Repeater Insertion Method and its Application to a 300MHz 128-bit 2-way Superscalar Microprocessor

Norman Kojima† Yukiko Parameswar• Christian Klingner• Yukio Ohtaguro‡
Masataka Matsui‡ Shigeaki Iwasa* Tatsuo Teruyama⋆ Takayoshi Shimazawa‡
Hideki Takeda‡ Kouji Hashizume° Haruyuki Tago‡ Masaaki Yamada†

† Design Methodology Dept., System LSI Design Div.
⋆ Microprocessor Design Dept., System LSI Design Div.
∗ RISC Processor Development Dept., Microprocessor Div.
‡ System ULSI Engineering Lab.
Toshiba Corporation Semiconductor Company
580-1, Horikawa-cho, Saiwai-ku
Kawasaki 210-8520, JAPAN
Phone: +81-44-548-2344 FAX: +81-44-548-8306
e-mail: kojima@dad.eec.toshiba.co.jp

• Toshiba America Elec. Components, Inc.
1060 Rincon Circle, San Jose
California 95131-1325, U.S.A.

◦ Advanced LSI Technology Development Dept.
Toshiba Microelectronics Corp.
580-1, Horikawa-cho, Saiwai-ku
Kawasaki 210-8520, JAPAN

Abstract— Interconnect delay is dominant in today's high speed VLSI circuits and there have been various works to resolve it [1],[2],[3]. A repeater insertion tool "RePertory" has been newly developed to solve the interconnect timing problem on a 300MHz 128-bit 2-way Superscalar Microprocessor. Because of its practical simple algorithm, the location of over 5700 repeaters distributed over 5 modules is calculated in 26 minutes on 5 UltraSPARC-II 360MHz in parallel. This paper describes the goals of RePertory for this project, the timing improvement flow with RePertory, the results and the analysis of this repeater insertion method.

I. Introduction

The "Emotion Engine" ("EE") is a 300MHz 128-bit 2-way Superscalar Microprocessor [4],[5] designed for a next generation video game console. The total of 13.5M transistors with a 0.18 μm gate length were placed on a 15.02mm $\times$ 15.04mm die. In the design of such high-speed large scale VLSI circuits, repeater insertion is indispensable to resolve electromigration problems and to improve interconnect delays.

The automatic repeater insertion tool "RePertory" has been developed to resolve the following restrictions of existing commercial repeater insertion tools:

1. Most tools have a proprietary timing engine that controls which nets will have repeaters inserted. Users have only very limited control.
2. The number, location and drive strength of the repeater cells are also decided by the tool and it is impossible to control them.
3. The results do not always meet the requirements (regarding timing, location, etc). This usually results in time-consuming editing by hand for each iteration step in the flow.

Better controllability in repeater insertion was absolutely necessary because in this project the tapeout criteria was that the post-layout netlist passes on a static timing analyzer, the so-called "Golden Timing Analyzer" (GTA). However, the proprietary timing analyzers that are used in the timing-driven repeater insertion with commercial tools show considerable deviations from the results of the GTA for high-speed circuits. Consequently, a net for that the GTA flags a timing violation is impossible to fix with such a repeater insertion tool if its timing engine does not detect the violation. The only way out of this problem would be to fix many nets manually. This process is not only time-consuming, there is also the risk of introducing new errors.

In addition to basic repeater insertion functions, the following features are implemented in RePertory.

1. Controllability of the repeater location (topological and geometrical) and the number of repeaters for the best result.
2. Preservation of module boundaries (repeater insertion in top level block only) to enable successive IPO[1].
3. More control: Repeaters can be forced into certain nets (i.e. most critical nets) and they can be prohibited in others (i.e. clock nets).

II. Goals and Features of RePertory

A. Goals of the development of RePertory

The following requirements were set for RePertory in this project:

[1]IPO: In Place Optimization; Circuit optimization without changing net-topology, but with cell-size adjustment and/or insertion/deletion of buffer(s).

