).

Therefore, an optimal solution is obtained by scanning $1 \le i \le N_{max}$ and $M_{low}(i, k) \le j \le M_{up}(i, k)$ on N-D-SS(i,j).

The procedure for solving a D-SS(k) problem is as follows.

1. Enumerate the candidate pairs of #stage and #slot (i, j) by the rule above.
2. Sort the candidate pairs in ascending order of #slot · #stage.
3. The procedure in the following is applied for each candidate pair in the order.
   a. If the solution of N-SS(i) also satisfies the constraints of D-SS(k), adopt the solution and quit the procedure.
   b. If not, solve the N-D-SS(i,j).
   c. If the solution satisfies the constraints of D-SS(k), adopt the solution and quit the procedure.

It is unlikely that many of the candidate pairs do not satisfy the condition in step 3a. Thus, the number of times of solving the ILP problem is drastically reduced.

5. Experimental Results

We compared our proposals with a method in which all the inter-FPGA signals are time-multiplexed to evaluate the effectiveness of our proposal. We used ILOG CPLEX 10.0 as a linear programming optimization engine to solve the LP and ILP problems of our proposed method. The experiments

Table 1 Benchmark circuits.

Circuit	Gates	Nets	FFs	Cutnets	#const.
s5378	2958	2993	179	182	367
s9234.1	5808	5844	211	307	959
s13207.1	8589	8651	638	330	1148
s15850.1	10306	10383	534	551	1499
s38417	23815	23843	1636	1087	4853
s38584.1	20679	20717	1426	1962	5346
A	251636	253215	24119	22611	87111
B	58011	52217	8181	4127	11565
C	202693	204582	18708	19566	88487
D	33036	33460	3794	2565	7078
E	177818	181962	14672	16737	57057
F	164204	167232	17487	15317	51642

Table 2 The number of time-multiplexed inter-FPGA signals.

	#stage													
Circuit	1	2	3	4	5	6	7	8	9	10	11	12	13	14
s5378	118	160	176	179	181	182	-	-	-	-	-	-	-	-
s9234.1	178	250	284	297	301	304	305	306	307	-	-	-	-	-
s13207.1	226	286	311	325	328	330	-	-	-	-	-	-	-	-
s15850.1	323	443	496	521	535	542	546	548	550	551	-	-	-	-
s38417	571	811	954	1032	1067	1082	1086	1087	-	-	-	-	-	-
s38584.1	1197	1700	1870	1931	1948	1958	1962	-	-	-	-	-	-	-
A	14592	19556	21343	22065	22364	22503	22564	22586	22596	22602	22606	22608	22610	22611
B	2749	3609	3916	4052	4096	4115	4125	4127	-	-	-	-	-	-
C	11621	16279	18138	18877	19216	19370	19466	19517	19545	19562	19566	-	-	-
D	1708	2268	2451	2511	2537	2548	2557	2563	2564	2565	-	-	-	-
E	10606	14491	16034	16552	16693	16728	16732	16735	16736	-	-	-	-	-
F	9567	13128	14584	15101	15256	15300	15314	15317	-	-	-	-	-	-

Table 3 The minimum clock period and its configuration of ISCAS89 benchmark circuits.

