Challenges to Data-Path Physical Design Inside SOC*

Jing Tong, Hong Xianlong, Cai Yici, Xu Jingyu, Yang Changqi,
Zhang Yiqian, Zhou Qiang and Wu Weimin

(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*)

Abstract: Previously, a single data-path stack was adequate for data-path chips, and the complexity and size of the data-path was comparatively small. As current data-path chips, such as system-on-a-chip (SOC), become more complex, multiple data-path stacks are required to implement the entire data-path. As more data-path stacks are integrated into SOC, data-path is becoming a critical part of the whole giga-scale integrated circuits (GSI) design. The traditional physical design methodology can not satisfy the data-path performance requirements, because it can not accommodate the data-path bit-sliced structure and the strict performance (such as timing, coupling, and crosstalk) constraints. Challenges in the data-path physical design are addressed. The fundamental problems and key technologies in data-path physical design are analysed. The corresponding researches and solutions in this research field are also discussed.

Key words: physical design; data-path; bit-sliced structure; system-on-a-chip; giga-scale integrated circuits; very-deep-submicron

EEACC: 2570

CLC number: TN47　　**Document code**: A　　**Article ID**: 0253-4177(2002)08-0785-09

1 Introduction

In recent years, the VLSI/ULSI (very/ultra large scale integrated circuits) technology has profoundly advanced[1,2]. The driving force behind the spectacular advancement of integrated circuit technology in the past thirty years has been exponentially reduced in scaling of feature size. It has been following Moore's Law[3] at the rate of a factor of 0.7 reduction every three years. It is expected that such exponential scaling will continue for at least another 10 to 12 years as projected in the 1997 National Technology Roadmap for Semiconductors (NTRS'97)[4].

With progress in giga-scale integrated circuits (GSI) and very-deep-submicron (VDSM) technology, we can design a single chip with large overall dimension but smaller feature size and wire space. We also can design it with more and more functions and many more transistors. Table 1 shows the latest technology roadmap trend for the semiconductor[5]. Meanwhile, Intel Corportion has designed a 0.09μm process desktop Pentium 4 processor, code-named Prescott, which is slated for introduction in the second half of 2003[6]. All of these advances enable system-on-a-chip (SOC) integration instead of system-on-a-board (SOB) integration[7].

* Project supported partly by Outstanding Faculty Support Project of Tsinghua University (No.[2002]4) and State Key Development Program for Basic Research of China (No. G-1998030403)

Jing Tong　male, was born in 1966, PhD, associate professor. His research interest is performance-driven layout.

Hong Xianlong　male, was born in 1940, professor. His research interests include layout algorithms and systems.

Cai Yici　female, was born in 1960, associate professor. Her research interest focuses on VLSI layout.

Received 13 May 2002

Table 1 2001~ 2016 technology roadmap trend for the semiconductor

Production year	2001	2002	2003	2004	2005	2006	2007	2010	2013	2016
DRAM half-pitch/nm	130	115	100	90	80	70	65	45	32	22
MPU/ASIC half-pitch/nm	150	130	107	90	80	70	65	45	32	22
MPU printed gate length/nm	90	75	65	53	45	40	35	25	18	13
MPU physical gate length/nm	65	53	45	37	32	28	25	18	13	9
On-chip local clock/MHz	1,684	2,317	3,088	3,990	5,173	5,631	6,739	11,511	19,348	28,751
Maximum number wiring levels	7	8	8	8	9	9	9	10	10	10

In the physical design of SOC, many new problems need to be solved. One of the most difficult is data-path physical design. In typical state-of-the-art microprocessor designs, data-paths comprise about 70% of the logic (excluding caches)[8] and occupy as much as 30% to 60% of the silicon area[9,10]. However, data-path design is typically done manually, and is often done in custom style[11]. Traditional place and route (P&R) tools are incapable of using this regularity to produce more compact designs that give higher performance[12]. The layout results are often inferior to those designed manually. But manual design is slow. Meanwhile, high performance data-path design is still very time-consuming. Thus, We will face great challenges in data-path physical design.

Data-path physical design becomes a field of active research. Many research groups in universities, institutes, especially in electronic industries, are doing in-depth studies in this research field. But, there is a scattered literature in the area of data-path physical design. This paper focuses on precisely this problem. The remainder of this paper is organized as follows. In Section 2, the repeating bit-sliced structure is introduced. In section 3, we intend to briefly survey them. Then, we will give some detailed analyses and suggestions.

