Clock Design of 300MHz 128-bit 2-way Superscalar Microprocessor

Fujio Ishihara

Processor Development Group
System ULSI Eng. Lab.
Microelec. Eng. Lab.
TOSHIBA CORPORATION
580-1, Horikawa-cho, Saiwai-ku
Kawasaki 210-8520, JAPAN
Tel: +81-44-548-2523
Fax: +81-44-548-8324
e-mail: fuji.ishihara@toshiba.co.jp

Christian Klingner

TOSHIBA AMERICA
ELEC. COMPONENTS, INC.
1060 Rincon Circle, San Jose
California 95131-1325, U.S.A.
Tel: +1-408-514-2610 (ex. 1104)
Fax: +1-408-514-2610 (ex. 1104)
e-mail klingner@mail.com

Ken-ichi Agawa

Mobile and Network Development. Group
System ULSI Eng. Lab.
Microelectronics Engineering Lab.
TOSHIBA CORPORATION
580-1, Horikawa-cho, Saiwai-ku
Kawasaki 210-8520, JAPAN
Tel: +81-44-548-2341
Fax: +81-44-548-8332
e-mail: agawa@sdel.toshiba.co.jp

Abstract— Less than 116ps overall clock skew has been achieved across the 15.02mm×15.04mm die by balanced clock path routing and differential clock signal distribution in the global clock tree of 300MHz 128-bit 2-way superscalar microprocessor. The shared clock wire configuration and clock buffer layout patterns over the whole die enhance the clock skew insensitivity to process fluctuation. Combination of three different clock tuning methods is successfully applied to the entire clock tree and the clock skew is minimized efficiently within a limited design period.

I. Introduction

Clock skew must be minimized in order to improve the maximum clock rate, evade a race condition, and guarantee more timing margin for circuit designers [1]. On-die clock skew, however, is getting difficult to control because RC interconnect delay becomes more significant as the die size of microprocessors and other logic LSI's becomes larger and the metal pitch dwindles. Distributing clock signals across a die with small skew requires us to employ special design methodologies, such as balanced H-like trees, die-wide grid clock network, and low resistivity interconnects in circuit layout [2], [3], [4], [5]. Automated clock tree tuning is also an indispensable technique to build up a large scale microprocessor in a limited design period.

This paper will discuss the clock distribution methodology and the clock tree tuning flow for a 300MHz 128-bit 2-way superscalar microprocessor [6], [7]. Both of the techniques enable us to reduce the simulated clock skew to less than 116ps.

II. Clock Distribution Methodology

The technology features of our die are shown in Table I. Careful design of the clock distribution network is necessary in order to minimize clock skew across the whole die with large area and dense transistor population.

TABLE I
TECHNOLOGY FEATURES

Die size	15.02mm × 15.04mm
Transistor count	13.5M
Supply voltage	1.8V
Typical Leff	0.18μm
Metal1 pitch	0.64μm
Metal2 pitch	0.80μm
Metal3 pitch	0.80μm
Metal4 pitch	4.00μm

Many studies on automation of clock layout and synthesis have been reported [8], [9], [10], and an ASIC design flow can utilize the automated methodologies effectively because of its standard cell (SC) based flexible layout. In contrast, almost all the clock network of this die is designed and drawn manually since a high percentage of the die area is occupied by custom blocks (CB's) and the floor plan is not flexible enough to automate the clock design. An advantage of the handicrafted clock network is that both CB's and SC regions can share the common clock wire configuration and the identical clock buffer layout patterns. By using the same wire and buffer layout geometry, clock buffer load constituents (gate/wire capacitance ratio) can be uniform in each clock tree level and process fluctuation has a constant effect on each clock delay. Consequently clock skew becomes insensitive to the process variation.

