

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

Contents

0. Announcement.....	P2
I. Introduction	P3
II. Problem Formulation	P5
III. Input Format	P5
IV. Output Format	P8
V. Evaluation Methodology	P8
VI. Submission Detail	P9
VII. References	P10
VIII. Alpha Report.....	P11
IX. Beta Report.....	P11
X. FAQ.....	P12

0. Announcement

October

- 2019-10--

September

- 2019-09-06- The FAQ of ProblemB is updated.
- 2019-09-03- The Beta Report and the FAQ of ProblemB are updated.

August

- 2019-08-13- The Beta Report and the FAQ of ProblemB are updated.

July

- 2019-07-19- The FAQ of ProblemB is updated.
- 2019-07-12- The evaluatorFast of ProblemB is updated.
- 2019-07-12- The Alpha Report of ProblemB is updated.
- 2019-07-12- ProblemB description and FAQ are updated.
- 2019-07-11- The FAQ of ProblemB is updated.
- 2019-07-09- The Alpha Report of ProblemB is updated.

June

- 2019-06-26- ProblemB description and FAQ are updated.
- 2019-06-04- ProblemB description and FAQ are updated.

May

- 2019-05-31- The FAQ of ProblemB is updated.

April

- 2019-04-30- The FAQ of ProblemB is updated.
- 2019-04-23- The FAQ of ProblemB is updated.
- 2019-04-23- The evaluator of ProblemB is updated.
- 2019-04-15- The FAQ of ProblemB is updated.
- 2019-04-12- The test cases of ProblemB is updated.
- 2019-04-11- The evaluator of ProblemB is updated.

March

- 2019-03-29- ProblemB description and FAQ are updated.
- 2019-03-27- ProblemB description and FAQ are updated.
- 2019-03-19- ProblemB is updated.

February

- 2019-02-19- ProblemB is updated.
- 2019-02-13- ProblemB is description updated.
- 2019-02-11- ProblemB is description updated.
- 2019-02-01- ProblemB announced.

ICCAD 2019 CAD Contest Problem B

Yu-Hsuan Su, Chao-Yuan Huang, 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.

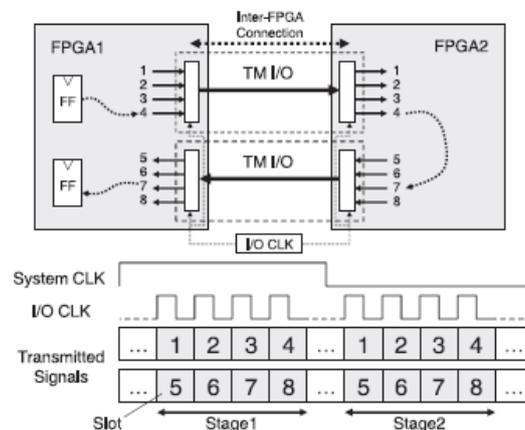


Fig. 1. The time multiplexing division (TDM) technique. Total eight routing signals can be transmitted in one system clock period. The technique increases the signal capability in one FPGA and the high routability for the prototyping system. However, this technique also slows down the inter-FPGA signal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

Here introduce the time multiplexing division technique (TDM) as shown in Fig. 1. 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].

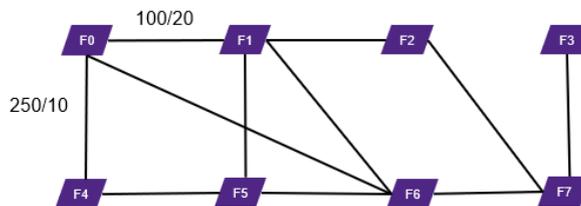


Fig. 2. We can model this system-level FPGA routing problem to a graph. Each FPGA connection can be modeled as one graph edge and each signal belongs to one or multiple net groups. Please kindly notice that TDM ratio is defined as an even number due to multiplexing hardware implementation. For example,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

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. Suppose there are total 250 wires

routing through the inter-FPGA connection F0 and F4, and there are total 10 I/O pins available in this inter-FPGA connection.

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

Kindly notice that to simply the problem, we assume that the capacity of each edge is one. The problem asks for assigning TDM ratio to each specific net and the TDM values on one edge should satisfy TDM constraint as shown in Section II.