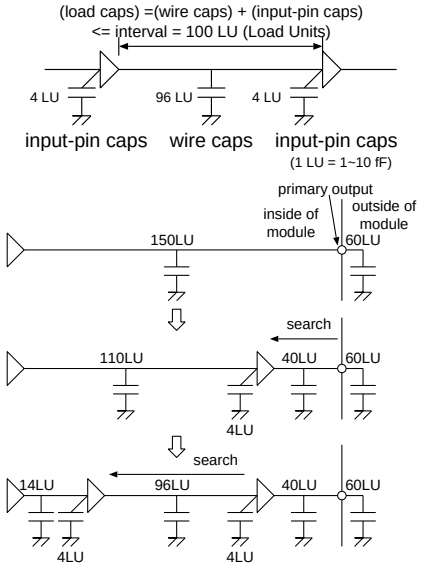

Fig.1. Example of "interval" and repeater insertion by RePertory

1. Short development time
2. Short processing time (run time)
3. Fixing electromigration violations
4. Achieving a clock frequency of 300MHz or more
5. Preservation of module boundaries in order to enable successive IPO

B. Features of RePertory to achieve the goals

Following is a description of the features that were implemented in order to achieve the above goals.

B.1. Practical algorithm for short development and processing time

In order to realize a short development time and short run times as well, RePertory ignores wire resistance and only considers load capacitance (sum of parasitic wire capacitance and gate capacitance) which is described in a DSPF[2] file.

The optimum load capacitance for each size of repeater from a timing point of view with a standard-width no-branch wire was determined in advance with GTA. This load capacitance was called "interval" (See Fig.1 (top)).

Whether or not a net will be subject to repeater insertion is determined by its total capacitance. If it is higher than what is called the "threshold" in this project, repeaters will be inserted. This threshold was calculated from the electromigration constraints (See section B.2).

The following method was applied to every net in the netlist. The capacitance of each net segment[3] is analyzed through recursive search from the source of the signal to the sink(s). If the total capacitance exceeds the threshold, repeaters are inserted beginning from the sink(s) to the source so that in the end none of the new subnets of this net exceeds the interval capacitance (See Fig.1 (bottom)). In most cases a user-specified repeater of medium size is employed by RePertory. If there is a large load beyond the module boundary that this repeater can not drive, a stronger repeater will automatically be selected.

Various parameters of the repeater insertion process such as repeater location, interval, size of repeater cell and timing constraints for each sink are adjustable. This allows users to help the tool fixing timing violations in a few nets where fully automatic repeater insertion did not yield optimum results.

B.2. Prevention of electromigration problems

The electromigration risk becomes more and more severe as currents increase and wire widths shrink. In the EE design, since the wire width of the long inter-connect nets that will have electromigration problems is not changed from the signal source to the sink(s), then the source of each net including the output of the driver of the net is the most critical point of electromigration. From this point of view, appropriate repeater insertion described below avoids electromigration problems.

Based on the design rules and the maximum current density a maximum load capacitance for each net was calculated to guarantee a certain life time. That capacitance value was used as the "threshold" described in section B.1. All nets with a load capacitance higher than the threshold will have repeaters inserted to avoid electromigration problems.

B.3. Timing-driven repeater insertion

Although the results of RePertory met the timing constraints for almost all nets, some of the long and multi-branch nets like the instruction bus signals had remaining timing violations.

In order to speed up those nets, the following parameters are adjustable by users on a net by net basis:

- "threshold" and "interval" (Tuning of those values refine the number of repeaters and the geometrical and topological repeater location. See B.1)
- timing constraints for each sink (so that repeaters can be inserted into the least critical branches to isolate the most critical branches[3])
- size (drive strength) of repeaters

For special nets such as nets with increased width and/or space, clock nets, etc., repeater insertion can be prohibited by the user.

B.4. Preservation of module boundaries

Repeaters will only be inserted into blocks that allow standard cell placement. In other words, hard macro blocks such as memories or datapaths do not allow repeater insertion.