2 Repeating bit-sliced structure

Informally, data-paths are circuits where the same or similar logic is applied to several bits of a bus[13]. Data-path stack (or simple stack, bit-stack[14,15]) is a common structure for data-path layout inside SOC. A data-path stack is made up of many custom word lines (or data-path macros[14,15], functional building blocks, FBBs[16]), such as registers, ALUs, adders, shifters, multiplexers, buffers, comparators to from the data flow of the functional units such as the fixed-point, floating-point execution units, the fetch unit, and the decode unit in a microprocessor.

A word line (WL) is made up of repeating “bit-slices”[17~19] or “bit-cells”[14,15] each performing its required function on a single bit of the data flow, plus a small control cell connected to the control logic external to the data-path.

Figure 1 shows the regularity placement and routing of a data-path circuit. Figure 2 shows the schematic layout of data-path and the detailed view of a bit-cell. There are two groups of interconnects

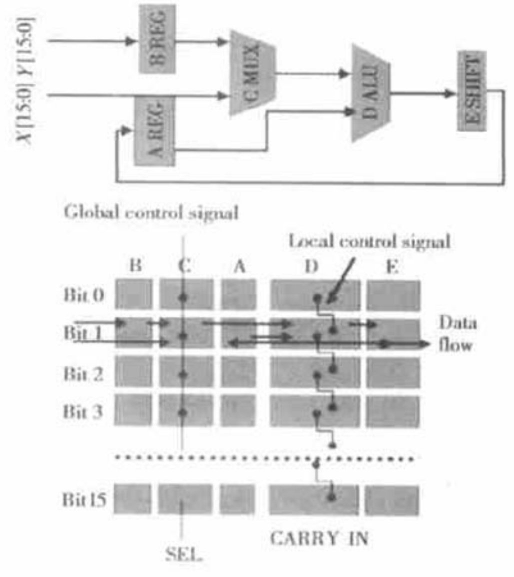

Fig. 1 Regularity placement and routing of a data-path circuit[17]

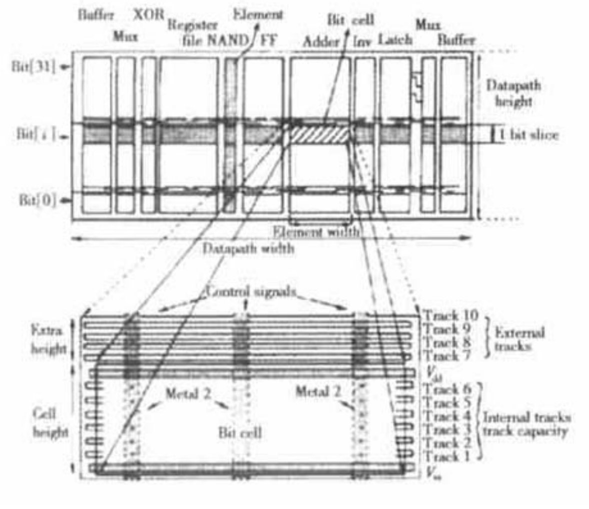

Fig. 2　Schematic layout of data-path and the detailed view of bit-cell[18]

flowing perpendicular to each other, as shown in Fig. 1[17]. One is the data flow, which runs horizontally along parallel wires. The other is the control flow, which goes vertically. The control flow can either be global control signals, which operate on every bit simultaneously (e. g. the CLOCK signal), or local control signals, which operate on adjacent bits (e. g. the CARRY-IN/OUT).

Figure 3 shows the bit-sliced abstractions of a data-path circuit and their circuit block configuration. The bit-sliced abstraction of a data-path circuit (called APM (abstract physical model) in Ref. [17]) consists of circuit blocks corresponding to one-bit operations in the data-path. The circuit block is the basic building unit of APM. Either it can be a standard cell from the library, e. g. AND, OR, XOR, DEF, Full Adder, or it can be a data-path leaf cell or a group of standard cells, e. g. 1-bit multiplier, Booth MUX. In APM, each circuit block is represented as a rectangular box, with wires indicating its input and output interconnections. The height of the box is the height of corresponding standard cell or data-path leaf cell, and the width is the cell width or the sum of widths of the cells in the block.

