Delay-Optimal Wiring Plan for the Microprocessor of High Performance Computing Machines

Jun Kikuchi, Tetsuo Sasaki, and Kazuhisa Miyamoto
Enterprise Server Division, HITACHI, Ltd.
jkikuchi@kanagawa.hitachi.co.jp,
tsasaki@kanagawa.hitachi.co.jp,
kmiyamo@kanagawa.hitachi.co.jp

Tohru Hashimoto
HITACHI Information Technology Co., Ltd.
thashimo@kanagawa.hitachi.co.jp

Abstract– This paper presents a delay-optimal designing method for high performance microprocessor. In order to develop such a microprocessor in short period, a novel planning method for global wires, called "wiring plan" has been introduced. The goal of this wiring plan is to solve timing issue of wires with minimum usage of wiring resources; wiring channels and inserted buffers. In this wiring plan, global assignment of wiring resources and improvement of local congestion for their usage have been done. As a result of applying them, our chip has been able to work at 400MHz, and channel usage has not exceeded its capacity and gate overhead for buffer insertions has been only 1.6% of total gates.

1. Introduction

The goals of microprocessor design are to achieve high performance and to develop in short period[1]. In order to achieve high performance, we can take several means; to implement large and complicated functions in a chip, to make cycle time shorter, or to shrink LSI process. But these means make chip design issues more difficult[2,3,7,9,10]. Especially, the growth of wiring delay has been outstanding in deep sub-micron LSI design. Many approaches which had been intended to reduce wiring delay were proposed in recent years, such as, buffer insertion[4,5], planning of wiring layer[6] and wire width selection[1]. Now, it becomes one of key technologies for high performance LSI development to control global wiring delay.

In fact, we have had a chance to develop a new microprocessor, however it has seemed to be very hard to develop a chip which would achieve required performance in conventional way, such as post-layout optimization for timing closure. It might be able to find local optimal solution, but it was said to be unavailable to our new chip design that channel usage has been very close to the limit of total channel capacity and insertion of too many buffers brings increase of noise and power. Then to satisfy both timing issue and wiring resources usage in short time, novel planner for global wire has been necessary.

In this paper, we introduce a planning method for global wires, named “wiring plan”. The overview and design flow of our chip are introduced in section 2. Details of the wiring plan are described in section 3 and results of applying this method to our microprocessor are shown in section 4.

2. Chip Overview

2.1 Chip Overview

Our microprocessor has been designed to be used in high performance computers such like massively-parallel machines[8][11]. Figure 1 is a photograph of our chip. Table 1 shows the specifications of our chip. As shown in there, one of design goal of the processor is 400MHz operation with CMOS 0.25um process and 1.8V supply. Moreover, chip size is 18.5mm x 18.5mm, and it gives very large impact to timing design for long wires.