When a repeater is inserted into a net, this net is split up into two nets. The number of nets in this circuit increases and a

[2] DSPF: Detailed Standard Parasitic Format; contains information for each net about geometry, R, C, connections, etc.

[3] A parasitics extraction tool extracts a wire (net) by breaking it up into segments of a certain maximum length. In addition, segments will be broken when this wire bends and/or branches.

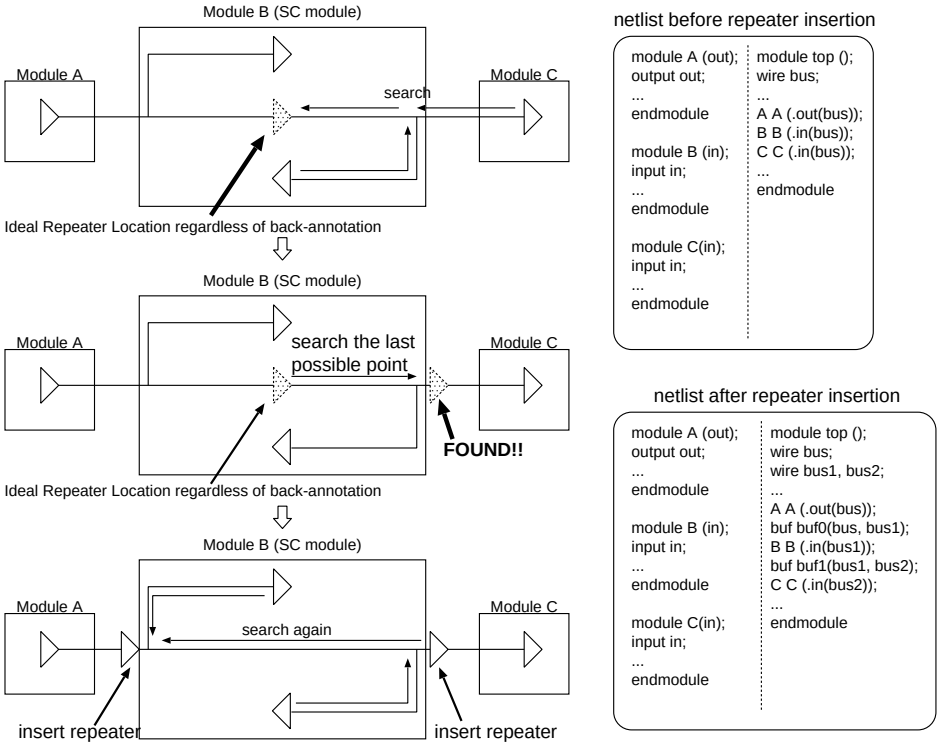

Fig.2. Repeater insertion with module boundary preservation

portion of the old net now exists under a different name. If this net has multiple physical pins on the boundary, some of those pins will keep the old name while others will be connected to the portion of the net that has a different name now. Then, the number of logical pins increases. This is not allowed because IPO is perfomed after repeater insertion on blocks individually outside the context of the full-chip netlist. Therefore RePertory will insert repeaters in the top-level hierarchy only.

Example results for repeater insertion with module boundary preservation are shown in Fig.2. If the boundary of module B is ignored, the ideal location is as shown in the top view and the middle view shows the result of the repeater insertion. The repeater is placed between the two branches in the module B and a new output pin is generated at the boundary of module B. This new output pin will cause a problem in the following IPO. The bottom view of Fig.2 shows the result of the repeater insertion without changing of the module B boundary.

III. Design Flow with RePertory

Fig.3 shows the back-end design flow of the EE , including repeater insertion. Gate-level Verilog-HDL netlists and timing constraints files are synthesized from the RTL. The first timing analysis is executed with pre-layout data to get an early estimate of the performance[6]. For that purpose a Pseudo-DSPF is generated based on the floorplan into that the repeaters will be inserted. The repeaters are also inserted into the netlist which will then together with the modified DSPF be passed on to the timing analyzer. Pre-layout timing analysis with inserted repeaters is possible because RePertory does not depend on a P&R database.