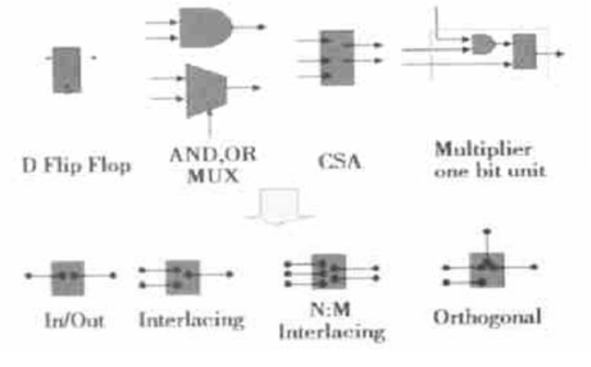

Fig. 3　Bit-sliced abstractions of a data-path circuit and their circuit block configuration[17]

It is not necessary that all the WLs have the same number of bit-cells. The number of bit-cells depends on the width of the data-path, typically 16, 32, 48, or 64bit. The majority of the data flow goes from one WL to another along the same bit, except for a very small portion of data flow, called the multi-bit data signals, which span more than one bit.

3　Survey and analysis

Data-path stacks inside SOC have certain special properties. As fabrication technology moves into VDSM feature size and giga-hertz clock frequencies, timing, coupling, and crosstalk become increasingly dominant factors in data-path layout. The majority of the data flows goes from one WL to another along the same bit, except for a very small portion of data flow that spans more than one bit. Repeating "bit-slices"/"bit-cells" are in the data-path. Thus, merely minimizing congestion and chip size is not adequate. We urgently need efficient performance-optimizing algorithms for data-path layout design. Meanwhile, bus-net routing becomes more important and structure regularity should be well taken into account in layout algorithm design.

In this section, based on the data-path physical design goal and physical design flow, we will discuss key technologies throughout data-path physical design process.

3. 1　Data-path physical design goal

The data-path physical design goal includes the following items.

(1) The quality of compiled data-path layout should be close to that of the handcrafted layout.

(2) Performance (such as timing, coupling[20], and crosstalk) is the first priority for optimization.

(3) Layout generation should be in the form of bit-sliced, regular designs.

(4) Designs are usually based on standard cell (SC) with a few macro cells.

3.2 Chip partition

Previously, a single data-path stack was adequate for data-path chips, and the complexity and size of the data-path was comparatively small. As current data-path chips, such as SOC, become more complex, multiple data-path stacks are required to implement the entire data-path. Then, we need to solve the problem of how to partition chips.

Figure 4 shows the partition of a multi-stack data-path chip, which includes data-path stacks, control logic, chip drivers, on-chip memory, and random logic. References[14, 15] and [21] present useful algorithms for chip partition.

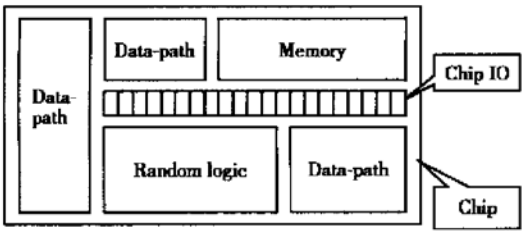

Fig.4 Partition of a multi-stack data-path chip

3.3 Floorplanning

The floorplanning algorithms can be generally classified into the following three categories.

3.3.1 Among all stacks

For these floorplanning algorithms, Refs.[14, 15] use the enumerated method and Refs.[22, 23] use the mixed mode method.

3.3.2 Inside a data-path stack

Since a data-path stack may contain macro cells besides common WLs, it may need a floorplanning algorithm inside a data-path stack. Meanwhile, to meet the needs of size or/and shape of a data-path stack, we may use the floorplanning method to optimize the area and shape ratio of the data-path stack.

3.3.3 Inside a bit-cell

Based on the O-tree algorithm[24, 25], Ref.[26] described a floorplanning algorithm inside a bit-cell.

3.4 Placement

Data-path with great regularity has the so called bit-sliced structure. Data-path is usually based on SC so that the bit-slices will be mapped into the rows of SC placement. This will guide the placement process. Here, we present the key strategies for data-path placement.

(1) Extract the structure regularity information from netlist. Extracting regularity information is equal to assigning each cell to one unique pair of bit-slice and WL.