Fig.1 shows the clock hierarchy of the die. Seven levels of clock buffers are responsible for propagating clock signals from the PLL outputs down to the local clocked elements. 'f' refers to the microprocessor core operating frequency (= 300MHz). The entire clock tree consists of two portions, the

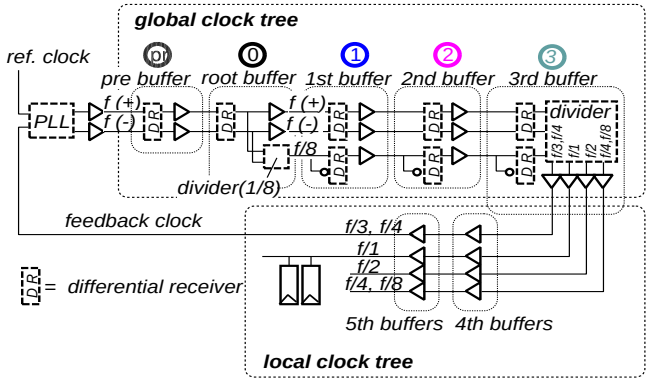

Fig. 1. Clock hierarchy

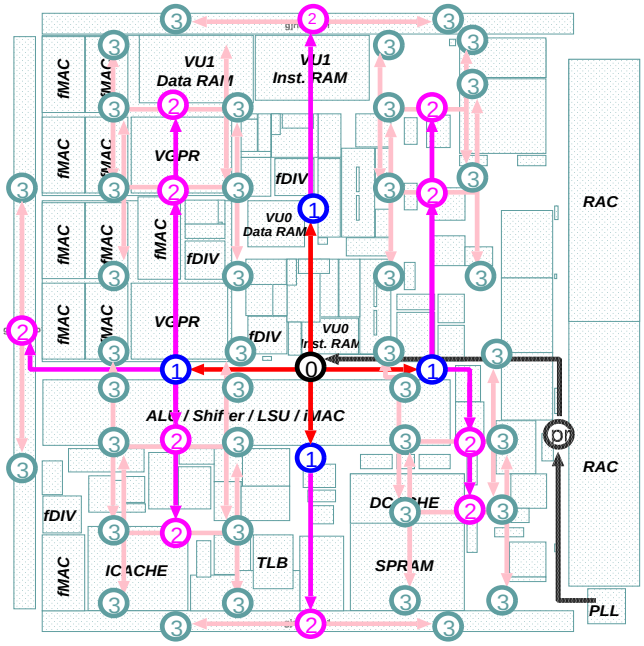

Fig. 2. Global clock distribution network

global tree starting from the PLL outputs to the 3rd buffer and the local tree starting from the 3rd buffer outputs down to local latches and flip flops. One of the 5th buffer outputs is fed back to the PLL to form a feedback loop.

A. Global Clock Distribution

The global clock tree, originating from the PLL output to the 3rd buffers, takes care of die-wide clock distribution as shown in Fig.2. The PLL output is routed to the root buffer (⓪) located at the die center via the pre-buffer (ⓟr). From the root buffer to the 3rd buffers (③), each clock level employs either a cross-shaped or H-like balanced tree structure to achieve matched clock path length. Metal 4 is used for the global tree interconnect because of its low resistivity and small RC-induced skew.

The global clock signals travel all over the die and they tend to be vulnerable to coupling noise caused by the transition of neighboring signals. As can be seen from Fig.1 schematic, the global clocks in our design are propagated differentially through the differential receivers together with the 'f/8' clock. By utilizing the differential clocks, common-mode noise imposed on Metal 4 clock wires by underlying signal interconnects can be effectively cancelled. The 'f/8' clock is generated in the root buffer to serve as a synchronization signal for the frequency dividers in the multiple 3rd buffers placed across the die. The frequency divider provides the divided clocks to the local clock network. Since the 'f/8' clock acts simply as the phase reset signal for the 3rd buffer dividers, its timing requirement is more relaxed than that of the complementary 'f' clocks in the sense that the slow clock is allowed to have edge fluctuation of half the cycle time of 'f' in order for the clock to be properly latched on the 'f' edge in the 3rd buffer divider. Therefore, the differential distribution is not applied to the 'f/8' clock and only its positive phase is propagated through the global tree. The differential receiver is still used for the 'f/8' clock to equalize the buffering delay with that of the 'f' clocks. Due to the vulnerable nature of the global clock to the surrounding noise, shielding wires are inserted between the three parallel global clock interconnects, 'f(+)', 'f(-)', and 'f/8'.

The entire global clock connection is represented both in the register-transfer-level (RTL) description and in the CB schematics.