RePertory is invoked again in the main stream of the design flow after the global routing has been executed. Sometimes the repeaters are not placed at their ideal location because that position may already be crowded with other cells. Therefore the global routing information is sufficiently accurate for RePertory and no detailed routing data is needed.

RePertory generates the new Verilog-HDL netlist (with repeaters) and the ECO (Engineering Change Order) placement file. These files are supplied to the P&R tool which then places the repeaters and executes the detailed routing. After that IPO is performed, but the sizes and number of repeaters are preserved during that run and the second P&R run (usually final run) can be started.

Finally, the post-layout data is supplied to the timing analyzer. If the performance is not sufficient, RePertory can be invoked again; but this time with timing constraints for the critical paths.

IV. Applying RePertory to the EE design.

A. Module structure and floorplan of the EE

The top module of the EE contains many submodules (See Fig.5) while four of them (RISC core, Vector Processing Unit (VPU), IPU submodule 1 (IPUsub1), IPU submodule 2 (IPUsub2)) and the top module itself were designed independently, only boundary information was shared. Hence, RePer-

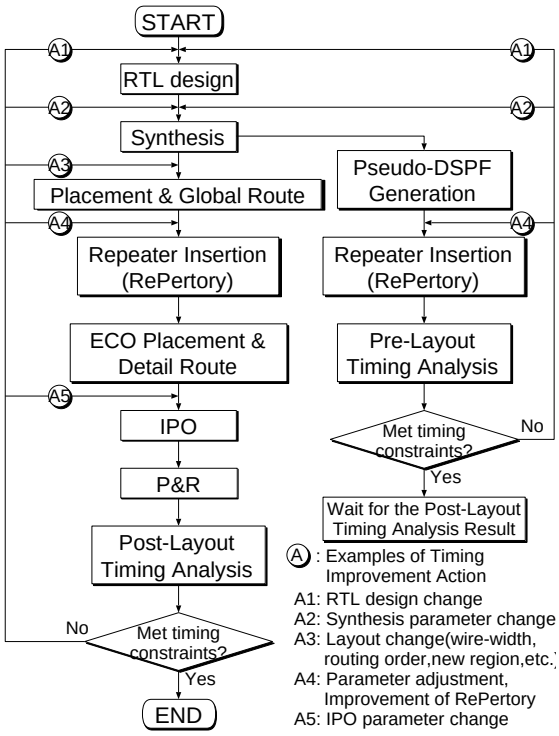

Fig.3. Design flow of EE design with RePertory

tory was applied to those modules independently. Above four submodules were considered hard macroblocks in the context of the top module.

In each of those five modules, there are geometrical rectangular regions allocated for hard macroblocks and standard cell (SC) submodules (soft macroblocks). Each SC region includes one or several SC submodules. All submodules are contained within a single region and not spread across several regions. The size of SC regions is designed smaller than the interval of repeaters so that no long wire net remains inside a SC region.

B. Topological location of repeaters

As section B.4 explains, repeaters are inserted only into inter-region nets. Synthesis and IPO already did a good job on optimizing intra-region nets.

C. Geometrical location of repeaters

Repeaters are allowed to be placed not only inside of SC regions but also in the wiring area. In some cases along critical nets, areas were reserved exclusively for repeaters (See Fig.5). Placing repeaters in the wiring area allows them to be close to their ideal location for minimum wire delays. Power and ground wires for those repeaters were connected after RePertory was done.

Because the wiring area of VPU is limited, "repeater prohibited areas" were placed in the wiring area in order to preserve precious routing resourses. The top module also contains "repeater prohibited areas" in the wiring area in order to avoid too much dispersion of repeaters which causes excessive amounts of power/ground wires.