(2) Assign rows for data-path cells. Row assignment will follow the order of bits extracted from netlist as closely as possible. The objective is to minimize the weighed wire length of nets across rows.

(3) Determine the inner-row order for each cell in that row. Inner-row ordering is similar to the one-dimensional element ordering problem[27, 28] which has been proved to be NP-hard[29]. The objective is to minimize weighed wire length of nets and minimize track congestion.

(4) Maintain the control signal alignment.

(5) Design timing and coupling driven placement algorithms. Determining net weight should take the timing and coupling effects into account.

The hybrid approach of genetic algorithm (GA) and simulated annealing (SA) is used for data-path placement in Ref.[18]. It can get a good solution. But it is time-consuming. New, fast and high performance placement algorithms need to be developed. References[30~ 32] presented transistor-level algorithms for data-path placement. The testing cases are on a very small scale. Reference [33] presented an algorithm for eliminating the net congestion of data-path chips.

The available data-path tools for placement can be fundamentally classed into three categories: generation tools (GT), synthesis tools (ST), and extraction tools (ET)[12]. Among them, ET can meet the needs of GSI progress. However, it is difficult to design an algorithm for extracting structure regularity information. References[17, 34] introduced such algorithms, but these algorithms rely on the synthesis results. Key features of the methodology in Ref. [35] are automatic extraction of regular structures and utilization of regularity even after extensive logic optimization, which is a useful algorithm.

Most inner-row ordering algorithms use heuristics, except for very small problems. Approaches using these algorithms include integer programming[32], the constructive method[27], the mini-cut[36, 37], the branch-and-bound search[16, 37], and the analytic method[21].

3.5 Routing

Routing for data-path is different from that for SC. One of the essential differences between them is bus-net routing. Therefore, data-path routing includes two aspects: signal-net routing and bus-net routing.

3.5.1 Signal-net routing

Signal-net routing is similar to SC routing. It has two routing stages: global routing and detailed routing.

(1) In signal-net global routing, we have the following optimizing objectives.

1) Congestion optimizing

2) Timing slack constraints

3) Shielding requirements of critical nets

4) Straight routing requirements of critical nets

In Refs. [14, 15], a maze runner-type of wiring tool was used to wire the chip. It is simple and easy to implement. But it has poor routing capability. In Ref. [19], a channel routing method was proposed to route the signal-net. If congestion occurs, the bit-slices are stretched. This method leads to a larger data-path area.

A congestion optimization algorithm for a four-layer data-path routing was presented in Ref. [38]. It gets a good routing result. However, it has some shortcomings: The special "pinrail" mechanism limits the algorithm application. It takes a long running time. Only a few small-scale artificial cases were tested.

Here, we greatly emphasize timing optimizing. Timing model and timing optimizing strategy are two important aspects of the optimization problem. The Elmore delay model[39] and the Sakurai delay model[40] are two kinds of timing models widely used in timing-driven routing[41~43]. However, to get more accurate delay value, we should use the table-lookup delay model[44] to calculate the transition and corresponding delay. Some useful researches[45~47] using the table-lookup delay model have been done on SC placement. Efficient timing optimizing strategy is also very important. Reference[48] presented an efficient algorithm to optimize timing in SC global routing.

(2) Signal-net detailed routing follows track assignment. This type of routing meets the technical needs of chip manufacturing.

1) If there is a via at the end of a wire, an additional short wire segment (like a_1 and a_2 in Fig. 5 (a)) will be needed. In Fig. 5(b), the wire space has to be enlarged because two additional short wire segments are needed at the same position. To avoid this case, these two vias need to be displaced, as showed in Fig. 5(c).

2) When a wire in the first layer is connected to a wire in the third layer, two vias will be needed. There should be a "ladder construction" in the second layer.

A fast signal-net gridless detailed router was presented in Ref. [8]. The goal of this detailed routing algorithm is to tackle very large industrial data-paths.

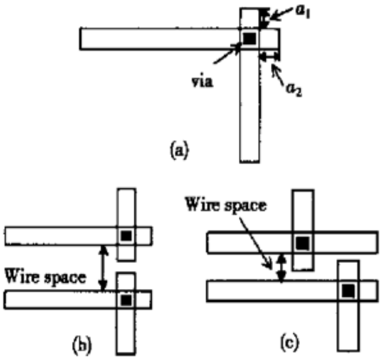

Fig. 5 Special operations in detailed routing

3.5.2 Bus-net routing

Bus-net routing can be done in two different ways: as special nets to be pre-routed before signal-net global routing, or as an independent group to be routed simultaneously with signal nets in the process of global routing. The former is easy to implement but the routing result is not optimal. The latter can get an optimal routing result but may be difficult to develop. However, to get good routing results, the latter is worthy of study.