	#pin=10				#pin=20				#pin=40			
Circuit	Opt(Sl, St)	TM	ILP	All(Sl, St)	Opt(Sl,St)	TM	ILP	All(Sl, St)	Opt(Sl,St)	TM	ILP	All(Sl, St)
s5378	112(28, 4)	7	1	132(22, 6)	42(14, 3)	13	-	60(10, 6)	18(6, 3)	31	-	36(6, 6)
s9234.1	280(40, 7)	7	3	288(32, 9)	110(22, 5)	14	1	144(16, 9)	40(10, 4)	30	1	81(9, 9)
s13207.1	228(38, 6)	9	1	228(38, 6)	96(24, 4)	14	1	108(18, 6)	40(10, 4)	34	-	60(10, 6)
s15850.1	580(58,10)	10	3	580(58,10)	266(38, 7)	15	1	300(30,10)	108(18, 6)	31	-	160(16,10)
s38417	910(130, 7)	9	7	912(114, 8)	406(58, 7)	19	-	464(58, 8)	192(32, 6)	35	-	224(28, 8)
s38584.1	1400(200, 7)	10	1	1400(200, 7)	700(100, 7)	20	-	700(100, 7)	336(56, 6)	36	-	350(50, 7)
Ave.	0.970				0.852				0.692			
	#pin=80				#pin=160				#pin=320			
Circuit	Opt(Sl, St)	TM	ILP	All(Sl, St)	Opt(Sl,St)	TM	ILP	All(Sl, St)	Opt(Sl,St)	TM	ILP	All(Sl, St)
s5378	8(8, 1)	16	-	24(4, 6)	2(2, 1)	61	-	12(2, 6)	-(-, -)	-	-	-(-, -)
s9234.1	18(6, 3)	48	-	45(5, 9)	6(6, 1)	31	-	18(2, 9)	-(-, -)	-	-	-(-, -)
s13207.1	16(8, 2)	37	-	36(6, 6)	6(6, 1)	38	-	24(4, 6)	2(2, 1)	113	-	12(2, 6)
s15850.1	48(12, 4)	45	-	80(8,10)	18(6, 3)	83	-	40(4,10)	4(4, 1)	81	-	20(2,10)
s38417	90(18, 5)	60	-	112(14, 8)	40(10, 4)	104	-	64(8, 8)	16(4, 4)	259	-	32(4, 8)
s38584.1	150(30, 5)	65	-	182(26, 7)	64(16, 4)	121	-	98(14, 7)	30(10, 3)	188	-	56(8, 7)
Ave.	0.569				0.413				0.567			

#pin: the number of I/O pins of an FPGA, Sl: #slot, St: #Stage,
TM: the number of TM I/Os, ILP: the number of solved *N*-D-SS problems

were conducted on a PC equipped with AMD Opteron 265 (1.8 GHz) CPU, 3 GB Memory, CentOS 4.3. Circuits used in the experiments are 6 circuits from ISCAS89 Benchmark Suite [2] and 6 industrial circuits shown in Table 1. FM method [1] was used to obtain min-cut partitions of the circuits. Cutnets in the table is the number of cutnets by the partition, that is, the number of the inter-FPGA signals. # Const. is the number of constraint inequalities of the corresponding *N*-SS problem except the boundary conditions for each variable.

First, our method obtains an optimal inter-FPGA signal selection of an *N*-SS problem, which maximizes the number of time-multiplexed inter-FPGA signals for #stage *N*, by solving the *N*-SS problem using the LP solver. The results are shown in Table 2. As shown in the table, the maximum number of #stage was 14 at most. Every *N*-SS problem was solved within 1 sec.

Next, the minimum of #slot was obtained for each #stage *N* under the number of I/O pins *P*. The optimal combination of #slot and #stage(=TM-depth) which minimize the system clock period was selected. Then, the corresponding D-SS problem was solved utilizing the results of *N*-SS problems in the manner explained in the previous section. Tables 3 and 4 show the result of the selection of ISCAS89 benchmark circuits and industrial circuits, respectively. For ISCAS89 benchmark circuits, the number of I/O pins is set to very small value in order to emphasize the difference. For industrial circuits, practical numbers of I/O pins are given. In Tables 3 and 4, *Opt*, Sl, and St shows the minimum system clock period under the given *P*, its combination of #slot and #stage, which are obtained by our method, respectively. TM and ILP in the tables show the number of required TM I/Os and the number of solved *N*-D-SS problems, respectively. As shown in Table 3, at most seven *N*-D-SS problems were solved for a D-SS problem. Each *N*-D-SS problem was solved within 1 sec though it is not guaranteed that the problem is solved within polynomial time in worst case. *All* shows the minimum system clock period when all the inter-FPGA signals are time-multiplexed. For *All*, the best #slot when all the inter-FPGA signals are time-multiplexed are given. *Ave.* shows the ratio of *Opt* to *All* on average of all the circuits in each table.