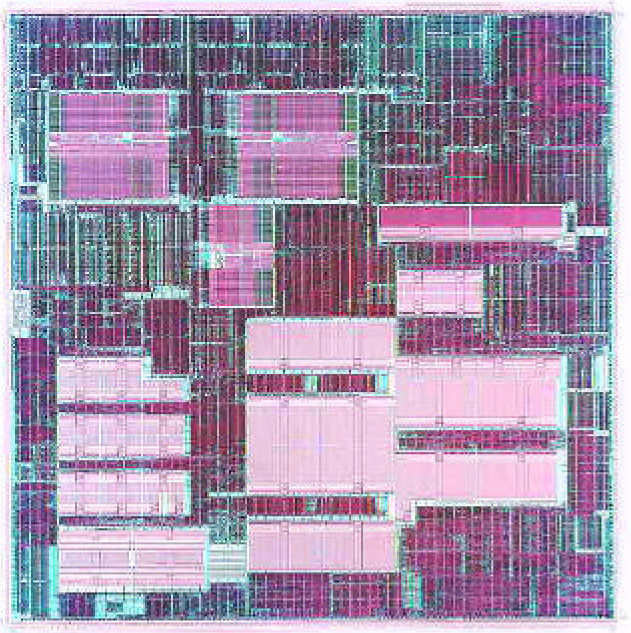


Figure 1 Photograph of the Microprocessor

Table 1 Specifications of the Microprocessor

Architecture	Based on 64 bit PowerPC
Design Rule	CMOS 0.25 um
Wiring Layer	7 Layer Metals
Operation Freq.	400 MHz
Supply Voltage	1.8 V
Gate/Trs. Volume	2.0 M gates/23 M Trs.
Chip Size	18.5 mm x 18.5 mm

2.2 Design Flow

Figure 2 shows outline of our chip design flow. The wiring plan assigns wiring resources to global wires based on the information from floorplan and timing budget, and informs these results to layout design. It works to reduce the number of long TAT loops such like feedback from post-layout timing verification to floorplan.

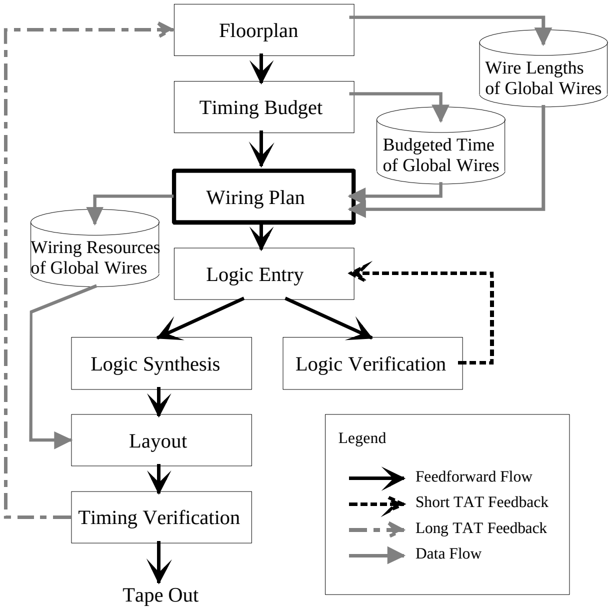


Figure 2 Microprocessor Design Flow

3. Wiring Plan at Top Level Design

3.1 Background and Motivation

Large scale chips are partitioned to some blocks. Nets are categorized into the two types, intra-block nets and inter-block nets according to their hierarchy. Inter-block nets are global wires whose wiring delays should be considered carefully because their lengths are very various.

Figure 3 shows path delay distributions of our chip estimated with three different way. A dashed line shows path delays which has been estimated in resource-pessimistic way (RP): no wiring resources have been used for wiring delay reduction. In this case, about half of paths have not been able to satisfy the target delay (Td). A doted line shows path delays of delay-pessimistic way (DP): wiring resources have been used to minimize wiring delay for all nets. A bold solid line shows path delays of the wiring plan (WP). Both worst delays of latter two cases have cleared Td. But mean point of distribution of DP has been shifted to left from that of WP. This implies that excessive usage of wiring resources has been caused in DP.

Figure 4 shows channel usage which has been estimated under same conditions as above. The channel usage for intra-block nets has been almost fixed and estimated that it would possess 70% of total channel capacity. As shown in RP case, even minimum channel usage has needed 87% of total capacity. In DP case, channel usage would be 106% of total capacity. The wiring plan would be a method that would satisfy timing constraints adjusting channel usage to its capacity.