In order to keep the timing balance (zero skew), we should use the same routing pattern for the nets in the same bus group. Crosstalk also should be considered throughout this process. Based on a simplified crosstalk model, we estimate the crosstalk risk for bus routing and give it a suitable number of routing tracks.

A bus routing algorithm in the process of detailed routing is proposed in Ref. [49], which is based on line-probe algorithm with gird. We may use the line-probe algorithm as the fundamental method for bus-net routing.

3.6 Other layout items

3.6.1 Resources estimation

Resources estimation follows placement, which includes the following functions:

(1) Create global routing graph (GRG). The size of global routing cell (GRC) is adjustable. Too large or too small capacity of the GRG edge will affect the routing performance.

(2) Route short nets on metal one (M1) layer.

(3) Estimate track capacity of the GRC. Give suitable capacity constraints as the routing resources except M1.

(4) Evaluate the placement result and judge whether to accept the result or not, which can avoid useless succeeding works (e. g., detailed routing).

In general, we need to pre-route special nets before resources estimation. The special nets routing includes power/ground (P/G) nets routing and clock nets routing.

3.6.2 Layer assignment

Layer assignment is a very useful step to reduce coupling. Before signal-net detailed routing, we should take this step to optimize performance. The goal of this step includes many aspects, such as reducing crosstalk, even distributing net wires, and minimizing wire length and via number. There are some noticeable strategies in layer assignment as follows.

(1) Coupling among wires in the same layer is much stronger than in every other layer.

(2) Do not assign too many long nets to one layer-pair.

(3) Assign short nets and long delay permitting nets to relatively lower layer-pair. Assign critical and long nets to higher layer-pair if possible.

(4) Assign one net to one layer-pair as possible.

(5) Reduce the number of higher layer-pairs in order to keep chip size small.

3.6.3 Track assignment

After layer assignment, the wires will be placed on tracks——track assignment. In the process of track assignment, the available tracks inside each bit-slice are assigned to wire segments for each net, as shown in Fig. 6.

Track assignment in data-path layout is similar to the cross point assignment[50] in SC layout. The difference between them is that track assignment is inside bit-slices. Some useful researches are

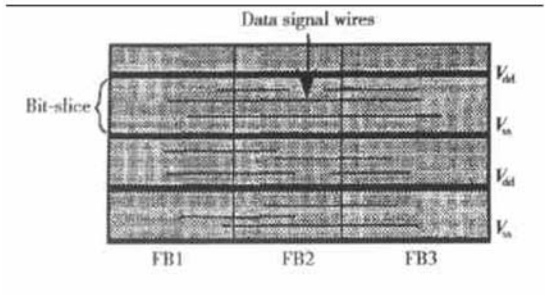

Fig. 6 Tracks inside the bit-slice[19]

reported in Refs. [18, 19]. Track assignment algorithms should take the following factors into account.

(1) Minimize the track density inside a bit-slice. The maximum track density in the track density profile determines the data-path height. Figure 7 shows the effect of track assignment on track density. If the track density[19] exceeds the track capacity[19], extra tracks are required on top of the internal tracks. This contributes to the increase of the data-path height and the chip area.

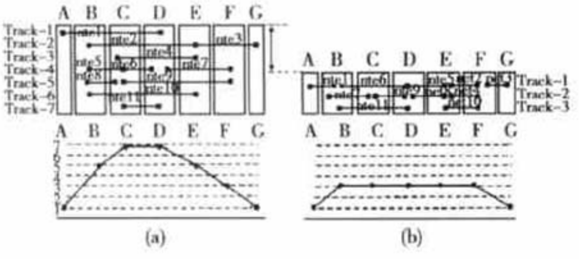

Fig. 7 Effect of track assignment on track density[18]

(2) To minimize the total wire length, we should consider the topology of the net when we assign a track to one of its wire segment. For example, the track assignment shown in Fig. 8(a) is better than that shown in Fig. 8(b).

