Budgeting-Free Hierarchical Design Method for Large Scale and High-Performance LSIs

Yuichi Nakamura
NEC Corp./
Waseda Univ.
1753, Shimonumabe
Nakahara-ku
Kawasaki
Japan
+81-44-431-7541

Mitsuru Tagata
NEC Software
Houriku Corp.
1, Anjo-ji
Hakusan
Ishikawa-pref.
Japan
+81-761-93-4626

Takumi Okamoto
NEC Corp.
1753, Shimonumabe
Nakahara-ku,
Kawasaki
Japan
+44-435-9486

Shigeyoshi Tawada
NEC Corp.
1-10, Nisshincho
Fuchu
Japan
+81-42-333-1313

Ko Yoshikawa
NEC Corp.
1-10, Nisshincho
Fuchu
Japan
+81-42-333-1312

{yuichi@az, m-tagata@pb, okamoto@ct, s-tawada@ax, k-yoshiawa@bx}.jp.nec.com

ABSTRACT

This paper describes a new hierarchical design method for large scale and high-performance LSIs, which eliminates the need to perform budgeting. The budgeting step in hierarchical design partitions the total propagation time constraint for a path between any two flip-flops (FFs) in different hierarchical blocks into budgets for the different segments of the path that lie within different blocks. In practice, budgeting may result in the need for additional iterations of the synthesis and physical design flow, or may achieve sub-optimal results in terms of area, power, or clock frequency. The proposed method makes the design process budgeting-free by moving the borders of the hierarchical blocks so that all borders of the hierarchical blocks are FFs. For a commercial 500MHz LSI with 141 million transistors, the design team required 2 months to archive the target frequency through try-and-try-again budgeting, while our budgeting-free method produced a design that meets the performance target within days.

Categories and Subject Descriptors

B.6.3 [**Logic Design**]; Design Aids, J6 [**Computer-Aided Engineering**]: Computer Aided Design;

General Terms

Design, Performance, Experimentation

Keywords

Hierarchical Design, Budgeting, Physical Synthesis

1. INTRODUCTION

Flat physical design flows, which consider all elements of a design at one time, are widely used in LSI backend design. The size of system LSIs is increasing according to process migration, however, and the flat physical design approach cannot handle the whole design all at once, when the LSI has tens or hundreds of millions of transistors. Two-stage hierarchical design methods based on the divide-and-conquer strategy are therefore used when designing larger LSIs. In the first stage, the physical design for each partitioned hierarchical block is performed independently, and then in the second stage, these blocks are treated as “black boxes” in the physical design process for the whole LSI.

Hierarchical design is very useful in shortening turn-around time and reducing the amount of computational resources needed for physical design, since the hierarchical block size is smaller than flat design size. Hierarchal design is very easy when the operating frequency of the LSI is relatively low but is very difficult for high-frequency LSIs because of the budgeting problem [1][2]. In general, this is the problem of dividing one clock cycle among the different segments of the path from a flip-flop (FF) in one hierarchical block to a FF in another block. Since the budgeting problem is very difficult, obtaining reasonable budgeting results generally takes a long time.

If the budgeting is not reasonable, the resultant LSI will either be unable to operate at the required frequency or will occupy an excessively large area, requiring design iterations. In this paper, instead of budgeting, we propose a budgeting-free design method for hierarchical physical design. To get results that are comparable to those obtained when optimizing the budgeting manually, but without design iterations, we introduce a method for moving the borders of the hierarchical blocks. After the border-moving, all borders of all hierarchical blocks in the resultant design are FFs and some elements that were in hierarchical blocks are outside of blocks. The simple border-moving method, however, causes many of the elements in hierarchical blocks to be moved to outside. Therefore, we additionally use a method that duplicates the wires and elements. The combined border-moving and duplication methods provide budgeting-free hierarchical design and leave as many elements as possible in their original hierarchical blocks. After the border-moving and duplication, we apply the physical design flow to each hierarchical block and the top-level design.

When we applied this methodology to three commercial chips that had over 100M transistors operated at 400–500 MHz in a few days, it took 2 months to obtain tape-out quality budgeting results by the try-and-try-again methods. The circuit area obtained before the final physical-design step was 4-5% larger than that obtained using manual try-and-try-again budgeting design. However, our methodology resulted in 2% smaller area after physical design. The reason why we can obtain a little smaller area results compared with try-and-try-again budgeting, is the reduced number of repeater buffers due to good delay constraints produced for the whole chip by our proposed method. We were thus able to design chips that operated at the target frequency and had a little smaller area compared to chips designed using manual try-and-try-again budgeting. The reduction in design time due to elimination of design iterations was around 2 months.