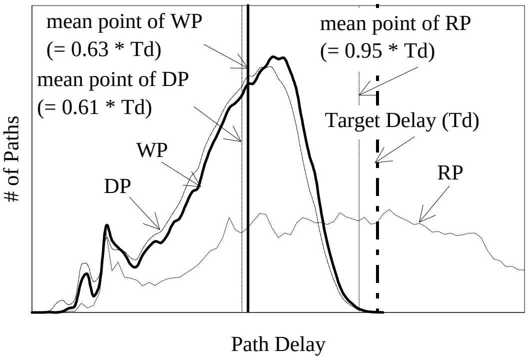


Figure 3 Estimation for Path Delay

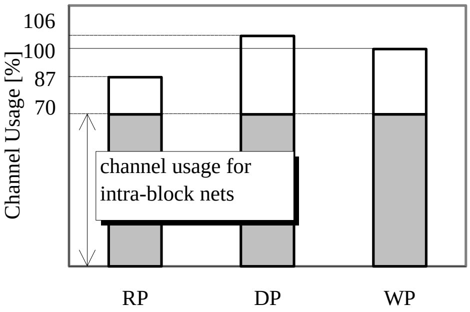


Figure 4 Estimation for Channel Usage

3.2 Wiring Class Assignment Issue

Usage of wiring resources for inter-block nets is shown in Figure 5. In our project, multi-level wiring layers has been used. Upper some layers are thicker than others for high-speed signal propagation. These are called “high-speed layer”. Wires on high-speed layers are broad and thick. Other layers are called “normal layer”. Normal width and broad width wires can be selected in normal layers. Buffers can be inserted in long wires for delay reduction. Interval of inserted buffers is categorized into two types, short insertion and long insertion. The short insertion makes fastest signal propagation. The interval of the long insertion is selected only to avoid cross talk noise errors.

We can use six combinations of wiring resources usage shown in Table 2. Six wiring delay curves according to these combination are shown in Figure 6. Wiring class indicates one of these combinations.

Selecting method of wiring classes is described as follows. The x-axis of Figure 6 is divided into four parts at three distinctive points; two points, p_0 and p_1, are where adjacent curves, (5) and (6) or (3) and (4), start to be spread. The point, p_2 is where two curves, (4) and (5), are crossing each other. Four parts of x-axis between these boundaries are set as R_L, R_M, R_S, and R_U. These parts are called “wire length ranges”. The characterful curves in each range are picked and indicated by circles in Figure 6. The curve of (2) can not be distinctive in any wire length range. Other five combinations of wiring resources are employed as wiring classes as shown in Table 2.

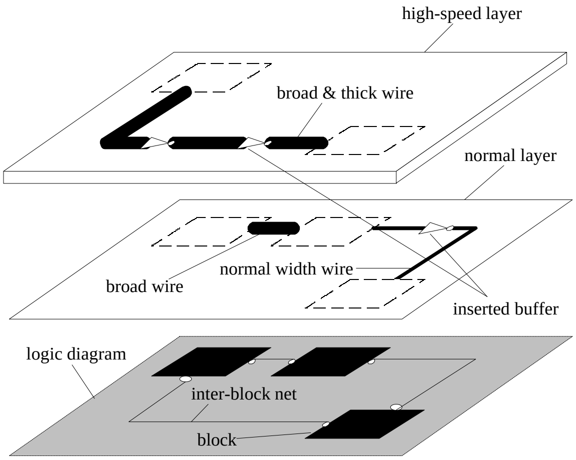


Figure 5 Wiring Plan for Inter-Block Nets

Table 2 Resource Usage and Wiring Class

#	wiring layer	wire width	interval of buffer insertion	wiring class
1	high-speed	broad	short	A
2	high-speed	broad	long	-
3	normal	broad	short	B
4	normal	broad	long	C
5	normal	normal	short	D
6	normal	normal	long	E

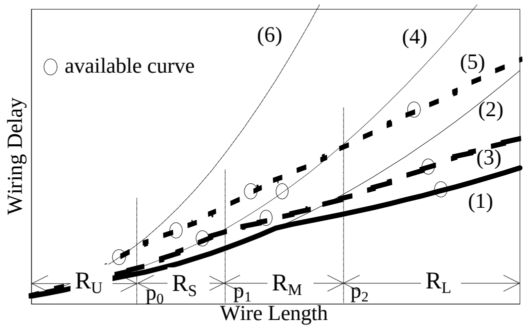


Figure 6 Wiring Delay Curves

Formulation of wiring class assignment issue is as follows. The objective function of wiring class assignment is to minimize sum of absolute values of slacks for all inter-block nets. This slack is a value which is subtracted estimated time from budgeted time. This objective function is represented as equation 1. In this equation, estimated time of net i, Te_i, is found from wiring delay curves in Figure 6 with its wire length and wiring resources.