TABLE I
PROCESSING TIME OF REPERTORY AND NUMBER OF NETS AND REPEATERS

module	# of nets	processing time (min, sec) location calculation	repeater relocation	total	# of repeaters
Top Module	95153	7m53s	4m27s	12m20s	1913
RISC core	59342	4m31s	3m00s	07m31s	2194
VPU	71021	4m21s	21m09s	25m30s	1633
IPUsub1	16514	0m25s	0m00s	00m25s	0
IPUsub2	20705	0m25s	0m01s	00m26s	8

D. Timing improvement flow

The timing improvement team consisted of the engineers in charge of RTL design, synthesis, P&R and RePertory maintenance. The timing analysis results, the route of the nets with repeaters, the RTL description with the critical paths, the repeater insertion parameters employed in the previous run and other data were analyzed carefully by the team members and the following methods were employed to resolve timing violations. During the time before tapeout, the timing improvement flow shown in Fig.3 was repeated 3 to 4 times a month. Almost every time the repeater insertion parameters for critical paths were tuned before the execution of RePertory.

1. Adjustment of the repeater insertion parameters

   Nets for repeaters to be inserted, topological and/or geometrical location of repeaters, timing constraints for each sink, etc. were adjusted and repeater insertion was executed again. Most of timing violations were eliminated in this phase (A4 in Fig.3).

2. Improvement of RePertory

   If a new function is indispensable for RePertory to meet the timing constraints, it was implemented in a couple of weeks and was used for further timing improvement (A4 in Fig.3).

3. Changes in RTL, synthesis, IPO and P&R

   In some critical paths, repeater insertion was not as effective for performance improvement as changes in RTL, synthesis/IPO parameters and P&R parameters. In this project, the following methods were effective in timing improvement: Fanout division, use of wider wires without repeaters, change of the net order in wiring, creation of repeater-only regions (A1-3 and A5 in Fig.3).