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 net, 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. That is, the smallest TDM ratio is two. In addition, the TDM ratios of all routing signals on a routing edge (on an inter-FPGA connection) should satisfy TDM constraint: $1 \geq (1/2) * r_2 + (1/4) * r_4 + (1/6) * r_6 \dots + (1/k) * r_k$, where r_i is #signal using TDM ratio i , $i > 0$ and i is a multiple of two.

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 ($1 \leq N_w \leq 1000000$), and N_g ($1 \leq N_g \leq 1000000$) 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 .

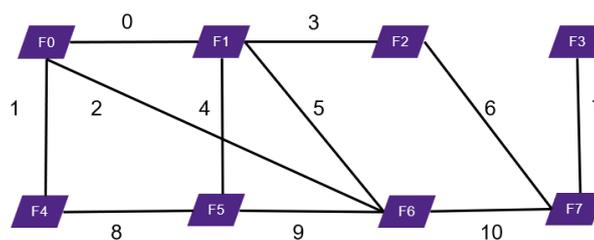
Finally we have N_g lines represent N_g net groups. The line number indicates net group id starting from zero to N_g-1 . The n -th line includes integers g_m ($0 \leq g_m < N_e$) separated by spaces represents that there are $|g_m|$ nets for net group id n . We guarantee that all nets will be in one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

Sample input of the inter-FPGA connections in 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

```



Net id	Source FPGA id	Target FPGA id
0	0	1
1	1	5
2	5	6
3	0	4, 5, 6
4	5	7

Fig. 3. The five nets and the corresponding FPGA source and target ids for the five nets from sample input. The number labelled on the edge is the edge id.

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>

A feasible solution for the sample input:

There could be several TDM ratios on a graph edge for different nets. For example, the TDM ratios on edge id 9 satisfy constraint because $1 \geq (1/2) * 1 + (1/4) * 2$.

1
0 2
1
4 2
1
9 2
3
1 2
8 2
9 4
2
9 4
10 2

V. Evaluation Methodology

To emphasize on the runtime impact for large designs, runtime is considered in the evaluation score. The evaluation score of all test cases is defined by runtime and the maximum total TDM ratios of all net groups. In addition, we define the hard time

limit as four hours. Each run will be killed after four hours execution.

The evaluation score is defined below:

$(\text{Maximum total TDM ratio of all net groups}) * (1 + \log_2(x_i / X) * 0.01)$,

where x_i is the runtime of team i and X is the medium of runtime for all contestants.

For example, total TDM ratios of net 0, 1, 2, 3, and 4 are 2, 2, 2, 8, and 6 respectively. So total TDM ratios of net group 0, 1, and 2 are 6, 8, and 6 respectively. The maximum total TDM ratio of all net groups is the maximum of 6, 8, and 6 which is equal to 8.

In addition, the usage of evaluator is:

```
./evaluator <test case file> <your output file>
```

The evaluator will dump below messages if your output result satisfies output format and TDM constraint.

```
start read FPGA graph file
read success!
start read user output file
read success!
start check TDM ratio and net connection
      No error, your score is 22
finish all check, program terminate.
```

For example, we have the sample input (SampleInput) and output (SampleOutput) for the sample test case.