As seen in Table 3, the minimum system clock period by our proposed method is the same as that by *All* when the number of I/O pins are very limited. The table shows that the improvement by our method compared with *All* becomes larger according to the increase of the number of I/O pins. Table 4 shows the capability of our method to actual circuits

Table 4 The minimum clock period and its configuration of industrial circuits.

	#pin=200		#pin=400		#pin=600	
Circuit	Opt(Sl,St) TM ILP	All(Sl, St)	Opt(Sl,St) TM ILP	All(Sl, St)	Opt(Sl,St) TM ILP	All(Sl, St)
A	1036(148, 7) 153 -	1596(114,14)	448(64, 7) 353 -	812(58,14)	276(46, 6) 490 -	532(38,14)
B	130(26, 5) 158 -	176(22, 8)	56(14, 4) 291 -	96(12, 8)	32(8, 4) 507 -	64(8, 8)
C	990(110, 9) 179 -	1078(98,11)	448(56, 8) 350 -	550(55,11)	280(40, 7) 487 -	374(34,11)
D	72(18, 4) 141 -	140(14,10)	30(10, 3) 246 -	80(8,10)	16(8, 2) 284 -	60(6,10)
E	528(88, 6) 191 -	756(84, 9)	240(48, 5) 349 -	378(42, 9)	160(32, 5) 522 1	252(28, 9)
F	504(84, 6) 183 -	624(78, 8)	230(46, 5) 332 -	320(40, 8)	150(30, 5) 509 -	208(26, 8)
Ave.	0.721		0.613		0.565	

	#pin=800		#pin=1000		#pin=1200	
Circuit	Opt(Sl,St) TM ILP	All(Sl, St)	Opt(Sl,St) TM ILP	All(Sl, St)	Opt(Sl,St) TM ILP	All(Sl, St)
A	204(34, 6) 663 -	420(30,14)	150(30, 5) 746 -	336(24,14)	120(24, 5) 932 -	280(20,14)
B	24(8, 3) 489 -	48(6, 8)	16(8, 2) 452 -	48(6, 8)	12(6, 2) 602 -	32(4, 8)
C	196(28, 7) 696 -	286(26,11)	150(30, 5) 642 -	220(20,11)	120(24, 5) 801 -	198(18,11)
D	12(6, 2) 378 -	40(4,10)	8(4, 2) 567 -	40(4,10)	6(6, 1) 286 -	40(4,10)
E	112(28, 4) 592 -	198(22, 9)	88(22, 4) 753 -	162(18, 9)	72(18, 4) 920 -	126(14, 9)
F	104(26, 4) 581 -	160(20, 8)	80(20, 4) 756 -	128(16, 8)	64(16, 4) 944 -	112(14, 8)
Ave.	0.531		0.472		0.450	

#pin: the number of I/O pins of an FPGA, Sl: #slot, St: #Stage,
TM: the number of TM I/Os, ILP: the number of solved *N*-D-SS problems

and FPGAs. The improvement ratio on average of all the settings of the number of I/O pins P in Tables 3 and 4 were 32.3% and 44.1%, respectively.

The improvement ratio on average in the experiments was 38.2%.

6. Conclusions

In this paper, we proposed a verification system speed acceleration method by optimizing a selection of inter-FPGA signals to be time-multiplexed, under the given number of I/O pins. In our method, the sub-problems of the inter-FPGA signal selection problem are formulated as ILP problems which can be efficiently solved by non-integer LP solver. Since the number of the sub-problems is limited, optimal solutions are efficiently obtained by our formulation. The direction of inter-FPGA signals is considered by solving a few additional simple ILP problems as a post-process.