V. RESULTS AND ANALYSIS OF THE APPLICATION TO THE EE DESIGN

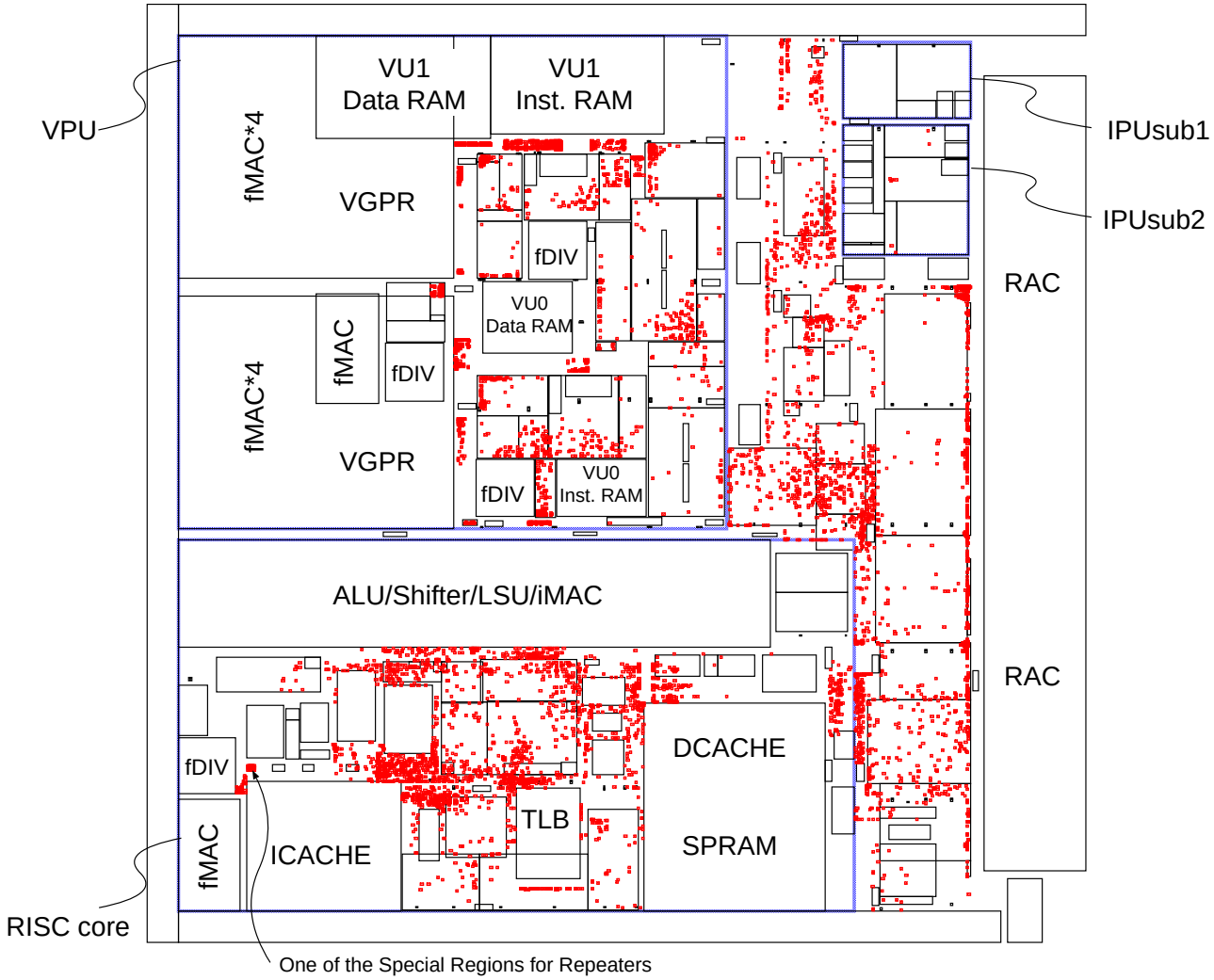

Fig.5. Distribution of repeaters in the EE

TABLE II
INCREASE IN NUMBER OF NETS AND CELLS AFTER REPEATER INSERTION

module	# of nets			# of cells		
	before insertion	after insertion	increase rate	before insertion	after insertion	increase rate
Top Module	95153	97066	+2.01%	81866	83779	+2.34%
RISC core	59342	61536	+3.70%	51260	54354	+4.21%
VPU	71021	72654	+2.30%	61603	63236	+2.65%
IPUsub1	16514	16514	±0.00%	14869	14869	±0.00%
IPUsub2	20705	21713	+0.04%	18569	18577	+0.04%
whole EE	262735	269483	+2.57%	228167	234815	+2.91%

TABLE III
INCREASE OF SC CELL AREA AFTER REPEATER INSERTION

module	before insertion (mm^2)	after insertion (mm^2)	increase rate
Top Module	9.44	9.79	+3.70%
RISC core	4.75	5.11	+7.59%
VPU	6.75	7.02	+4.00%
IPUsub1	1.36	1.36	±0.00%
IPUsub2	1.767	1.769	+0.08%
whole EE	24.06	25.05	+4.08%

A. Achievement of goals

All of the requirements described in section II were achieved with RePertory in this project. Fig.4 shows the slack of paths in the RISC core that are longer than a single stage and hence pass through multiple latches. Because the EE is a latch-based design, timing analysis of critical paths through multiple latches is essential. According to fig 4, hundreds of critical paths were resolved by repeater insertion.

A.1. Processing time

RePertory was written in Perl and executed on Sun UltraSPARC-II 360MHz computers. Processing time (calculation of repeater location and relocation of repeaters that were overlapping other cells) of RePertory for each module is described in table I.

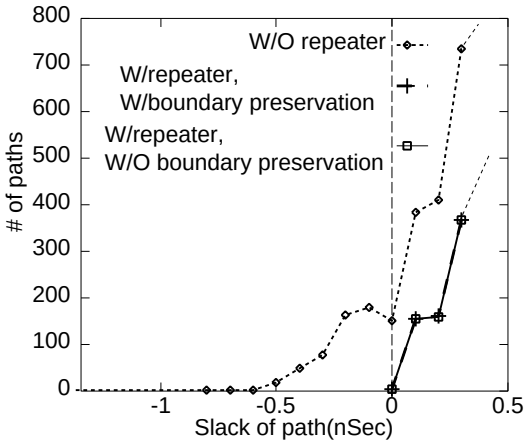

Fig.4. Slack of critical paths with multiple latches in RISC core with and without repeater insertion