```
./evaluator <SampleInput> <SampleOutput>
```

We will prepare two versions of evaluator. One is the original version with the precise TDM calculation. The other is a new version with faster runtime but the precision may not good as the original version. Contestants can choose the version they use. But the final evaluation will depend on the original version due to high quality of precision.

Also, kindly remind you that if there the TDM ratio is large after the reduction of a fraction, there will be long runtime for evaluator.

VI. Submission Detail:

Please follow the executive format: `./iccad2019B <input path> <output path>`

iccad2019B is your binary, <input path> is the input file path, and <output path> is

the output file path.

There are two files you should submit for problem B. One is the binary and the other is the README file.

The binary naming should include the id and the name for your team.

Also, you should provide a README file to describe the binary execution and how to generate your output file.

VII. 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-multiplexed 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

VIII. Alpha Report

rank	synopsis#01		synopsis#02		synopsis#03		synopsis#04		synopsis#05		synopsis#06		error message
	TMD ratio	runtime	TMD ratio	runtime	TMD ratio	runtime	TMD ratio	runtime	TMD ratio	runtime	TMD ratio	runtime	
1	46880	0m19.80s	33183296	1m47.501s	144243078	31m30.491s	7516180	58m55.066s	5499942	128m19.735s	16040327668	226m16.566s	
2	48708	0m49.189s	64465104	0m47.144s	689740924	9m25.184s	29376332	32m3.651s	6305502	93m46.254s	40888765796	93m30.468s	
3	51748	0m41.216s	177923052	1m3.497s	2565451810	6m35.627s	91892380	32m5.813s	10514886	107m39.059s	126542369930	131m34.336s	
4	1111222	0m1.117s	221291422	0m3.446s	3404391322	0m41.742s	168421326	1m20.111s	23511320	2m28.628s	165336691420	9m17.821s	
5	53778	0m3.715s	220627410	0m5.936s	3423757318	1m5.247s	108738484	3m27.729s	22422356	7m57.549s	165177650238	13m31.563s	
6	53778	0m5.702s	220627410	0m12.181s	3423757318	2m06.869s	108738484	6m35.258s	22422356	13m20.174s	165177650238	0m53.058s	
7	1111222	0m2.068s	221291422	0m11.785s	3404391322	2m14.192s	168421326	4m34.274s	23511320	7m27.722s	165336691420	26m29.748s	

IX. Beta Report

Rank	case1			case2			case3			case4			case5			case6			hidden1		hidden2			
	TMD ratio	runtime(s)	score	TMD ratio	runtime(s)	score	TMD ratio	runtime(s)	score	TMD ratio	runtime(s)	score	TMD ratio	runtime(s)	score	TMD ratio	runtime(s)	score	TMD ratio	runtime(s)	score	TMD ratio	runtime(s)	
1	42146	2.985	41910.49	31864716	8.929	31684591.66	129454080	59.375	127559774.5	6915358	209.022	6849205.053	5136582	422.784	5078162	15756161338.00	612.943	15355473605.00	409800270	12.078	400423274.9	45945897000.00	281.005	44948346673.15
2	42782	1.778	42223.15	32168962	3.939	31607308.8	129173202	59.43	127284732	7070992	155.698	6973305.289	5184992	340.39	5109806	15762643270.00	646.087	15373766449.48	409891642	23.966	404564775.9	45953296186.00	493.7	45329203360.52
3	38152	29.533	39200.32	31690158	134.868	32752295.99	128557096	1765.266	132967354	6435198	6096.914	6686797.726	4701952	8085.152	4848650	15795939934.00	5165.951	15879998797.68	408872518	272.982	417909278.1	45964769822.00	1801.034	46198738062.11
4	40460	4.056	40412.88	33144546	11.199	33065503.23	133886290	163.72	133886290	7168366	397.529	7166272.563	5291976	926.49	5291686	15992165418.00	4966.638	16068190654.57	413444398	58.648	413409311.8	46263845820.00	1440.277	46350145166.34
5	45518	5.72	45690.74	34138458	12.345	34105028.93	149410572	158.593	149341990.3	10819858	405.658	10819858	8304686	888.153	8299167	15813152724.00	7601.889	15985433681.72	424810276	58.994	424810276	46066926288.00	2328.653	46472170915.96

X. FAQ

Q1. The descriptions under the picture Fig.1 and Fig.2 in page2 is almost the same.

A1. There should be two corresponding descriptions for Fig. 1 and Fig. 2

Q2. How to set the TDM ratio in the output file? Is each group of nets set from 2 if not used?

Because there seems to be no mention how to set the TDM ration in the whole document, only mention the TDM ratio=26, 250/10. What is 250 and 10 ? and there is no similar number in the Input file.

A2. Please refer to the problem formulation: The TDM ratio is an even number and at least two.

