ICCAD 2019 CAD Contest Problem B

System-level FPGA Routing with Timing Division Multiplexing Technique

Yu-Hsuan Su, Emplus Huang, Hung-Hao Lai, and Yi-Cheng Zhao
Synopsys Taiwan Co., Ltd.,
1F., No. 25, Gongye E. 4th Rd., Hsinchu Science Park, Hsinchu, Taiwan

I. Introduction

As IC process geometry shrinks rapidly, the size of VLSI circuits as well as fabrication costs are also increased. Logic verification is one of the most important methodologies in advanced sub-nanometer technology and beyond, according to ITRS roadmap.

One of the strategies for logic verification is software logic simulation. It provides visibility and debugging capabilities. However, it consumes huge runtime and significant cost since it have to emulate each logic gate one by one and the circuit size is extremely large. Another way to perform logic verification is the hardware emulation. It greatly reduces runtime but the implementation cost is very high. The other approach for logic verification is using FPGA prototyping system. The FPGA prototyping verifies the circuit by a configurable FPGA system. Due to the capacity limitation of one FPGA, a multi-FPGA prototyping system is developed to verify a large circuit design. It is faster and cheaper. Thus, the FPGA prototyping system is widely used in industry.

To adapt a design to the FPGA prototyping system, a large VLSI circuit must be partitioned into sub-circuits and each of them fits a single FPGA. Since the number of I/O pins in an FPGA is fixed and limited, the routing signals can usually exceed the number of I/O pins. A time-multiplexing division [Babb, et al.] is required to transfer multiple I/O signals by time division technique.

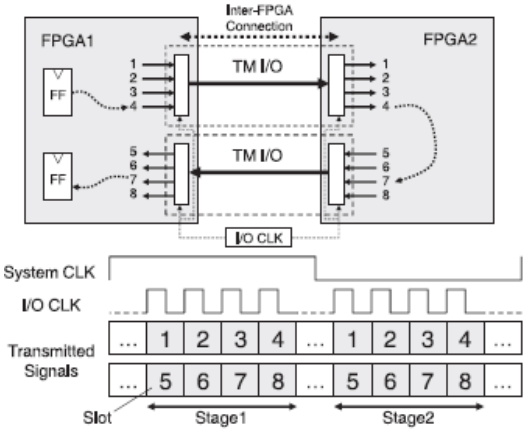

Fig. 1. The time multiplexing division (TDM) technique. Total eight routing signals can be transmitted in one system clock period. The technique increases the signal capability in one FPGA and the high routability for the prototyping system. However, this technique also slows down the inter-FPGA signal delay. [Babb, et al., "Virtual wires: Overcoming pin limitations in FPGA-based logic emulators", IEEE FCCM 1993]

Fig. 1 shows the time multiplexing division technique (TDM). Without this technique, only two routing signals can be transmitted through one FPGA in one system clock period. With the time multiplexing technique, eight routing signals can be transmitted in one system clock period. The technique dramatically increases routing capability in the FPGA prototyping system. However, this technique also slows down the inter-FPGA signal delay which is increased by the time multiplexing rate. For example, the time multiplexing rate in Fig. 1 is eight.

The challenges for system-level FPGA routing lies in the side-effect of TDM. For example, with TDM, routing can always complete. But the inter-FPGA signal delay is increased by time-multiplexing of I/O pins [M. Inagi, et al., 2010].

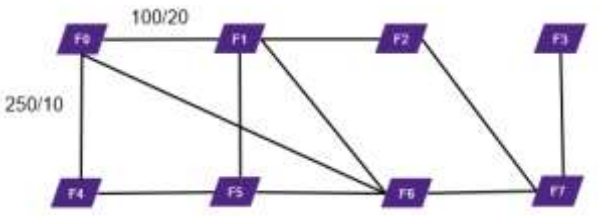

Fig. 2. We can model this system-level FPGA routing problem to a graph. Each FPGA connection can be modeled as one graph edge and each signal belongs to one or multiple net groups. Please kindly notice that TDM ratio is defined as an even number due to multiplexing hardware implementation. For example, TDM ratio = 26 for 250/10 but not 25 for the F0 and F4 connection.

Intuitively we can model this system-level FPGA routing problem to a routing graph as shown in Fig. 2. Each FPGA connection can be modeled as one graph edge and each signal belongs to one or multiple net groups. We can also define the capacity of each edge as the number of pins, and demand of each edge as the number of signals. Please kindly notice that TDM ratio is defined as an even number due to multiplexing hardware implementation. Each signal in an edge should have its TDM ratio and the TDM ratio is defined as an even number. For example, TDM ratio = 26 for 250/10 but not 25 for the connection between F0 and F4 in Fig. 2.

In addition, we will define a net group including several nets in one group due to design purpose. For example, nets belongs to similar attributes or same power consumption would be in the same net group.