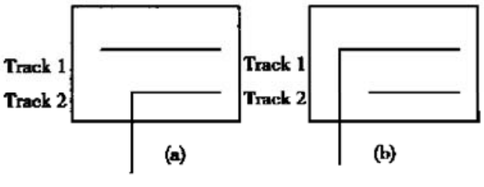

Fig. 8 Comparison on track assignment

(3) To minimize the total wire length and the number of vias, we should stretch long wires as straight as possible.

(4) To keep the timing balance (zero skew), we should use the same routing pattern for the nets in the same bus group. Thus, we also need to consider this case in the process of track assignment.

(5) Reduce crosstalk risk by minimizing the overlap-length between two adjacent wires. Or, not route two long parallel wires in two adjacent tracks.

3.6.4 ECO(engineering change order)

ECO is very useful for designers to improve their designs for data-path layout. It provides some special commands for chip designers to do ECO tasks. By using these special commands, the designers can improve their local design from their knowledge and experience. For example, they may adopt buffer inserting[51] to optimize wire timing, or change the position of some cells to obtain required space, or route wires for critical nets to meet shielding requirements.

Acknowledgements The authors wish to thank graduate students in EDA group, Tsinghua University who have contributed to this paper: Zhou Yi, Hu Yu, Hou Wenting, Zhang Ling, Yang Liu, Yao Hailong, and Cai Qi.

References

[1] Hong X L, Yan X L, Qiao C G. The theories and algorithms for VLSI layout design. Beijing: Science Press, 1998 (in Chinese) [洪先龙, 严晓浪, 乔长阁. 超大规模集成电路布图理论与算法. 北京: 科学出版社, 1998]

[2] Jing T, Hong X L, Cai Y C, et al. The key technologies and related research work of performance-driven global routing. Journal of Software, 2001, 12(5): 677(in Chinese) [经彤, 洪先龙, 蔡懿慈, 等. 性能驱动总体布线的关键技术及研究进展. 软件学报, 2001, 12(5): 677]

[3] Moore G E. Cramming more components onto integrated circuits. Electronics Magazine, 1965, 38(4): 114

[4] Semiconductor Industry Association. The national technology roadmap for semiconductors. USA, 1997

[5] Allan A, Edenfeld D, Joyner W H Jr, et al. 2001 technology roadmap for semiconductors. IEEE Trans Comput, 2002, 35 (1): 42

[6] Robertson J. Intel discusses "Prescott" microprocessor. EE-Times, February 20, 2002

[7] Hong X L, Jing T, Cai Y C. The new design concept of electronic system design: system-on-a-chip. Global Electronic Components, 2002, 81: 24(in Chinese)[洪先龙, 经彤, 蔡懿慈. 电子系统设计的新概念——系统级芯片. 世界电子元器件, 2002, 81: 24]

[8] Das S, Khatri S P. A regularity-driven fast gridless detailed router for high frequency datapath designs. In: Proc ACM ISPD, Sonoma, USA, 2001: 130

[9] Kyung C M. HK386, an x86-compatible 32 bit CISC microprocessor. In: Proc IEEE/ACM ASP-DAC, 1997: 661

[10] Yim J S, Hwang Y H, Park C J, et al. A C-based RTL design verification methodology for complex microprocessor. In: Proc ACM/IEEE DAC, 1997: 83

[11] Nemani M, Tiwari V. Macro-driven circuit design methodology for high-performance datapaths. In: Proc ACM/IEEE DAC, Los Angeles, USA, 2000

[12] Ienne P, Griebing A. Practical experiences with standard cell based datapath design tools. In: Proc ACM/IEEE DAC, San Francisco, USA, 1998: 396

[13] Weste N H E, Eshraghian K. Principles of CMOS VLSI design, VLSI system series. Addison-Wesley, Second edition, 1993

[14] Luk W K, Dean A A. Multistack optimization for data-path chip layout. IEEE Trans CAD, 1991, 10(1): 116

[15] Luk W K, Dean A A, Mathews J W. Multi-terrain partitioning and floor-planning for data-path chip (microprocessor) layout. In: Proc IEEE/ACM ICCAD, Santa Clara, USA, 1989: 492

[16] Cai H, Note S, Six P, et al. A data path layout assembler for high performance DSP circuits. In: Proc ACM/IEEE DAC, Orlando, USA, 1990: 306

[17] Ye T T, Micheli G D. Data path placement with regularity. In: Proc IEEE/ACM ICCAD, San Jose, USA, 2000: 264