I have updated in the introduction: Suppose there are total 250 wires routing through the inter-FPGA connection F0 and F4, and there are total 10 I/O pins available in this inter-FPGA connection.

Q3. The TDM ratios of all " routing signals "on a routing edge should satisfy $1 \geq (1/2) * r_2 + (1/4) * r_4 + (1/6) * r_6 \dots + (1/k) * r_k$, where r_i is #signal using TDM ratio i , $i=1 \sim k$.

Does "routing signal" mean a net group?

Because the TDM ratio of edge 9 has two values in the output file given by the example, I think the above description means each net groups can only have own TDM ratio.

A3. Routing signals mean the routing wires.

There could be several TDM ratios on a graph edge if it satisfies $1 \geq (1/2) * r_2 + (1/4) * r_4 + (1/6) * r_6 \dots + (1/k) * r_k$. But there is a typo in the value so I have updated the sample output. The TDM ratio of edge id 9 is 4 because it satisfies $1 \geq 1/4 * 3$. In addition, there are three wires route through edge id 9 which are net ids 2, 3, and 4.

Q4. Last paragraph in the example output file

1

9 4

Is there a wrong answer here?

Because this net is routing from FPGA id = 5 to 7, but edge 9 is only connected from 5 to 6, so we need to route extra edge 10 to connect from 5 to 7.

A4. The sample output has been modified as

2

9 4

10 2

Because there is only one routing wire routes through edge id 10. So TDM ratio of net 4 on edge id 10 is two.

Q5. Follow problem 4, how to set TDM ratio = 4 of edge 9? Because the previous net group 1 (0 1 2) has used edge 9 (TDM ratio = 2), so the TDM ratio = 4 here ?

A5. Yes, there is a wrong output in the current version. We have updated “9 2” to “9 4” for net 3 to satisfy $1 \geq (1/2) * 1 + (1/4) * 2$.

Q6. Is the FPGA I/O pin is fixed to 1 in this problem?

A6. To simply the problem, we abstract the multiple FPGA I/O pin hardware architecture as just one graph edge between two FPGAs and define the #pin as the edge capacity. For example, there are 100 FPGA pins for an FPGA interconnection. Then the capacity of this graph edge is 100.

Q7. What is the function of the net group? Is used to calculate the Evaluation score? How to determine the maximum total TDM ratio of all net groups? Could you give some example?

A7. Yes, it is used to calculate the evaluation score. For example, the total TDM ratio of net group 1, 2, and 3 are 100, 24, and 50. Then the maximum total TDM ratio of the design is 100.

Q8. Does each net can't have multiple source? Like the 4th net group, I want to output '6 2' rather than '9 2'. And this output seems better than previous one.

A8. Thanks for your question. Yes, each net is single source.

Q9. How is the output '8 2' generated? Does that mean the signal has been transferred to F4 and now it can directly be transferred from F4 to F5?

A9. The source of net id 3 is F0 and targets are F4, F5, and F6. So it could route through edge 1, 8, 9 and route through F4, F5, and F6.

Q10. Why the TDM ratio = 4 of edge 9 at the end of the output while it is 2 for the 3rd net? I think the ratio of edge 9 should all be 9. Is there something wrong here?

A10. Thanks for your observation. We have updated “9 2” to “9 4” to satisfy the TDM constraint equation (Equation 1 in problem formulation).

Q11. How to set TDM ratio =2 of edge 9? The net id 2, 3, and 4 are all through the edge which edge id is 9, but why their TDM ratio is not the same.

A11. There could be many TDM ratio values in one edge (one FPGA connection). But the ratio should satisfy TDM constraint equation (Equation 1 in problem formulation). So there is one net with TDM ratio = 2 and two nets with TDM ratio = 4 to satisfy $1 \geq (1/2) * 1 + (1/4) * 2$. We have updated the example output.

Q12. In the example output file

3

1 2

8 2

9 2

Is there a wrong answer here?

The signal goes from FPGA 0 to FPGA 5 through the edge 1 and the edge 8, why this net's TDM ratio is not 4.

A12. Yes, there is a wrong output in the current version. We have updated "9 2" to "9 4" for net 3 to satisfy $1 \geq (1/2) * 1 + (1/4) * 2$.

Q13. I have some further question about Q3 in Problem B, FAQ.