And there are constraints for wiring resource capacities. One of wiring resources is wiring channel. Total wiring channel is sum of grid lengths of each wiring layer where wires can be put on. High-speed layers can be used for nets of class A and normal layers are used for other classes. Constraints for channel capacity for high-speed layers and normal layers must be considered respectively. These are represented as equation 2 and equation 3. These limits of channel capacity are approximated by wirability, total channel capacities, and pre-fixed channel usage of intra-block nets, clock nets and power/ground nets.

$$\sum_{i}^{\text{all nets}} \mid Tb_i - Te_i \mid \rightarrow \min \tag{1}$$

Tb_i: budgeted time of net i
Te_i: estimated time of net i

$$W_A * \sum_{j}^{\text{all nets in class A}} l_{Aj} \le L_H \tag{2}$$

$$\sum_{i}^{B,C,D,E} (W_i * \sum_{j}^{\text{all nets in class i}} l_{ij}) \le L_N \tag{3}$$

l_{ij}: length of wire j belonging the class i
L_H: channel capacity of high-speed layer
L_N: channel capacity of normal layer
W_i: wire width for class i

Meanwhile, number of inserted buffers is proportional to wire length as shown in equation 4. We have estimated that overhead of gates increase would become 5% of total gates in delay-pessimistic way. Inserted buffers tend to concentrate around macro blocks, RAMs and resister files. This local congestion of buffers causes noise problems. Increase of 5% gates of inserted buffers could not be guaranteed chip performance from noise analysis. In our project, upper limit of buffer overhead has been set to 2% from tolerance for noise as shown in equation 5.

$$n_i = \sum_{j}^{\text{all nets in class i}} l_{ij} / D_i \tag{4}$$

$$g * \sum_{i}^{\text{all class}} n_i \le G, \qquad G / G_0 \le 2\,\% \tag{5}$$

n_i: # of inserted buffers for class i
g: gate size per buffer
D_i: interval of buffer insertion for class i
G_0: total gate volume of the chip

To ease handling these constraints, constraints for channel and buffers should be mapped to constraints for wiring class usage. This is represented as equation 6.

For each class i;

$$\sum_{j}^{\text{all nets in class i}} l_{ij} \le L_i \tag{6}$$

l_{ij}: length of wire j belonging the class i
L_i: channel capacity of the class i

Channel capacities for each wiring class are determined as follows. A curve graph in Figure 7 shows distribution of number of inter-block nets according to their length. These inter-block nets are categorized to four groups by their length according to wire length range described above. And inter-block nets in each wire length range are partitioned into the number of available wiring classes equally, except wiring class A. The number of nets which can be assign to the wiring class A is subtracted from total number of nets in R_L. Initial capacities for each wiring class are represented by sizes of polygons. At last, it is needed to check if the number of buffers does not exceed its capacity. Number of buffers for each wiring class can be calculated by net length and interval of buffer insertion.

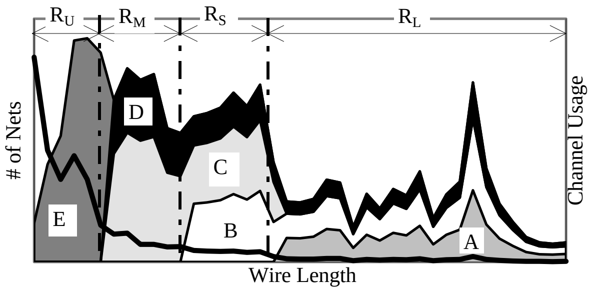


Figure 7 Initial Channel Budget

3.3 Procedures of Wiring Class Assignment

One of the roles of the wiring plan is to satisfy delay constraints with minimum wiring resources, and another role is to evaluate the acceptability of current design before layout design. If current design has not been acceptable, chip designers must take restructuring of floorplan, modification of logic design, e.g. moving of latch point, and so on. The wiring plan changes its role with progress of the chip design, and the process of the wiring class assignment is also changed. We have used twofold wiring resources assignment: global wiring class assignment considering total capacity of wiring resources and improvement of wiring resources usage considering local congestion.