2. CONVENTIONAL APPROACES

The hierarchical physical design method is very useful in designing large scale LSIs, but it is necessary to budget the delay between FFs in different hierarchical blocks [1]. The simplest method [2], based on the try-and-try-again strategy, repeats budgeting and block-level and LSI-level physical design many times. Reasonable results are obtained after several physical design and timing reports. For the commercial LSIs considered in our experiments, it took 2-3 months to get reasonable results when using the try-and-try-again method.

Many methods for reducing the number of design flow iterations have been proposed, and the most typical approach is based on "prototyping" [2][3]. In the prototyping method, physical prototyping models of the LSI are constructed before real physical design. The areas and timing properties of these models are similar to those of the LSI being designed, but the models can be constructed in a shorter time than is needed for real design. The first version of budgeting is determined by analyzing the prototyping model.

Various other methods have also been proposed. One estimates the delay budget by using the characteristics of the area-delay-curve [4]. Another combines upper-bounded delay budgeting and placement-modification methods [5]. This method tries to control the budgeting by improving the placement. The method proposed in [6] inserted FFs (called "synchronizers") to improve the results of wrong budgeting by changing the timing of the designed circuit, while the method proposed in [7] transformed and partitioned hierarchical blocks to unit block size to make it easy for the budgeting to be calculated.

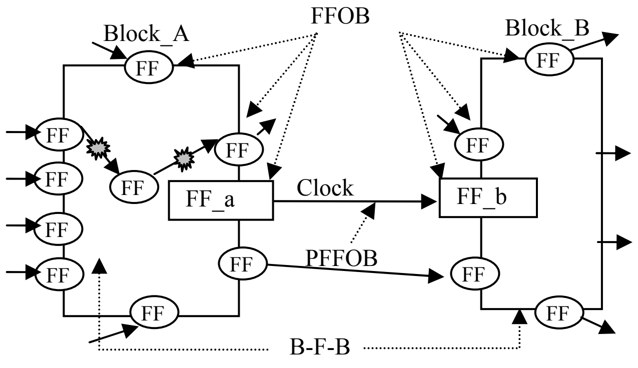

Figure 1 B-Free Design

3. BUDGETING FREE DESIGN

All conventional methods make it easy to fix the budgeting or to reduce the computation time, but they eventually need to budget the delay time between any two specified FFs. The method we propose, on the other hand, completely eliminates the need for budgeting.

To perform hierarchical physical design, we use a two-level hierarchy, consisting of a top level and a block level. In hierarchical physical design, when all inputs and outputs in all hierarchical blocks except for all primary inputs/outputs of the LSI are directly connected to FFs, budgeting is not needed because the path between two FFs that are in the different blocks goes through only the top-level block [8]. We call this design style, in which **all** the connections between hierarchical blocks are directly connected to FFs, B (Budget)-Free design (Figure 1). A hierarchal block that satisfies the B-Free design style is called a B-F-B. Moreover, we call an FF placed on the border of blocks, FFOB (FF on the border). If the design is B-Free, all hierarchical blocks are B-F-B. If a hierarchical block is B-F-B, all FFs on the border of the block are FFOBs. Also, we call the path between FFOBs a PFFOB (Path between FFOBs). The B-Free design is easily applied to physical design, since all the hierarchical blocks are designed solely by the FF-FF delay constraint. There is no constraint from input/output of the blocks to FF.

Although the B-Free design style is very useful for physical design, it is difficult to design in the B-Free style at the functional level. Functional designers, who work at the RTL (Register Transfer Level) or netlist level, can consider the function of the block (*e.g.*, controller, ALU, or sequencer). If the B-Free style, i.e. all blocks are B-F-B, was used in the design at the functional level, redundant FFs might appear in the circuit. In general, it is impossible to design a high-performance LSI when using the B-Free style at the RTL or netlist level.

We propose a border-moving method to achieve the B-Free style before layout. The main concept of this method is that the borders of all hierarchical blocks are moved from their original positions to the first FF along the path from the border. And then the elements between the original border and the new border are put into the top level. Figure 3 presents example of border moving. This LSI has two hierarchical blocks, Block_A and Block_B. To archive the B-Free style, the borders of the two blocks are moved to new positions and the new blocks, Block_A' and Block_B' are created. All inputs and outputs of the two new blocks are FF's. Thus, Block_A' and Block_B' are B-F-B. According to border moving, the combinational blocks P, Q, R, S, T, U, V, and W are moved to the top level. After the border-moving to construct a B-Free design, physical design tools can be used to determine placement and routing under the frequency constraints, which are now the same as FF-to-FF delay constraints. The timing constraints from input/output (border) of the blocks to FFs are not necessary. After the block-level physical design, the blocks are treated as black boxes and physical design tools are applied to the top level (which includes elements transferred from blocks) under the same constraints used for the block-level physical design.