There are mainly two approaches for the system-level FPGA routing system. One is to model as an optimization problem and solved by ILP [M. Inagi, et al.] and the other is a heuristic method by modelling as a routing graph [M. Turki, et al.].

II. Problem Formulation

Given a netlist (with two-pin and/or multi-pin nets), a multi-FPGA system with timing division multiplexing (TDM) channels between each FPGA connection pair, and net groups, route the nets and assign TDM ratio for each channel, with minimized maximum total TDM ratio for each net group and minimized run time. Each net could belong to several net groups. The output is the routing path of each net and the TDM ratio along this routing path. The objective is to minimize the maximum total TDM ratio of each net group and runtime simultaneously.

Each routing signal should have its TDM ratio. The TDM ratio is an even number and at least two. The TDM ratios of all routing signals on a routing edge should satisfy $1 >= (1/2) * r_2 + (1/4) * r_4 + (1/6) * r_6 ... + (1/k) * r_k$, where r_i is #signal using TDM ratio i, $i=1\sim k$.

III. Input Format

There are mainly four parts in the input format.

The first part describes total number of FPGAs, total number of FPGA connections,

total number of nets, and total number of nets including two-pin nets and multi-pin nets, and total number of net groups.

Format: **<Total number of FPGA> <Total number of FPGA connections> <Total number of nets> <Total number of net groups>**

The second part is FPGA connections. The number of lines in the second part is the number of FPGA connection.

Format: **<FPGA id> <FPGA id>**

The third part is the net description. The first label is the source and the remaining labels describing the targets. The number of lines for the third part is equal to the total number of nets.

Format: **<FPGA source> <FPGA target 1> <FPGA target 2>…**

The final part is the net groups describing net ids in one net group. The number of lines for the fourth part is equal to the total number of net groups.

Format: **<Net id> <Net id>…**

More precisely, the first line of input file contains four positive integers N_f $(1 \leq N_f \leq 500)$, N_e $(1 \leq N_e \leq N_f(N_f-1)/2)$, N_w $(1 \leq N_w \leq 106)$, and N_g $(1 \leq N_g \leq 106)$ separated by space. N_f is the total number of FPGA, N_e is the total number of FPGA connections or graph edges, N_w is the total number of nets, N_g is the total number of net groups.

Next we have N_e lines represent N_e graph edges. The line number indicates edge id starting from zero to N_e-1. The i-th line contains two positive integers j and k $(0 \leq j < k < N_f)$ separated by space represents the edge id i $(0 \leq i < N_e)$ connects FPGA F_j and F_k. We guarantee that FPGA graph is a connected graph.

And then we have N_s lines represent N_s routing nets. The line number indicates net id starting from zero to N_s-1. The m-th line includes positive integers s and t_k $(0 \leq t_k < N_f)$ separated by spaces represents the source of the net id m $(0 \leq m < N_s)$ is s and the target is t_k.

Finally we have N_g lines represent N_g net groups. The line number indicates net group id starting from zero to N_g-1. The n-th line includes integers g_m $(0 \leq g_m < N_e)$ separated by spaces represents that there are $|g_m|$ nets for net group id n. We guarantee that all nets will be in one group.

Sample input of Fig. 2 is below:

```
8 11 5 3
0 1
0 4
0 6
1 2
1 5
1 6
2 7
3 7
4 5
5 6
6 7
0 1
1 5
5 6
0 4 5 6
5 7
0 1 2
3
4
```

IV. Output Format

Output the data in the order of the net id referred to the input format.

For each net, the first line prints out the total number of edges for the net. Then print out n_e lines for total n_e edges of this net. There are two numbers for each line. One is the net id and the other is the TDM ratio for this net on this routing path.

For example, below is the format of a routing path using two routing edges:
<total number of routing edge>
<edge id> <TDM ratio>
<edge id> <TDM ratio>

Sample output for Fig. 2:

```
1
0 2
1
4 2
1
9 2
3
1 2
8 2
2 2
1
9 4
```

V. Evaluation Methodology

To emphasize on the runtime impact for large designs, runtime is considered in the evaluation score.

Evaluation score = alpha * runtime + beta * maximum total TDM ratio of all net groups, where alpha = beta = 0.5.

VI. References

1. M. Inagi, Y. Takashima, and Y. Nakamura, “Globally optimal time-multiplexing of inter-FPGA connections for multi-FPGA prototyping systems.” TSLM, pp. 388-397, 2010
2. M. Inagi, Y, Takashima, Y. Nakamura, and A. Takahashi, “ILP-based optimization of time-multiplxed I/O assignment for multi-FPGA system.” IEEE 2008
3. M. Turki, Z. Marrakachi, H.Mehrez, and M. Abid, “Signal multiplexing approach to improve inter-FPGA bandwidth of prototyping platform.” DATS, 2015