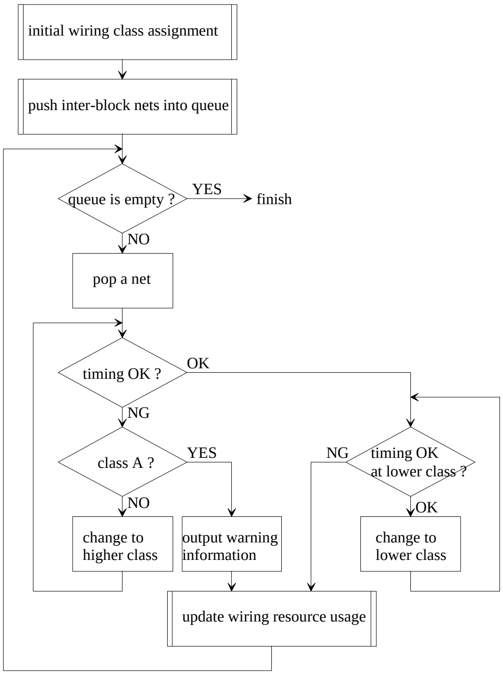
Details of global wiring class assignment are described as follows. The most significant role of global wiring class assignment is to let chip designers know the feasibility of current design as soon as possible. It is important that process time is short and output information is easy to analyze capacity violations. The flow of global wiring class assignment is shown in Figure 8.

First step, initial wiring class assignment is done in each wire length range as shown in Figure 7. Nets in each wire length range are reordered by their slack values. And wiring classes are assigned to nets from higher class to lower class until total length of assigned wire exceeds capacity of current wiring class.

Next, classes are assigned to nets to satisfy timing constraints. Order of nets to be assigned wiring class is determined by their slack values, but it is the reverse of first step. To make capacity of wiring resources, nets which have large slack values are reassigned to lower wiring class first. Then, nets which have negative slack values are reassigned to higher class using capacity of wiring resource which are made previously. In this assignment process, if timing constraint of a net has not been able to be satisfied even with the highest class A, warning messages are output to chip designers with information which includes net length, estimated time and budgeted time. Warning messages for wiring resources usage are also output, when channel usage or buffer usage exceeds its capacity.

Meanwhile, improvement of wiring resources usage reduces local congestion. To manage local congestion, chip area is divided into some small areas like a mesh. This small area is called “cell”. A set of whole these cells is called congestion map. An example of congestion map for channel usage is shown in Figure 9. This chip area was divided into 64 x 64 cells. Height of the population graph shows quantity of nets which run through there.

Improvement process of wiring resources usage is done as follows. First, nets are prioritized by wiring class and their length. Nets of lower class are easy to find alternative class, and exchanging wiring class of shorter nets gives small impact on wiring resources usage. Next, nets running through cells where wire/buffer usage exceed their capacities, are picked up, and alternative wiring classes which satisfy their timing constraints are given to these nets.



initial wiring class assignment
push inter-block nets into queue
queue is empty ?
YES
finish
NO
pop a net
timing OK ?
OK
NG
class A ?
YES
NO
change to higher class
output warning information
NG
timing OK at lower class ?
OK
change to lower class
update wiring resource usage