4. BORDER MOVING

The border-moving operation is illustrated in Figure 2. The operation is applied to each given hierarchical block, and moves the combinational blocks, which prevent the block from being a B-F-B, to the top level. This operation is continued until the all blocks are B-F-B.

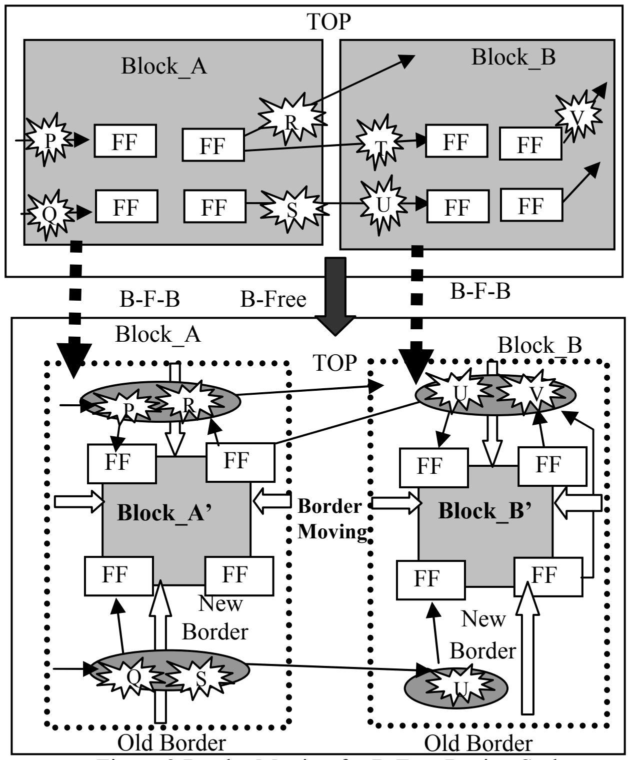

Figure 2 Border Moving for B-Free Design Style

The simple border-moving operation, however, may produce a situation in which all the hierarchical blocks are almost empty. For the designs used in our experimental results (see Chip2 on Table 2 in Section 5), only 6.9% (=0.4K (after moving)/5.8K (before moving)) of the elements originally within hierarchical blocks were still in those blocks after the first and second stages of border-moving. That is, 93.1% of the elements were moved to the top level. If all the hierarchical blocks are almost empty, the top level design is the same as the flat design.

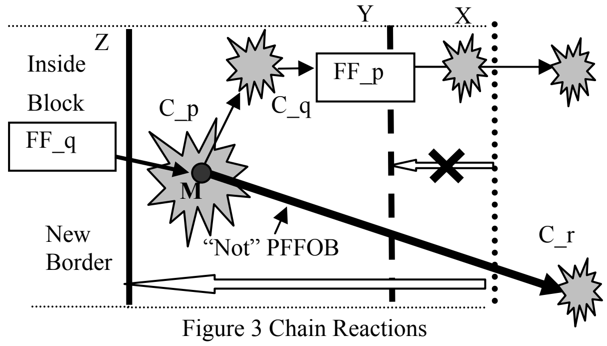

Figure 3 Chain Reactions

One of the reasons that almost all elements are removed from blocks is multiple fanouts from FFs. Figure 4 shows a situation in which the border moving operation is like a chain reaction. This block is not B-F-B with the border "X" and tried to construct B-F-B. When "Y" based on FF_p is assumed to be the candidate new border, the "not" PFFOB from FF_q , via the multiple fanout X in the combinational block C_p, to the combinational block C_r, prevents Y from actually being the new border. As a result, "Z" based on FF_q, which is a transitive fanin of FF_p, is the new border and FF_p, C_p, and C_q are moved to top level. The repetition of operations like this results in almost all elements being moved into the top level. To obtain a reasonable B-Free design, a small top level design is desirable, because of the limit on the capacity of physical design tools. When the size of the top level design is larger than the threshold (in our experience, about 10M Tr.), the physical design of the top level design cannot be performed in a day. Thus, we introduce the methods to reduce the top level area in the B-Free design style.