It said that "The TDM ratio of edge id 9 is 4 because it satisfies $1 \geq 1/4 * 3$. In addition, there are three wires route through edge id 9 which are net ids 2, 3, and 4." However, in part IV, the sample output had the edge id 9 with 3 TDM ratio 2, 2, 4, respectively.

Therefore, in my point of view, $r_2=2$ and $r_4=1$ referring to this inequality " $1 \geq (1/2) * r_2 + (1/4) * r_4 + (1/6) * r_6 + \dots + (1/k) * r_k$ ".

Here comes my question. Why is the inequality shown in Q3 said the TDM ration in sample output satisfied $1 \geq 1/4 * 3$, not $1/2 * 2 + 1/4 * 1$ (which violate the inequality)? Then, when we calculating total TDM ratio, whether we should use 2 or 4 for edge id 9 in netgroup 1 and 2? That is, total TDM ration for netgroup (0, 1, 2) is either (2+2+2, 2+2+2, 4+2) or (2+2+4, 2+2+4, 4+2)?

Besides, can you please give some other examples which can help us understand this problem more?

A13. Thanks for your question. Please refer to the answer to A12.

Q14. I think there is a misleading inequality specified in the problem description, for edge number 9 in the given example, this equation: $1 \geq (1/2) * 2 + (1/4) * 1$ actually does not hold? Is there any mistake made in the official FAQs?

A14. Thanks for your question. Please refer to the answer to Q12.

Q15. A net may contain several edges with different TDM ratio, how to determine the corresponding TDM ratio during evaluation?

A15. Each routing segment in each graph edge has different TDM ratio. The TDM ratio for each net in each graph edge has to be recorded in the output file. (Please refer to the output format section.) So the evaluator add TDM ratio of each graph edge of each net for each net group. The objective is to minimize the max. total TDM ratio of for all net groups.

Q16. Is direction considered?

A16. The graph edge is unidirectional.

Q17. All the edges in the FPGA connection graph have capacity equals 1?

A17. Yes. It is a simplified model to have TDM constraint.

Q18. What is the size range of the benchmarks in terms of nets, FPGA numbers, connection graph size?

A18. Please refer to the input format section.

1. Size range of nets: Total number of nets

2. Size range of FPGA numbers: Total number of FPGA

3. Size range of connection graph size: it will be a sparse connection graph dependent on the graph nodes and edges.

Q19. I have some problems about the problem B. That is, is the net must obey the sequence given by the input?

For instance, the net '0 => 4, 5, 6'. Can I route the net from '0 => 6 => 5 => 4'?

A19. Yes.

Q20. Can we use **multiple thread** in the contest?

A20. We will ask the server team in the contest and update the problem B.

Q21. Why the Q5 say it satisfies $1 \geq 1/4 * 3$, and Q12 say it satisfy $1 \geq (1/2) * 1 + (1/4) * 2$.

There are two different methods to satisfy.

And why there say $r_i \% 2 = 0$ at the end of Problem Formulation?

r_i isn't the number of the routing wires using TDM ratio i on each routing edge.

Besides, it is a contradiction with Q12 answer.

A21.

1. A5 is updated accordingly.
2. Yes, the latest version has updated accordingly. The TDM ratio i is an even number and ri is #number of the routing wires using TDM ratio i on each routing edge.

Q22. Regarding Q18, I understand we can read those information from the input file.

However, I wonder if it's possible to let us know how large the design will be, for example, at most around 1M nets, etc.

A22. Please refer to the problem introduction Section III Input format:

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 ($1 \leq N_w \leq 10000000$), and N_g ($1 \leq N_g \leq 10000000$) 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.

Q23. Could you tell me what is the definition of runtime in Section V.

Evaluation Methodology?

According to the evaluator, sample input and sample output you gave, I got the Evaluation score is 8.

Base on the formula in Section V.

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

$8 = 0.5 * \text{runtime} + 0.5 * 4$, why the runtime is 12 ?

I want to know the relationship between runtime and output.

A23. Runtime is the final user time in second.

The evaluator only calculates maximum total TDM ratio of all net groups. So the evaluation score may depend on the machine you use in your local terminal.

We have modified the output information of evaluator:

```
start read FPGA graph file
read success!
start read user output file
read success!
start check TDM ratio and net connection
      No error, maximum total TDM ratio of all net groups is: 8
finish all check, program terminate.
```

Q24. About the FAQ6 answer.

Does it mean there may well be given two edges between F0 and F1 in the input, then the I/O pins between F0 and F1 will be 2?

If not, where can I get the I/O pins information?

A24. There will be only one edge between each pair of FPGA. The problem assumes that the capacity of each graph edge is one. We simplify the real hardware architecture to consider only one pin pair in one pair of FPGA and thus ignore the I/O pin information.

Q25. Can we use Boost C++ Libraries?

A25. We will double check if there is Boost in the server. Also, we will suggest to use static link.

Q26. Is the routing edge in a net continuous? For example, in net '3', can we route edge '1 8' and edge '2' ?

A26. Yes. In net '3', you can route edge '1', '8' and edge '2'.

Q27. Is the definition of TDM ratio, total wires / (I/O pins), just a background for this problem? Does it have any relationship with solving the problem?

A27. We simplify the #I/O pin of each pair of FPGA is one so the capacity of each graph edge is one.

Please refer to the problem formulation.

Q28. Will the released-testcases be used to evaluate our final version program, or you will only use all hidden-testcases?

A28. We have six hidden test cases. The case size will be almost the same as the published case.

Q29. It seems that the evaluator_V2 does not work correctly when checking the netid of a netgroup. I got the error message "start read FPGA graph file Error! Number out of range." when I apply all the released testcases (synopsys01-06). If I change the netid to the number which is less than total # FPGAs, the checker can work correctly. I think the evaluator_v2 need to check the netid less than total # FPGA connections not total # FPGAs, right?

A29. We have updated the new evaluator binary. Please try again and let us know if there are still issues.

Q30. In the document there is a statement says "We can also define the capacity of each edge as the number of pins, and demand of each edge as the number of signals." The problem assumes that the capacity of each edge is one, and the problem asks for assigning TDM ratio to each edge with specific net. Does it mean that we are

assigning the demand of each edge, namely the number of wires of each edge because the number of I/O pin is fixed to 1?

For example, consider the edge between F0 and F4. If I assign TDM ratio 26 for the edge, does it mean that I assign 26 (the total number of wires) between F0 and F4 because the number of I/O pin is fixed to 1?

A30. No. The main difficulty of the problem lies in the TDM constraint satisfiability. The problem assumes the capacity of each edge is one and asks to assign TDM values for each net on the edge along the routing path.

For example, if there are three nets on one edge and the TDM values are 2, 4, and 4, respectively. Then it satisfies TDM constraint because $1/2+1/4+1/4 = 8/8 \leq 1$.

Q31. Is c++11 supported on the servers? There are two scenarios:

(1) When the program is compiled with "-std=c++11 -static" flag on our local computers, will the program run smoothly on your servers?

(2) When we try to compile the program with "-std=c++11" flag on CIC servers, will the program compile? Or are the kernels on your servers too old for c++11?

A31. You can compile with an CentOS environment with c++11 then submit the binary.

Q32. It seems that the evaluator supports multiple source of one net. I mean that input `0 4 5 6` can output

`

1 2

2 2

9 4

`

Here, F0 has 2 outputs.

A32. Yes, F0 is source and it may have more than one output.

Q33. What does the unidirectional graph mean?

Can one edge be routed in two different directions?

For example, one net routes `F0 => F1` while another routes `F0 => F1`.

A33. One edge can be routed in two different directions. But kindly notice that the TDM is calculated together for the two directions.

Q34. why your official input can't be read correctly?