[18] Yim J S, Kyung C M. Datapath layout optimization using genetic algorithm and simulated annealing. IEE Comput Digit Tech, 1998, 145(2): 135

[19] Nakao H, Kitada O, Hayashikoshi M, et al. A high density datapath layout generation method under path delay constraints. In: Proc IEEE CICC, San Diego, USA, 1993: 9. 5. 1

[20] Yim J S, Kyung C M. Reducing cross-coupling among interconnect wires in deep-submicro datapath design. In: Proc ACM/IEEE DAC, New Orleans, USA, 1999: 485

[21] Riess B M, Doll K, Johannes F M. Partitioning very large circuits using analytical placement techniques. In: Proc ACM/IEEE DAC, 1994: 646

[22] Yu H, Hong X L, et al. MMP: a novel placement algorithm for combined macro blocks and standard cell layout design. In: Proc IEEE/ACM ASP-DAC, Yokohama, Japan, 2000: 271

[23] Yu H, Hong X L, Cai Y C, et al. A novel algorithm for placement of mixed macro block and standard cell designs. Acta Electronica Sinica, 2000, 28(5): 1(in Chinese)[于泓, 洪先龙, 蔡懿慈, 等. 一种新型宏模块和标准单元的混合模式布局算法. 电子学报, 2000, 28(5): 1]

[24] Guo P N, Cheng C K, Yoshimura T. An O-tree representation of non-slicing floorplan and its application. In: Proc ACM/IEEE DAC, New Orleans, USA, 1999: 268

[25] Pang Y, Cheng C K, Yoshimura T. An enhanced perturbing algorithm for floorplan design using the O-tree representation. In: Proc ACM ISPD, San Diego, USA, 2000: 168

[26] Serdar T, Sechen C. Automation datapath tile placement and routing. In: Proc IEEE/ACM ICCAD, San Jose, USA, 2001: 552

[27] Kang S M. Linear ordering and application to placement. In: Proc ACM/IEEE DAC, 1983: 457

[28] Hong Y S, Park K H, Kim M. A heuristic approach for ordering the columns in one-dimensional logic arrays. IEEE Trans CAD, 1989, 8(5): 547

[29] Garey M R, Johnson D S. Computers and intractability: a guide to the theory of NP-Completeness. San Francisco: W. H. Freeman and Co, 1979

[30] Askar S, Ciesielski M. Analytical approach to custom datapath design. In: Proc IEEE/ACM ICCAD, San Jose, USA, 1999

[31] Serdar T, Sechen C. AKORD: transistor level and mixed transistor/gate level placement tool for digital data paths. In: Proc IEEE/ACM ICCAD, San Jose, USA, 1999

[32] Kim J, Kang S M. A timing-driven data path layout synthesis with integer programming. In: Proc IEEE/ACM ICCAD, 1995: 716

[33] Wen H, Tang P S. Net congestion elimination for datapaths by placement refinement. Chinese Journal of Semiconductors, 2000, 21(4): 325(in Chinese)[文化, 唐璞山. 采用布局精细优化来消除 data-path 的线网拥挤. 半导体学报, 2000, 21(4): 325]

[34] Chowdhary A, Kale S, Saripella P. A general approach for regularity extraction in data-path circuits. In: Proc IEEE/ACM ICCAD, 1998

[35] Nijssen R X T, Van Eijk C A J. Greyhound: a methodology for utilizing data-path regularity in standard design flows. Integration, the VLSI J, 1998, 25: 111

[36] Kernighan B W, Lin S. An efficient heuristic procedure for partitioning graphs. Bell Syst Tech J, 1970, 49(2): 291

[37] Auer E, Schiele S, Sigl G. A new linear placement algorithm for cell generation. In: Proc IEEE/ACM ICCAD, 1991: 486

[38] Raman S, Sapatnekar S S, Alpert C J. Datapath routing based on a decongestion metric. In: Proc ACM ISPD, San Diego, USA, 2000, 122

[39] Elmore W C. The transient response of lumped linear networks with particular regard to wideband amplifiers. J Appl Phys, 1948, 19(1): 55

[40] Sakurai T. Approximation of wiring delay in MOSFET LSI.