B. Local Clock Distribution

The local clock tree supplies single-phase divided clocks to the local clocked elements via the 4th and the 5th buffers. Clock paths in the tree have unequal length and clock path RC delay causes the clock skew directly. Unlike the clock wires in the global tree, those in the local tree are drawn with lower metal layers (Metal 1, 2, and 3) and they are, thus, subject to local layout restrictions. A detailed layout guideline on clock wire length/width and shielding/spacing width, has been set up for CB designers and place-and-route (P&R) tool operators to control the local clock skew within a budgeted range.

Fig.3 shows the local clock structure of our design. In the SC region, the 5th buffer outputs are shorted together and connected to the ladder-shaped clock network so that every cell can tap the clock signal from the identical clock net and RTL designers of random logic blocks do not have to specify multiple clock nets in their code. On the other hand, the CB designers place the 5th buffers separately to each loading group and the buffer outputs are not merged. This is because the buffer outputs are sometimes distant and shorting wires would be unnecessarily long. Such long shorting interconnects are

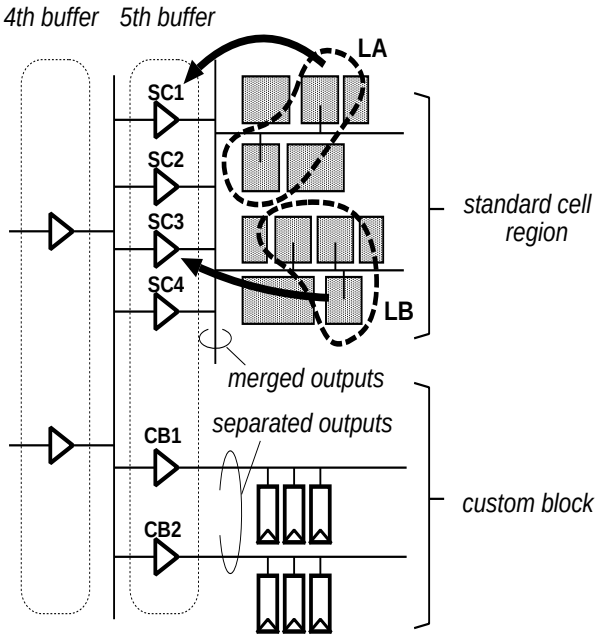

Fig. 3. Local clock distribution

not only area consuming but also practically meaningless in reducing clock skew due to their high resistance.

The SC 5th buffers and their connections are generated and inserted automatically in the gate-mapped netlist before the P&R process. All the CB and SC 4th buffers, the CB 5th buffers, and their connections are described explicitly in the RTL code and in the CB schematics. This technique provides random block RTL designers freedom to modify their code frequently and try an updated floor plan in short turn-around time.

III. Clock Tuning Flow

Clock tuning is one of the final stages of the tapeout process and its flow must be automated to get accurate tuning results promptly. Our design employs three different automated tuning methods shown in Fig.4 for adjusting clock buffer drive strength. The most appropriate method of the three is applied to each level of the entire clock tree. All RC parasitics are extracted from the layout in the detailed standard parasitic format (DSPF) using a third party extraction tool.

A. Method-A : Global tree and 4th buffer outputs

Despite our effort to implement a balanced global clock network, path length difference and gate loading imbalance are inevitable in some parts of the die. Since the clock buffers in the global clock tree are placed sparsely and less than 60 buffers take care of the entire die clock distribution, they tend to drive the die-size-comparable long clock interconnects and their drive strength needs to be tuned by adjusting the buffer size according to the local situation of wire and gate loadings.

In the beginning stage of this method, all the clock buffers in a level are assigned the initial uniform buffer size and clock delays are evaluated with the initial size. When the delay and skew results are obtained, a numerical analysis tool determines the next buffer size candidates by calculating clock delay derivatives, and the tool feeds them to the second delay evaluation. This iteration continues until the simulated skew meets the budget. In order to reduce the number of buffer size variables and the time for derivative calculation, neighboring buffers are grouped together and are assigned the same buffer size. The 4th buffer tuning takes the most advantage of the grouping benefit since the 4th buffer count is much larger than upper buffer counts in the global clock tree.

The iterative skew improvements for three different 4th buffer outputs are plotted in Fig.5. Four to five iterations are enough to achieve the budget skew (see Table II) irrespective of the clock domain. This tendency is also the case with upper clock levels.