```
$ ./evaluator_V2 synopsys01.txt synopsys01.out
```

```
start read FPGA graph file
```

```
Error! Number out of range.
```

A34. We have updated a new evaluator "evaluator_v3".

Q35. Evaluation score = $\alpha * \text{runtime in sec} + \beta * \text{maximum total TDM ratio of all net groups}$, where $\alpha = \beta = 0.5$.

Do alpha and beta are always equal to 0.5 for all test case?

A35. We will determine the final values soon.

Q36. Based on FAQ A16, "The graph edge is unidirectional."

Is there any guarantee that the testcases are solvable?

For example, consider the sample FPGA instance as shown in Fig. 2 (Fig. 3), if there are two nets, (F6, F3) and (F3, F7), conflicts occur in the graph edge between F3 and F7.

A36. There will always can find feasible solution because there could be many nets in an edge. A feasible solution is a solution when the TDM ratio satisfies TDM constraints.

Q37. What is the command to invoke our program?

Is it

```
./our_program <input file> <output file>
```

Please kindly let me know if there are any updates on the question concerning c++11, too.

A37. The format is correct.

Q38. Will there be a hard time limit for each testcase?

A38. Yes, the hard time limit is four hour for each test case and it is updated in the evaluation methodology section.

Q39. The evaluation score, $\alpha * \text{runtime in sec} + \beta * \text{maximum total TDM ratio of all net groups}$, where $\alpha = \beta = 0.5$. Will $\alpha = \beta = 0.5$ hold true for all testcases? This question has be raised in the QA section without a clear answer for the time being.

A39. We have defined the new evaluation score and updated in the evaluation methodology section.

Q40. Are we allowed to use multi-thread computing?

A40. Yes, it is allowed to use multi-thread computing.

Q41. In input net '3' is " 0 4 5 6 ".

The source is F0 and the target is F4 F5 F6.

The output net '3' is

3

1 2

8 2

9 2

Does it mean that should route F0 to F4, F0 to F5, and F0 to F6?

But in routing F0 to F4 we route edge'1' .

F0 to F5 we route edge'1' and edge'8'.

F0 to F6 we route edge'1' edge'8' edge'9'.

Edge'1' is routed 3 times.

So, in the TDM constraint equation.

Should I output TDM ratio = 4 to satisfy $1 \geq (1/4) * 3$?

For example:

3

1 4

8 2

9 2

A41. The edge '1' would be calculated only once because you could find a sub-tree to connect all sources and targets.

Q42. The problem B document does not specify which files should we submit and how to do the naming. Therefore, we just need to submit a binary executable file (without naming restrictions?) and a README file?

A42. The submitted binary execution method is: `"/your_binary <inputFile> <yourOutFile>"` just like the usage of evaluator.

There are two files you should submit for problem B. One is the binary and the other is the README file.

The binary naming should include the id and the name for your team.

Also, you should provide a README file to describe the binary execution and how to generate your output file.

Q43. Are there any instructions on how we should submit the binary?

A43. Please refer to Q43.

Q44. We are working on CAD problem B and what we discover is that when the evaluator checks TDM ratio constraint , it may reduce fractions of all the ratios' reciprocal to a common denominator, which will end up a extremely large denominator and numerator.

We have do some experiments on the evaluator : when we assign same TDM ratio to all the nets on any single edge, the evaluator will run extremely fast.

However, if we start to modify some of the ratios and run the evaluator ,it will not end (ten times slower in case 1 and exceed 6 hours in the rest cases).

That is to say, we know the evaluator will check the constraint very precisely, but we cannot see the desired output(way too slow.....).

Is there any solution?

A44. We will prepare two versions of evaluator. One is the original version with the precise TDM calculation. The other is a new version with faster runtime but the precision may not good as the original version. Contestants can choose the version they use. But the final evaluation will depend on the original version due to high quality of precision.

Also, kindly remind you that if the TDM ratio is large after the reduction of a fraction, there will be long runtime for evaluator.

We will update this information in the web.

Q45. We are working on problem B and is it legal to import matlab function or library into our code?

A45. It is legacy if you can run the binary on the TSRI machine.

Q46. As Q40 mentioned, How many threads we can use?

A46. We are asking the organizer and will response to you soon.

Q47. I have found the alpha submission report. Sorry for any inconvenience. But I also want to know if there are some hidden testcases or not. Thank you for your help.

A47. There is no hidden case in alpha test.

Q49. I want to know if there are any hidden testcases in final test or not. And I also want to know if the rank of the final test will base on the average score of all testcases or not.

A49. There would be hidden test cases in the final test.

The rank would be based on the average score of all test cases.

ps. According to the rules, the final grade is only according to the final test. The alpha test and the beta test are just for helping contestants to verify the correctness and performances of their programs.

Q50. I have a question about the final average score in the alpha report of the problem B.

In the alpha report, the final average score is the average of scores of all cases. But apparently the score of the case 6 dominates, which means the performance of other cases don't matter much. Here I want to know whether the final evaluation is in the same method (average of all cases' scores, instead of average of all cases' ranks)?