Figure 8 Global Wiring Class Assignment

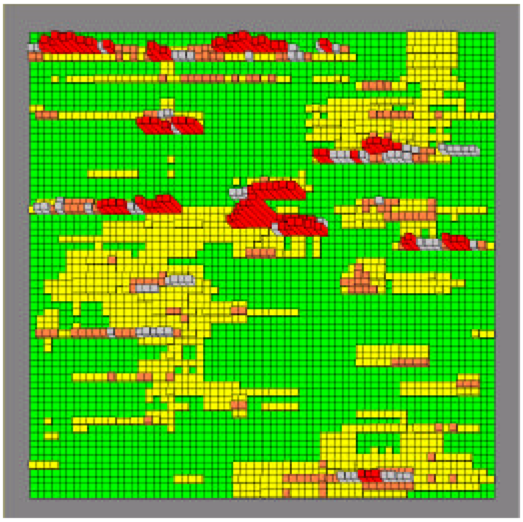


Figure 9 Example of Wire Congestion Map

4. Results of Applying the Wiring Plan

In this section, results of applying the wiring plan are described as follows. The wiring plan has been applied to our microprocessor chip. The statistics for wires of our chip is shown in Table 2.

4.1 Chip Performance

As a performance target of our chip design, the final path delay distribution of our chip is shown in Figure 10. The x-axis represents time for path delay and the y-axis represents sum of paths whose delays belong with the x-axis scale. As shown in this figure, no path delay exceeds the target delay of the chip, 2.5ns, by the wiring plan, and our microprocessor can work at 400MHz.

4.2 Wiring Resources Usage

Comparison for wiring resources usage is as follows. A method which assigns wiring resources to global wires in delay-pessimistic way (DP) is compared with the wiring plan. Wiring classes are assigned to nets according to their lengths without consideration of their slack values in DP. Wiring resources usage of three cases, design target, DP, and wiring plan are shown in Table 4.

Channel usage of design target has been set to 400m, from total channel capacity, wirability of the chip, and so on. Channel usage in DP case is 430m, and this is 108% of design target value. However, with the wiring plan, channel usage becomes 391m, and it clears design target.

Number of wires for each wiring class is shown in Table 5, and final channel usage of each class in each wire length range is shown in Figure 11.

As shown in Table 5, numbers of nets in class B and class C are different between final result of the wiring plan and DP. It can be said that excessive usage for higher classes in DP has been avoided by the wiring plan. Table 5 also shows transitions of numbers of nets which have been assigned to each class in the wiring plan. Especially, it is outstanding that the number of nets for class A increases from 1818 to 2941, though capacity of class A does not change. This means that average length of nets belonging to class A has become shorter. As a proof of that, Figure 11 shows that about half of nets in R_M are assigned to class A, though class A has been assigned to only nets in the range R_L initially.

Meanwhile, budgeted value of inserted buffers of design target has been set to 40k gates, or 2.0% of total gate volume, due to tolerance of noise, as described above. Buffer overhead in DP case is estimated as 100k gates, and this is more than twice of design target value. However, buffer overhead by the wiring plan is 32k gates, and it clears its design target.

Final result of inserted buffer distribution is shown in Figure 12. In this figure, height of the population graph and thickness of shade represent the quantity of inserted buffers in each cell. The black rectangles are macro areas, RAMs and register files, where inserted buffers can not be placed on. As a consequence of avoidance of macro areas, surrounds of macro areas are crowded by inserted buffers. But numbers of inserted buffers in these areas are resulted by improvement of local congestion of the wiring plan. It is confirmed by some simulators that no fatal error due to noise has been found in this chip.

4.3 Development Period

At last, comparison of development period between our new chip and our previous chip is shown in Figure 13. Gate volume of new chip is about four times of previous chip and its target clock frequency is twice of previous.

The development period of new chip has been almost same as that of previous chip with wiring plan in spite of those increase of volume and difficulty of chip design.

Figure 13 also shows their design time ratios between logical design and layout design. Layout design had spent longer time than logical design in previous case, because many iterative delay optimizations had been done in layout design phase. Since the wiring plan has been introduced, layout design has required shorter time than previous. Due to reduction of layout design time, much time has been able to assigned to logical design, such as design entry and logic verification, for large and complicated logic of our new microprocessor.