IEEE J Solid-State Circuits, 1983, 18(4): 418

[41] Hong X L, Xue T X, Kuh E S, et al. TIGER: an efficient timing-driven global router for gate array and standard cell layout design. IEEE Trans CAD, 1997, 16(11): 1323

[42] Bao H Y, Hong X L, Cai Y C. Timing-driven Steiner tree algorithm based on Sakurai model. Chinese Journal of Semiconductors, 1999, 20(1): 41(in Chinese)[鲍海云, 洪先龙, 蔡懿慈, 等. 基于 Sakurai 模型的时延驱动 Steiner 树算法. 半导体学报, 1999, 20(1): 41]

[43] Xu J Y, Hong X L, Jing T, et al. An efficient hierarchical timing-driven Steiner tree algorithm for global routing. In: Proc IEEE/ACM ASP-DAC, Bangalore, India, 2002: 473

[44] Lillis J, Buch P. Table-lookup methods for improved performance-driven routing. In: Proc ACM/IEEE DAC, San Francisco, USA, 1998: 368

[45] Yao B, Hou W T, et al. Timing-driven detailed placement for standard-cells based on lookup-table delay model. In: Proc World Computer Congress—Conference on Chip Design Automation, Beijing, 2000: 305

[46] Yu H, Hong X L, et al. A new timing-driven placement algorithm based on lookup-table model. Chinese Journal of Semiconductors, 2000, 21(11): 1129

[47] Hou W T, Hong X L, et al. Methods of improving rout ability in timing-driven placement. In: Proc IEEE ASICON, Shanghai, 2001: 110

[48] Jing T, Hong X L, Bao H Y, et al. A novel and efficient timing-driven global router for standard cell layout design based on critical network concept. In: Proc IEEE ISCAS, Scottsdale, USA, 2002

[49] Zhang Y Q, Zhang Y, Cai Y C, et al. An algorithm for bus routing based on line-probe. Microelectronics, 2000, S0: 13(in Chinese)[张轶谦, 张雁, 蔡懿慈, 等. 一种基于线探索的 BUS 线布线算法. 微电子学, 2000 年, 增刊: 13]

[50] Huang S J, Hong X L, Cai Y C, et al. Parallel cross point assignment algorithm with net priority. Microelectronics, 2000, S0: 28(in Chinese)[黄松珏, 洪先龙, 蔡懿慈, 等. 带线网优先级分类的并行过点分配算法. 微电子学, 2000 年, 增刊: 28]

[51] Alpert C J, Hrkic M, Hu J, et al. Buffered steiner trees for difficult instances. In: Proc ACM ISPD, Sonoma, USA, 2001: 4

SOC 中 Data-Path 布图设计面临的挑战*

经 彤 洪先龙 蔡懿慈 许静宇 杨长旗 张轶谦 周 强 吴为民

(清华大学计算机科学与技术系, 北京 100084)

摘要: 目前所设计的系统级芯片 (SOC) 包含有多个 data-path 模块, 这使得 data-path 成为整个 G 大规模集成电路 (GSI) 设计中最关键的部分. 以往的布图理论及算法在许多方面已不能满足 data-path 布图设计的需要, 这主要是由于传统的布图工具没有考虑 data-path 所特有的电路结构特点. Data-path 具有规整的位片结构, 具有很高的性能指标要求, 如对于时延、耦合效应和串扰等性能都有严格的要求. 此外, data-path 中还存在大量成束状结构的 BUS 线网. 文中提出了 data-path 布图设计所面临的挑战. 从介绍 data-path 布图的基本问题入手, 重点分析了 data-path 布图设计中的关键技术, 并在讨论已有研究工作的基础上针对不同的布图阶段提出了可行的技术路线与设想.

关键词: 布图设计; data-path; 位片结构; 系统级芯片; G 大规模集成电路; 超深亚微米工艺

EEACC: 2570

中图分类号: TN47 **文献标识码**: A **文章编号**: 0253-4177(2002)08-0785-09

* 清华大学骨干人才支持计划(No.[2002]4)和国家重点基础研究发展规划(No. G-1998030403)资助项目

经 彤 男, 1966 年出生, 博士, 副教授, 从事 VLSI 性能驱动布图理论与算法的研究工作.

洪先龙 男, 1940 年出生, 教授, 博士生导师, 长期从事 VLSI CAD 的科研与教学工作.

蔡懿慈 女, 1960 年出生, 副教授, 从事 VLSI CAD 的教学和科研工作.

2002-05-13 收到