By the way, when do you provide a new evaluator? The old evaluator is too slow to check the performance.

A50. Yes, you are right. But we found the Top 5 ranks of each test case have high correlation with the average scores.

That is, the rank of 1 to 5 of each test case is consistent with the average scores of all test cases.

Q51. How many threads we can use...

A51. 8 threads.

Q52. Is there will be any new case in the beta test? When we can get the case if there is.

A52. There will be hidden test cases in beta tests.

Q53. According to the reply of Q49, the final score and ranking is based on average score of all test cases. However, the benchmarks provided so far have different scale of FPGA system and setlist. Therefore, the final ranking will be highly dependent on "big cases". It means that even we have better performance on synopsys01-05, we will get lower ranking if we screw up on synopsys06. I want to ask different test cases will be evaluated equally without weighting? This is very important because it will basically alter our tactic.

A53. Generally we will rank teams by the average score for all cases because the large test cases can differentiate the efficiency and effective for the algorithm. In our observations based on the alpha and beta evaluations, top five teams can compute good results for all test cases.

Q54. Is it possible to release the source code of the evaluators?

We would like to know how the evaluators check our solutions so that we can

- (1) eliminate any possible mismatches
- (2) try to expedite the evaluation process.

A54. The evaluator has released in the link:

https://github.com/jacky860226/ICCAD2019_B_evaluator?fbclid=IwAR1sRmep7HMS0leRv_GZdXIRJ36oyEu3v2B5_yfc4BA3Pp9YMg2iDtfyPjA

There are two branches for the evaluator. You can choose what you prefer to use.

Branch 1: master. It is a slow version but accurate.

Branch 2: long_double. It is a faster version but not so accurate.

The contestants can pull request via the link.

Q55. Is there any penalty for using parallel computing?

A55. No. There is no penalty to use parallel computing.

Q56. Is the execution time calculated from cpu time or real time?

A56. Real time. In the beta result, we use CPU time but we should use real time. But the time out condition is correct to use real time. We apology for the mistake. Will update the corrected score asap.

Q57. What is the biggest TDM value among all cases in the final evaluation(including hidden cases)?

The biggest TDM we know so far is synopsys06.txt, about 10^{10} . Will there be even bigger TDM values in the finals?

We are wondering if unsigned long long int will be able to handle all cases in the final evaluation and not have any numerical overflows.

A57. For TDM ratio, the TDM ratio are the same for cada0036 and cada0029 for the beta results among all test cases.

For runtime evaluation, indeed we run one task at a time on one single machine for beta evaluation.

We use the new IP of TSRI machine 140.110.214.87.

But at the same time, organizer server team also released this machine to contestants.

It causes runtime difference because other contestants may access this server when we are doing evaluation.

The server team will force other teams to log out this server when the final version is submitted.

So that there will be only topic chair can use this machine.

Q58. Is there any memory limit? When we tested our program on the CIC machine, we found that it suffer from “bad alloc” when our program occupies more than 3900M memory. So, we want to know what the specific memory limits in the final evaluation environment is?

Q58. There is no specific memory limit or memory constraint. We use the standard server for evaluation.

Q59. Is the final evaluation environment the same as the CIC machine? In other words, if our program can run smoothly on the CIC machine, can it guarantee that our program will also be able to run successfully in the final evaluation environment?

Q59. We use the server IP 140.110.214.87 for beta evaluation.

We will request a new server with the same quality of this 140.110.214.87 for final evaluation.

The reason is that we are requesting a dedicated machine for evaluation.

Q60. How do you get the corrected runtime of ours as well as the median? Did you rerun our program or use the runtime report you generated before?

Do you have the corrected runtime and score of other teams? If so, could you send us the revised table or update the specification on the contest website?

A60. We rerun the program. Then get the median and total score based on the updated results.

The table will be updated by the organizer.

Q61. I have a question about Problem B.

I want to know whether the TDM ratio of each edge should be less than 429496729 (the upper bound of integer) ?

I didn't see any description about the upper bound of TDM ratio in the specification. But in the source codes of evaluator, in the line 44-45 of userOutput.h, the TDM ratio larger than 429496729 is judged as wrong output:

```
if (tdm_ratio > bigN(4294967296))  
    return "TDM ratio: "+err::out_of_range;
```

I think there may be good solutions including some large TDM ratios (larger than 4294967296) in a very large hidden case. In this case, are these solutions invalid? Is it a constrain that all TDM ratios should be less than 4294967296?

Because the deadline is coming, we will appreciate if you can kindly response us as soon as possible.

Thanks for your help.

A61. We simplify the problem to set a very soft constraint 4294967296. The limit of unsign int.