B. Method-B : 5th buffer outputs in standard cell regions

Another tuning method is used for the 5th buffer resizing in SC regions. Because the number of 5th buffers is much larger than that of the upper level buffers, Method-A, an iterative approach, would be very time consuming and inappropriate.

In Method-B, abundant number of 5th buffers are embedded in the SC region layout prior to the tuning. The next step is to assign each gate and wire capacitive loading to the closest buffer as shown in Fig.3. The figure depicts the situation where two loading groups, ‘LA’ and ‘LB’, are assigned to their nearest 5th buffers, ‘SC1’ and ‘SC3’, respectively. Such correspondence between loadings and buffers cannot be obtained simply by examining circuit connectivity, but the correspondence must be determined by using DSPF geometry information. In the SC 5th buffer clock level, all local clocked elements are connected to the same clock net and the connectivity information only gives us the ratio between the total buffer count and the total amount of capacitive loading. If we do not take the geometry into account, local balance between the buffer count and the surrounding loadings is overlooked and it results in large clock skew.

Following the load assignment to the buffers, excessive buffers are picked up and deleted from the netlist. It should be noted that this method trims the drive strength of the 5th buffers not by shrinking the buffer size, but by reducing the buffer count. The major reason why the tuning is done in this manner is that the size tuning requires a wide variety of standard cell clock buffers with different cell names, but the buffer count tuning can be accomplished easily by deleting unnecessary buffer instances out of the netlist. Method-B is a non-iterative approach which realizes locally balanced load-to-buffer-size ratio in every buffer’s vicinity.