Table 3 Statistics for Wires of our Chip

Total # of Wires	939,613
Total Wire Length	367 m
# of Global Wires (percentage for total)	42,725 (4.5 %)
Global Wire Length (percentage for total)	83 m (22.6 %)

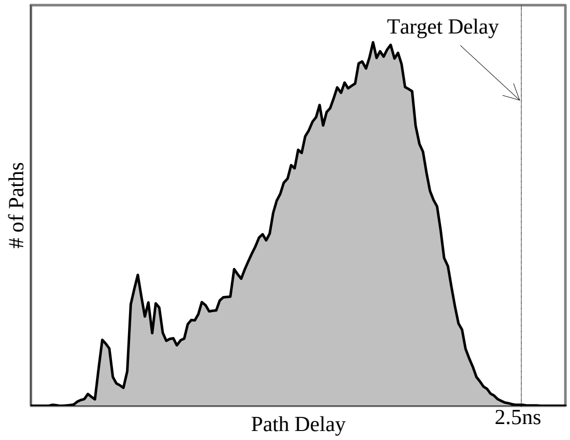


Figure 10 Final Path Delay Distribution

Table 4 Wiring Resources Usage

Resource	Channel Usage	Overhead of Inserted Buffers
Design Target	400 m	40 k gates
DP (comparison with target)	430 m (108 %)	100 k gates (250 %)
Wiring Plan (comparison with target)	391 m (98 %)	32 k gates (80 %)

Table 5 Number of Wires for Each Wiring Class

Wiring Class	Wiring Plan		DP
	Global Assign.	Final Result	
A	1,818	2,941	3,194
B	1,443	1,547	3,411
C	4,158	4,265	1,611
D	4,461	3,987	3,664
E	30,845	29,985	30,845

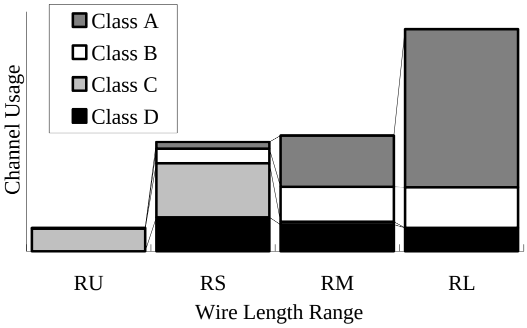


Figure 11 Final Result of Channel Usage

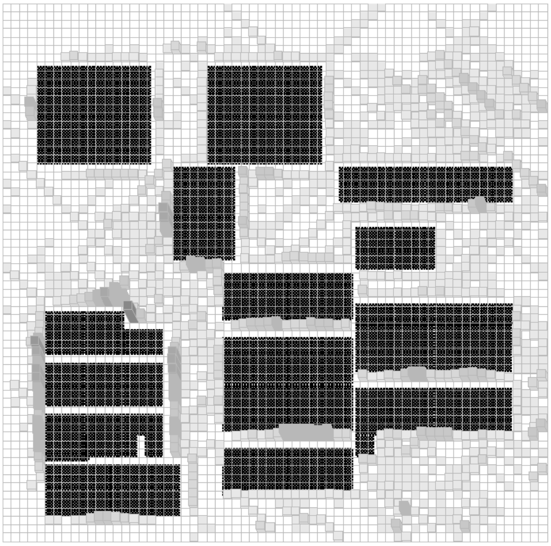


Figure 12 Final Buffer Distribution

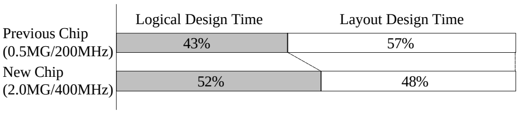


Figure 13 Comparison of Development Period

5. Conclusion

We have presented a novel wiring plan method at top level design for high performance microprocessor. The task of this wiring plan has been to assign wiring resources, channel usage and inserted buffers, to inter-block nets, considering both timing closure and wiring resources limitation simultaneously. To achieve such a goal in short time, global assignment of wiring resources with wiring class and improvement of local congestion for their usage has been done hierarchically. As a result of applying this wiring plan to development of our microprocessor, it has been able to work at 400MHz with no violation for wiring channel usage and overhead of inserted buffers has been only 1.6% of total gates.