Our future work includes simultaneous use of time-multiplexed I/Os which have different #slot, development of partitioning algorithms considering optimal inter-FPGA signal selections, and simultaneous optimization of a partition and its signal selection.

References

[1] C.M. Fiduccia and R.M. Mattheyses, "A linear time heuristic for improving network partitions," Proc. DAC82, pp.175–181, 1982.

[2] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," Proc. ISCAS89, pp.1929–1934, May 1989.

[3] J. Babb, R. Tessier, and A. Agarwal, "Virtual wires: Overcoming pin limitations in FPGA-based logic emulators," Proc. IEEE Workshop FCCM, pp.142–151, April 1993.

[4] J. Gateley, M. Blatt, D. Chen, S. Cooke, P. Desai, M. Doreswamy, M. Elgood, G. Feierbach, T. Goldsbury, and D. Greenley, "UltarSparc-I emulation," Proc. ACM/IEEE DAC95, pp.13–18, 1995.

[5] R. Brown, J. Hayes, and T. Mudge, "Rapid prototyping & evaluation of high-performance computers," Proc. Conf. Experimental Research in Computer Systems, NFS Experimental Systems, pp.159–168, June 1996.

[6] J. Babb, R. Tessier, M. Dahl, S. Hanono, D. Hoki, and A. Agarwal, "Logic emulation with virtual wires," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.16, no.6, pp.609–629, June 1997.

[7] W.J. Cook, W.H. Cunningham, W.R. Pullayblank, and A. Schrijver, Combinatorial Optimization, John Willy & Sons, 1998.

[8] K.R. Azegami, M. Inagi, A. Takahashi, and Y. Kajitani, "An improvement of network-flow based multi-way circuit partitioning algorithm," IEICE Trans. Fundamentals, vol.E85-A, no.3, pp.655–663, March 2002.

[9] E. Bozorgzadeh, S. Ghiasi, A. Takahashi, and M. Sarrafzadeh, "Optimal integer delay-budget assignment on directed acyclic graphs," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.23, no.8, pp.1184–1199, 2004.

Masato Inagi received his B.E. degree in computer science and M.E. degree in communications and integrated systems from Tokyo Institute of Technology in 2000 and 2002, respectively. He was a researcher at the Kitakyushu Foundation for the Advancement of Industry, Science and Technology from 2005 to 2007. He was with the University of Kitakyushu as a researcher from 2007 to 2008. Since 2008, he has been a research associate at Hiroshima City University. His research interests include combinatorial algorithms for circuit layout design.

Yasuhiro Takashima received his B.E., M.E., and D.E. degrees in electrical and electronic engineering from the Tokyo Institute of Technology in 1993, 1995, and 1998, respectively. He was a research associate in the School of Information Science at the Japan Advanced Institute of Science and Technology from 1998 to 2003. He was with the Kitakyushu Foundation for the Advancement of Industry, Science and Technology as an invited researcher from 2003 to 2005. Since 2005, he has been an associate professor of the University of Kitakyushu. His research interests include combinatorial algorithms for VLSI design.

Yuichi Nakamura received a B.E. in information engineering and a M.E. in electrical engineering from the Tokyo Institute of Technology in 1986 and 1988, respectively. He received D.E. degree from the Graduate School of Information, Production, and Systems, Waseda University, in 2007. He is currently a principal researcher at System IP Core Research Laboratories, NEC Corp. His research interests include the system LSI design and their algorithms.

Atsushi Takahashi received his B.E., M.E., and D.E. degrees in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1989, 1991, and 1996, respectively. He was with the Tokyo Institute of Technology as a research associate from 1991 to 1997 and has been an associate professor since 1997. He visited University of California, Los Angeles, U.S.A., as a visiting scholar from 2001 to 2002. He is currently with Department of Communications and Integrated Systems, Graduate School of Science and Engineering, Tokyo Institute of Technology. His research interests are in VLSI layout design and combinational algorithms. He is a member of IEEE and IPSJ.