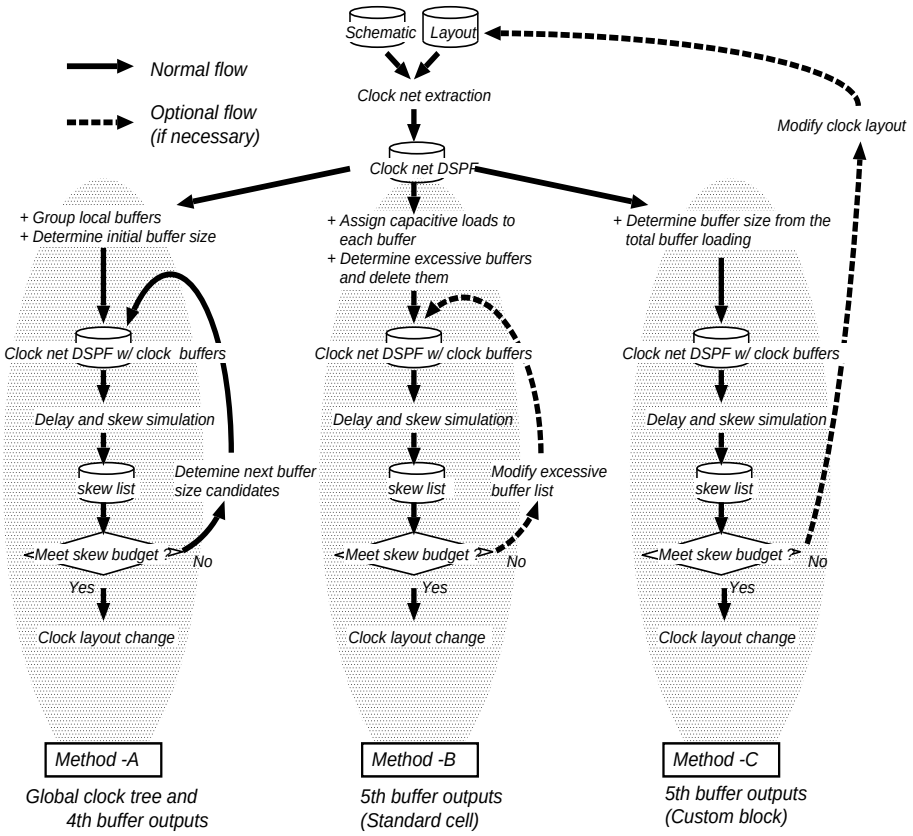

Fig. 4. Clock tuning methods

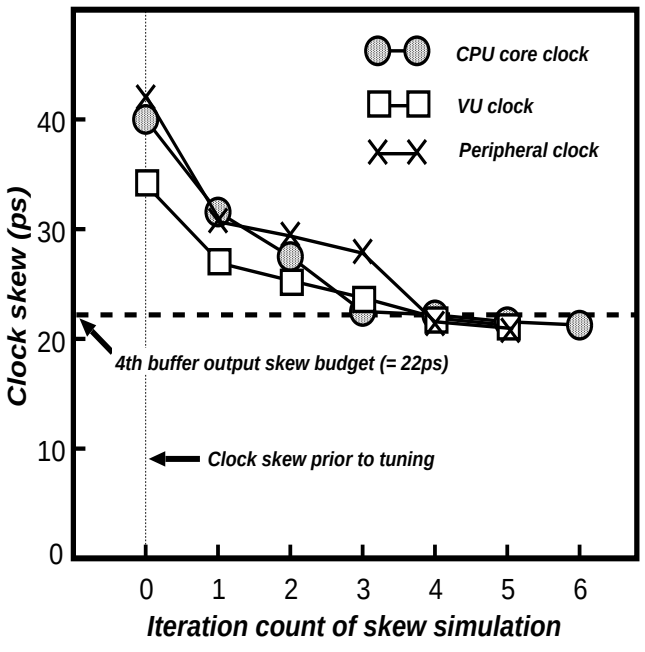

Fig. 5. Skew transition during the 4th clock buffer tuning

C. Method-C : 5th buffer outputs in custom blocks

As for the 5th buffer outputs in the CB's, much simpler tuning is employed. As shown in Fig.3, the outputs of the CB 5th buffers are separated and each 5th buffer size can be determined only from its output loading. In this case, the DSPF geometry information is not needed and connectivity information between a buffer and its loading is sufficient to determine the appropriate buffer drive strength.

Unlike the clock tuning in SC regions, the drive strength of the 5th buffers in the CB's is tuned by modifying the buffer size for two major reasons. Firstly, the buffer size tuning provides fine control of the buffer drive strength in contrast to the buffer count tuning especially in the situation where the clock buffer outputs are separated. This is the case with the CB 5th buffers while it is not with the SC 5th buffers. Secondly, in the buffer count tuning, initial placement of the abundant buffers wastes some layout area because a part of the initial buffers are removed by tuning. Such area inefficiency should be avoided in densely packed CB layout patterns.

In order for the above three methods to work effectively and for clock skew to be confined within the budget, CB layout designers and P&R operators must follow the detailed guideline on clock layout.

IV. Clock Skew Results

The high operating frequency of the die requires tight control over the clock skew. The clock skew budget and the simulated post-tuning clock skew value for each clock level is listed in Table II. The budget values are determined from initial RC delay estimation in the early stage of our design. Larger target skews are budgeted to the local clock tree because of its unmatched clock path length and non-H-like network structure. The relaxed skew budget is also due to higher resistivity of the lower metal wires since the lower clock tree is drawn with these layers.

Skew is simulated level by level by measuring a delay between the clock driver input and each loading gate input using a SPICE engine. In this simulation, zero-skew synchronized waveforms are assumed for all the clock driver inputs. Each output terminal of the root, 1st, and 2nd buffers independently drives separate loading and thus an individual clock path delay from the root buffer to the 3rd buffer can be obtained by simply summing each level clock delay in the path. The clock skew in the top row of the table is calculated in this manner.

As shown in Fig.5, the pre-tuning clock skews in the 4th - 5th level is 1.5 - 2 times the budget and Method-A successfully improves the skews down to the goal value with reasonable iteration counts. As is obvious from the table, all of the simulated skews are less than or equal to the budget values with the only slight violation in the 3rd - 4th level. The simulated total skew meets the budget with small margin and it indicates that each one of the three different tuning methods works effectively to reduce the clock skew in every clock level. It is remarkable that the obtained total clock skew, 116ps, is comparable to or better than the reported skew values in the recent publications when die area, transistor count, metal pitch, metal layer count, and metal material are taken into consideration [5],[11],[12].