Acknowledgment

We would like to thank Tadahiko Nishimukai and Eiki Kamada for giving an opportunity to develop the new high performance microprocessor, and also thank Ryo Yamagata, Tatsuki Ishii and Yohsuke Nagao for their useful suggestions to our project.

Reference

[1] H.Terai, K.Gemma, Y.Nagao, Y.Satoh and Y.Ohno, "Basic Concept of Cooperative Timing-driven Design Automation Technology for High-speed RISC Processor HARP-1", Proc. of 31st DAC, pp.262-269, 1994.

[2] D.E.Hoffman, R.M.Averill, B.Curran, Y.H.Chan, A.Dansky, R.Hatch, T.McNamara, T.McPherson, G.Northrop, L.Sigal, A.Pelella and P.M.Williams, "Deep Submicron Design Techniques for the 500MHz IBM S/390 G5 Custom Microprocessor", Proc. of ICCD, pp.258-263, 1998

[3] S.Posluszny, N.Aoki, D.Boerstler, J.Burns, S.Dhong, U.Ghoshal, P.Hofstee, D.LaPotin, K.Lee, D.Meltzer, H.Ngo, K.Nowka, J.Silberman, O.Takahashi and I.Vo, "Design Methodology for a 1.0 GHz Microprocessor", Proc. of ICCD, pp.17-23, 1998

[4] C.J.Alpert, A.Devgan and S.T.Quay, "Buffer Insertion for Noise and Delay Optimization", Proc. of 35th DAC, pp.362-367, 1998.

[5] J.Culetu, C.Amir and J.MacDonald, "A Practical Repeater Insertion Method in High Speed VLSI Circuits", Proc. of 35th DAC, pp.388-391, 1998.

[6] R.H.J.M.Otten and R.K.Brayton, "Planning for Performance", Proc. of 35th DAC, pp.122-127, 1998.

[7] D.Edelstein, J.Heidenreich, R.Goldblatt, W.Cote, C.Uzoh, N.Lustig, P.Roper, T.McDevitt, W.Motsiff, A.Simon, J.Dukovic, R.Wachnik, R.Schulz, L.Su and J.Slattery, "Full Copper Wiring in a Sub-0.25um CMOS ULSI Technology", Proc. of IEDM, 1997.

[8] N.Ohkubo, T.Kawashimo, M.Suzuki, Y.Suzuki, J.Kikuchi, M.Tokoro, R.Yamagata, E.Kamada, T.Yamashita, T.Shimizu, T.Hashimoto and T.Isobe, "A Fault-Detecting 400MHz Floating-Point Unit for a Massively-Parallel Computer", ISSCC Digest of Technical Papers, vol.42, pp.368-369, 1999.

[9] P.Bames, "A 500MHz 64b RISC CPU with 1.5MB On-Chip Cashe", ISSCC Digest of Technical Papers, vol.42, pp.86-87, 1999.

[10] D.H.Allen, G.Aipperspach, D.T.Cox, N.V.Phan and S.N.Storino, "A 0.2um 1.8V SOI 550MHz 64b PowerPC Microprocessor with Copper Interconnects", ISSCC Digest of Technical Papers, vol.42, pp.438-439, 1999.

[11] T.Kurihara, E.Kamada, K.Shimada and T.Shimizu, "A RISC Processor for SR8000: Accelerating Large Scale Scientific Computing with SMP", Proc. of Hot Chips 11, pp.13-22, 1999.