TABLE IV
NUMBER OF REPEATERS WITH AND WITHOUT BOUNDARY PRESERVATION

	boundary preservation	
module	yes	no
Top Module	1913	1942
RISC core	2194	2202
VPU	1633	1646
IPUsub1	0	0
IPUsub2	8	8
whole EE	5748	5798

The top module took a longer time than the RISC core even though the number of repeaters in this module was less than that in the RISC core. Because the top module had almost twice as many nets as the RISC core. It also had more "repeater prohibited areas" than the RISC core which caused a longer time for relocation. The VPU took the longest processing time because the location of repeaters was restricted due to the small wiring-area and it took a long time to relocate repeaters distributed in the wiring-area into the specified repeater regions.

Because 5 machines were used in parallel, it took only 25.5 minutes to insert 5748 repeaters into the EE chip which was fast enough for this project.

B. Analysis of the repeater insertion results

B.1. Impact of the inserted repeaters

The increase in the number of nets (+2.57%) and cells (+2.91%) after repeater insertion is shown in table II and table III shows the increase of SC area (+4.08%) after repeater insertion. This is even less than the area increase caused by the spare gate insertion.

B.2. Impact of the module boundary preservation

Table IV shows that the number of repeaters slightly increased when repeaters were allowed to be placed inside the SC regions. But this was forbidden in this project in order to prevent the module boundary from changing.

In the whole EE , the difference of the number of repeater cells between two ways of repeater insertion in the table IV was less than 0.02%. Fig.4 shows that there is no difference between the results with boundary preservation and without it. In other words, preserving the module boundary did not have adverse effects on timing.

VI. CONCLUSION

The practical repeater insertion tool RePertory and its satisfactory results from the application to a 300MHz 128-bit 2-way Superscalar Microprocessor were presented.

The following features will be implemented in the next generation of RePertory:

- Flexibility for use in other projects. Features specific to the EE such as parameter files, design rules, etc. have to be isolated from program codes.
- Efficient use of timing constraints in order to avoid too many repeaters in non-critical paths.
- Shorter processing time by using the C language. RePertory is coded in Perl for easy maintenance and quick enhancements. But some functions like overlap removal of repeaters can account for most of the total run time; here is potential for an enormous speed-up.

ACKNOWLEDGEMENTS

The Authors would like to thank H. Landman of Toshiba America Electronic Components, Inc., S. Hayakawa, T. Osanai, T. Kamei, S. Koino, K. Hatanaka, M. Matsui and S. Kawakami of Toshiba Corporation for providing a lot of valuable advice, information, thoughts and scripts.

REFERENCES

[1] L. P. P. P. van Ginneken, "Buffer Placement in Distributed RC-tree Networks for Minimal Elmore Delay," *Proc. International Symposium on Circuits and Systems*, pp. 865-868, 1990.

[2] J. Lillis, C.-K. Cheng and T.-T. Y. Lin, "Optimal Wire Sizing and Buffer Insertion for Low Power and a Generalized Delay Model," *IEEE/ACM International Conference on Computer-Aided Design*, pp. 138-143, 1995.

[3] T. Okamoto, J. Cong, "Buffered Steiner Tree Construction with Wire Sizing for Interconnect Layout Optimization," *Proc. IEEE Intl. Conf. on Computer-Aided Design*, pp.44-49, 1996.

[4] "A Microprocessor with a 128bit CPU, 10 Floating-Point MACs, 4 Floating-Point Dividers, and an MPEG2 Decoder," *IEEE ISSCC Digest of Tech. Papers*, Vol.42, pp.256-257, 1999.

[5] "A High Bandwidth Superscalar Microprocessor for Multimedia Applications," *IEEE ISSCC Digest of Tech. Papers*, Vol.42, pp.258-259, 1999.

[6] T. Kamei, et.al., "A 300MHz Design Methodology of VU for Emotion Synthesis," *Proc. ASP-DAC*, to appear, 2000.