We introduced a step duplicating the transitive fanouts of FFs with multiple fanouts. Figure 4 shows an example of this duplication process for the circuit in Figure 3. Recall that the original block border, X, was moved to Z, in order to achieve a B-Free design style. With duplication of the circuits C_p1 and C_p2, which are duplicates of C_p, the path from FF_q to outside blocks via M and C_p2 is PFFOB. As a result, the border Z without duplication is replaced by the new border W, which leaves a larger amount of logic inside the block.

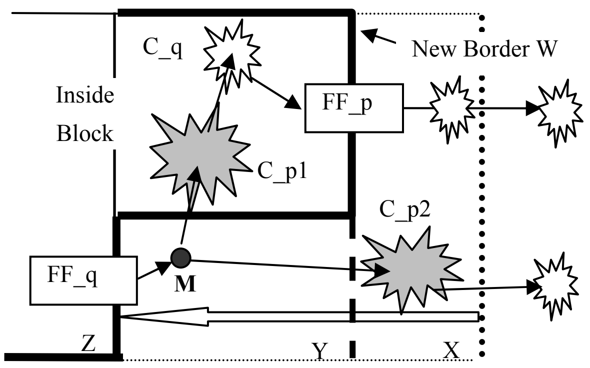

Figure 4 Duplication

Duplication is the method for recovery from multiple-fanouts-based exclusion and is effective for the reduction of the top level area. However, there may be the PFFOB in the top-level after the duplication and the border moving.

5. CASE STUDY

After the proposed border moving with duplication, the original design is transformed into a B-Free design with a set of B-F-Bs and a top level design that has as little area as possible. Physical synthesis [9] is applied to each B-F-B block under only the FF-to-FF delay constraint. Finally, physical synthesis is applied to the top level design (which includes the elements excluded from hierarchical blocks by the border–moving process), and we can obtain the target physical design without budgeting.

The total evaluation design flow is indicated in Figure 5. In the "traditional" manual budgeting flow, since the manual budgeting, logic synthesis and physical design are applied more than 15 times, it takes 0.5-2 months to obtain the reasonable results. In the case of the proposed B-Free design flow, if every path delay inside of the design of B-F-B blocks is met the given constraint, the logic and physical design is applied only "one" time.

Table 1 Chip Specification

	Chip 1	Chip 2	Chip 3
Target	CPU Chipset	CPU Chipset	CPU
Technology	90 nm	130 nm	90nm
Frequency	500 MHz	400 MHz	500MHz
# of Tr. (w/o Memory)	141M (15.2M)	32M (5.8M)	59M (23.4M)
# of blocks	11	3	26

We used the proposed method for designing 3 very large, high-performance commercial chips, the specifications of which are listed in Table 1. The experimental results achieved by the proposed methods are represented here by the values listed in Tables 2 and 3 (the numbers of transistors in the chips after various design procedures and the consequent percentage changes in circuit area). Since the area of memory is not affected by the proposed method, we focus on the logic area. It takes about 2-4 hours to apply the total proposed flow, namely B-Free with duplication for Chip 1, 2 and 3.

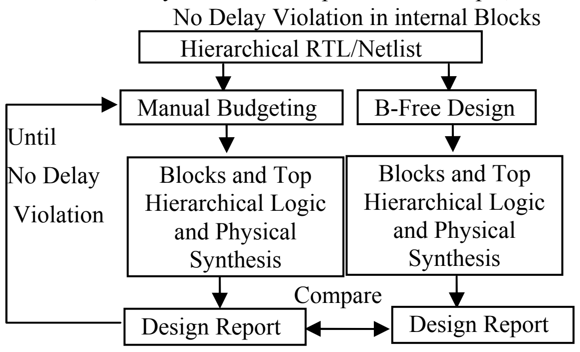
No Delay Violation in internal Blocks
Hierarchical RTL/Netlist
Manual Budgeting
B-Free Design
Until No Delay Violation
Blocks and Top Hierarchical Logic and Physical Synthesis
Blocks and Top Hierarchical Logic and Physical Synthesis
Compare
Design Report
Design Report

Figure 5 Evaluation Design Flow

Table 2 Results of Chip 1, 2 and 3
The number of Tr. without Memory

	Chip 1		Chip 2		Chip3	
	Top	Block	Top	Block	Top	Block
Original Manual	148K 0.7%	15.2M 99.3%	1K 0.1%	5.8M 99.9%	2.1M 9%	21.3M 89%
B-F w/o Dup	13.2M **87%**	2M 13%	5.4M **93%**	0.4M 7%	19.9M **85%**	3.5M 15%
B-F w/ Dup.	3.4M **21%**	12.8M 79%	1.5M **25%**	4.5M 75%	4.0M **17%**	20.6M 83%