Fig.6 shows the accumulated clock delay from the root buffer input down to the 5th buffer inputs. It should be noted that the accumulated delay values in the figure are calculated by summing the clock delay simulation result of each clock level with buffer location taken into account, and that they are not obtained by running the SPICE engine for the entire clock tree netlist. The delay surface and its contours, therefore, shows rough image of the clock distribution. The waving terrain in both x- and y-directions reflects the periodical placement of the 3rd and the 4th buffers. It seems that the right side of the die is supplied with a bit earlier clock compared with the left side. It is due to the 3rd buffer deployment on the right side of the die being a bit more dense than that on the left side, but neither high mountains nor deep valleys are observed and the overall clock skews are well controlled. The overall skew after Fig.6 is 48ps and it is far smaller than 73ps, a simple sum of the level-by-level simulated skews in Table II from the root buffer to the 5th buffer inputs (= 28 + 24 + 21ps). This is because the simple summation picks up the worst skew in each clock level without considering geometry information and, thus, it results in the too pessimistic value. The more realistic skew, however, must be estimated by summing the level-by-level skews considering the geometry as is done in Fig.6. The figure implies that the real total clock skew may be less than 116ps in the table by about 35ps (= 73 - 48ps), but the total skew of 116ps is still the value we can state clearly in this paper as the possible worst case skew since an accurate overall skew should be obtained by SPICE simulation for the entire clock tree. Such SPICE simulation, however, is not feasible due to a huge size of the clock netlist.

TABLE II
CLOCK SKEW BUDGET AND SIMULATED SKEW

Clock level	Budget(ps)	Simulated skew (ps)
Root - 3rd	36	28
3rd - 4th	22	24
4th - 5th	22	21
5th - Latch, F/F	18	18
Total RC skew	98	91
Process fluctuation	25	25 (Not simulated)
Total skew	123	116

V. Summary and Conclusions

The detailed planning and the symmetrical structure of the clock network, the common configuration of the clock wires and the buffer layout patterns between the CB's and the SC regions, and the noise-tolerant differential clock distribution are the key features in reducing the clock skew successfully. Combination of the three different automatic tuning methods is applied effectively to the entire clock tree resulting in the total clock skew of less than 116ps. The accumulated clock skew from the root buffer to the 5th buffer inputs using the geometry information gives the much smaller skew of 48ps than the possible worst case skew of 73ps. It implies that the on-silicon clock skew may be less than the worst case total skew of 116ps.

Acknowledgments

The authors would like to thank M. Nagamatsu and Y. Urakawa of Toshiba Corporation for planning clock distribution and tuning methodologies in the early stage of the die development and giving us instructive advice throughout the design period.

References

[1] E. G. Friedman, "Clock Distribution Networks in VLSI Circuits and Systems," *IEEE Press*, pp.1-36, 1995.

[2] Charles F. Webb, et al., "A 400MHz S/390 Microprocessor," *IEEE ISSCC Digest of Technical Papers*, Vol.40, pp.168-169, 1997.