The top level design of the B-Free design obtained by simple border-moving is very large, and hence cannot be subject to physical design. For example, in table 1, the top level area of Chip 3 has 19.9MTr. Thus, the physical design tool [9] cannot finish in one month. After applying the duplication, however, the area of the top level decreased to 21%, 25%, and 17% in Chip 1, 2 and 3 from 87%, 92%, and 85%, respectively. The increase of total area is only 5.3%, 4.0% and 1% for Chip1, 2 and 3. Although the duplication process increased the total area of the chips, the rate of increase is very small.

The results of applying the total design flow (including physical design) are shown in Table 3. Since the original flow requires much iteration for determining a high-quality budgeting, it takes about 2 months to archive the target frequency. In contrast with the original flow, the proposed method finishes without try-and-try-again attempts, and the design was completed in several days and achieved the target frequency.

Table 3 After P&R Results (#of Tr. w/o Memory**)**

	Chip 1	Chip 2	Chip3
Manual Before P&R	15.4M	5.8M	23.4M
B-Free Before P&R	16.2M	6.0M	24.6M
Manual After P&R	17.1M	6.8M	25.3M
B-Free After P&R	16.6M	6.6M	24.9M

In the results of Table3, the area before P&R by the original flow is smaller than the proposed flow, since the proposed flow involves logic duplication. However, after P&R, the area obtained by the proposed flow is a little smaller than the original flow. In the case of Chip1, the original flow obtains a design that is 0.8M Tr. smaller before P&R and 0.5M Tr. larger after P&R than the proposed flow. The major reason why the proposed flow can obtain smaller results after P&R is the difference in the number of repeater buffers and inverters. The number of repeater buffers and inverters (which are used in delay optimizations) in the results of the proposed flow is 226K (1.2M Tr.) smaller than of the number required for the original flow in the case Chip1 (Table 4). This result says that our proposed flow can provide better delay constraint for the design than the original.

Table 4 The number of Large Size Buffers/Inverters

	Chip1	Chip2	Chip3
Original Manual	300K	184K	358K
Proposed B-Free	54K	97K	49K
Diff{ (Org.)- (B-F)} #of Tr.	1.2M Tr.	0.4M Tr.	1.45M Tr.

6. Conclusion

We proposed a budgeting-free hierarchical design method for high-performance LSIs. To eliminate the need for budgeting, we convert the original design to one in a B-Free design style in which the inputs and outputs of all hierarchical blocks are connected directly to FFs. Since the simple border-moving method used to generate the B-F-B version of the design yields hierarchical blocks that are almost empty, we additionally use duplication methods. Experimental results for 3 commercial chips showed that before physical design the B-Free style circuit that would be produced by our budgeting-free method was 4-5% larger than the original design, which was designed by 2 months of try-and-try-again budgeting, but that after physical design it was a little, 1.6-3% smaller. These results show that our method can reduce 2 months of the design period while resulting in comparable if not better quality chips.

References

[1] A. Mehrotra, L. van Ginneken, and Y. Trivedi, “Design flow and methodology for 50M gate ASIC,” ASP-DAC, Jan. 2003, pp. 640–645.

[2] W.-J. Dai, “Hierarchical physical design methodology for multi-million gate chips,” ISPD April 2001, pp. 179–181.

[3] J. Koehl, D.E. Lackey, and G. Doerre, “IBM's 50 million gate ASICs,” ASP-DAC, Jan. 2003, pp. 628–634.

[4] C.-C. Kuo and A. C-H. Wu, “Delay budgeting for a timing-closure-driven design method,” ICCAD, Nov. 2000, pp. 202–207.

[5] M. Sarrafzadeh, D. Knol, and G. Tellez, “A delay budgeting algorithm ensuring maximum flexibility in placement,” IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, vol. 16, no. 11, pp. 1332–1341, Nov 1997.

[6] A. Edman and C. Svensson, "Timing closure through a globally synchronous, timing partitioned design methodology," DAC, July 2004, pp. 71–74.

[7] F. Mo and R. K. Brayton, "A timing-driven module-based chip design flow," DAC, Jul, 2004, pp. 67–70.

[8] M. Keating and P. Bricaund, Reuse Methodology Manual for System-On-A-Chip Designs, 2nd Edition, Kluwer Academic Publishers, 1999, pp. 114–115.

[9] Synopsys Physical Compiler, http://www.synopsys.com