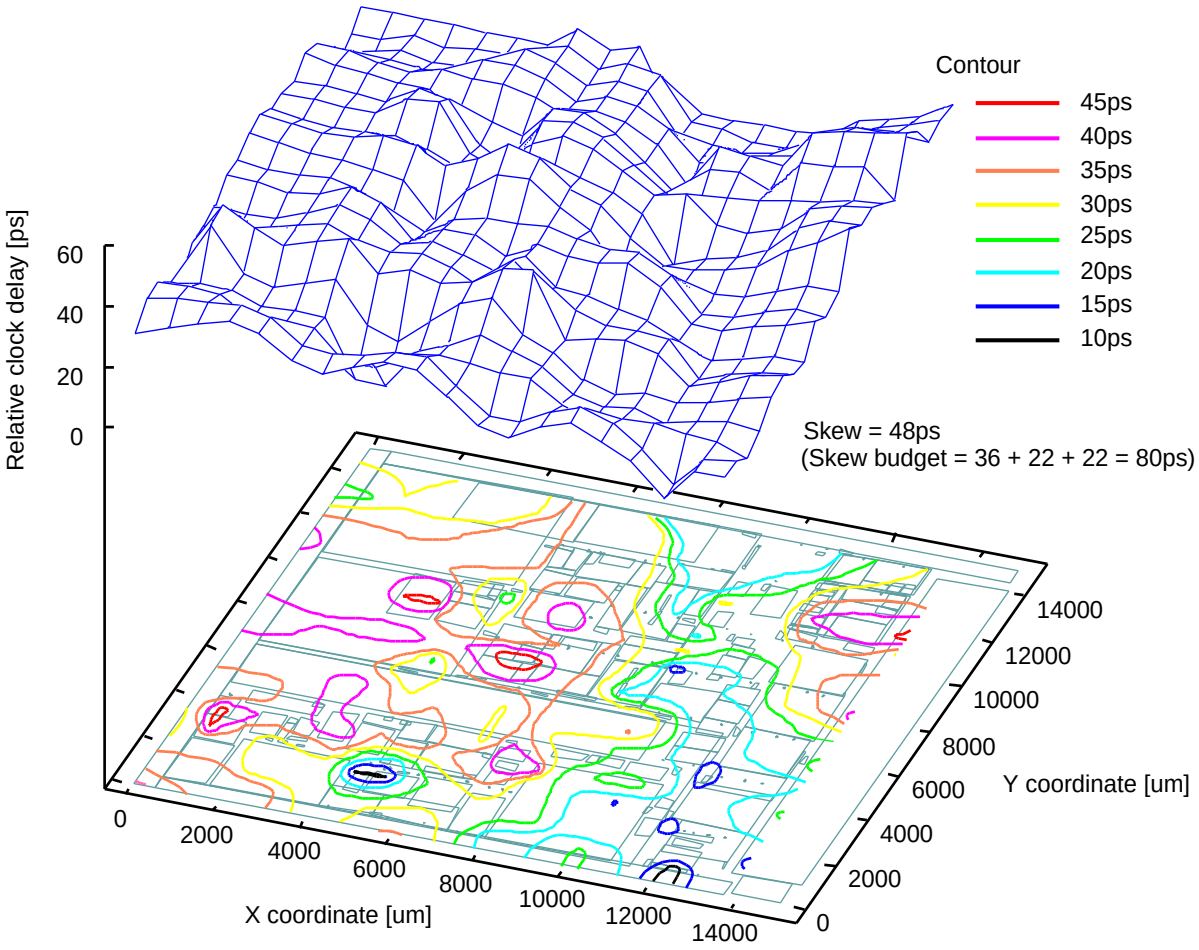

Fig. 6. Accumulated clock delay from the root buffer input to the 5th buffer inputs

[3] D. W. Dobberpuhl, et al., "A 200-MHz 64-b Dual Issue CMOS Microprocessor," *IEEE Journal of Solid State Circuits*, Vol.SC-27, No.11, pp.1555-1565, 1997.

[4] H. Fair and D. Bailey, "Clocking Design and Analysis for a 600MHz Alpha Microprocessor," *IEEE ISSCC Digest of Technical Papers*, Vol.41, pp.398-399, 1998.

[5] N. Rohrer, et al., "A 480MHz RISC Microprocessor in a 0.12um Leff CMOS Technology with Copper Interconnects," *IEEE ISSCC Digest of Technical Papers*, Vol.41, pp.240-241, 1998.

[6] K. Kutaragi, et al., "A Microprocessor with a 128b CPU, 10 Floating-Point MACs, 4 Floating-Point Dividers, and an MPEG2 Decoder," *IEEE ISSCC Digest of Technical Papers*, Vol.42, pp.256-257, 1999.

[7] F. Michael Raam, et al., "A High Bandwidth Superscalar Microprocessor for Multimedia Applications," *IEEE ISSCC Digest of Technical Papers*, Vol.42, pp.258-259, 1999.

[8] R. -S. Tsay, "Exact Zero Skew," *Proceedings ICCAD 91*, pp336-339, 1991.

[9] F. Minami, et al., "Clock tree synthesis based on RC delay balancing," *Proceedings CICC 92*, pp28.3.1-4, 1992.

[10] J. L. Neves and E. G. Friedman, "Circuit Synthesis of Clock Distribution Networks Base on Non-Zero Clock Skew," *Proceedings of IEEE International Symposium on Circuits and Systems*, 1994.

[11] R. Khanna, et al., "A 0.25mm x86 Microprocessor with a 100MHz Socket 7 Interface," *IEEE ISSCC Digest of Technical Papers*, Vol.41, pp.242-243, 1998.

[12] N. Higashi, "A 2.5GFLOPS 6.5M Polygons per Second 4-Way VLIW Geometry Processor with SIMD Instructions and a Software Bypass Mechanism," *IEEE ISSCC Digest of Technical Papers*, Vol.42, pp.226-227, 1999.