

# Design-for-Debug Routing for FIB Probing

Chia-Yi Lee, Tai-Hung Li, and Tai-Chen Chen

Department of Electrical Engineering, National Central University, Taoyuan, Taiwan  
{995201027, 975401006}@cc.ncu.edu.tw; tcchen@ee.ncu.edu.tw

**Abstract**—To observe internal signals, physical probing is an important step in post-silicon debug. Focused ion beam (FIB) is one of most popular probing technologies. However, an unsuitable layout significantly decreases the percentage of nets which can be observed through FIB probing for advanced process technologies. This paper presents the first design-for-debug routing to increase the FIB observable rate. The proposed algorithm, which adopts three FIB states and costs to enhance the maze routing, keeps at least one FIB candidate for each net while routing. Experimental results demonstrate that the proposed method can significantly increase the FIB observable rate under 100% routability.

## I. INTRODUCTION

In the increasing density of circuits, post-silicon debugging and circuit editing play specific roles during the processes of chip design as unexpected situations occurred to lead nets open or short. An efficient post-silicon debugging not only reduces the cost of mask modification but also accelerates speed of verification and shortens the time to market [1], [2]. Focused ion beam (FIB) is a physical probing technology using widely in post-silicon debugging and circuit editing [3], [4], [5]. FIB adopts an ion beam to remove inter-layer dielectric (ILD) and remain an FIB hole as shown in Fig. 1 [6].

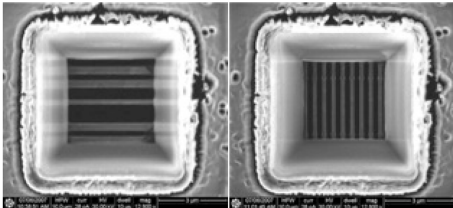


Fig. 1. FIB holes.

An FIB system operates in a similar manner as a scanning electron microscope (SEM) or a transmission electron microscope (TEM) except that the FIB system utilizes a focused beam of ions (gallium most of the time) instead of a beam of electrons. When operating at a low beam current, FIB can be used for imaging the sample surface with high resolution. When operating at a high beam current, FIB can be used for milling the surface. Because the ions are larger, heavier, slower, and positive compared to electrons, the ion beam cannot easily penetrate within individual atoms of the sample and hence can more easily break the chemical bonds of the substrate atoms, which makes FIB suitable for surface milling [7]. Fig. 2 illustrates an example of using FIB technique to observe a target signal inside a chip. In Fig. 2(a), the surface milling is performed by applying a focused ion beam of Ga<sup>+</sup> to hit the surface of ILD, breaking the bonds of a certain amount of surface material, sputtering out ions (mostly positive ions),

and gradually forming a hole right above the target signal. Meanwhile, an electron beam is applied to the surface to neutralize the sputtered positive ions, and sometimes certain gas (such as XeF<sub>2</sub>) is also applied to assist the etching (mainly for preventing the re-deposition of the sputtered surface material). Next, in Fig. 2(b), FIB is used to deposit metal (Pt in this case) onto the dug hole. When the ion beam hits the gas of metal, the metal will chemisorb on the surface through FIB-assisted chemical vapor deposition [7]. The deposition of metal then forms a probe pad for the target signal.

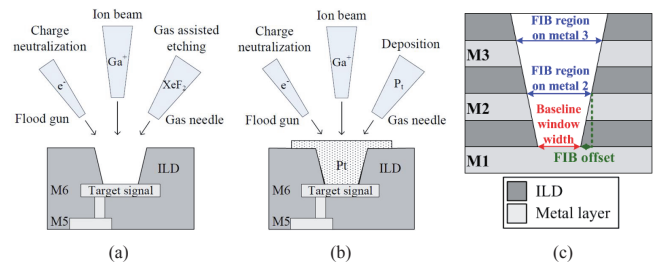


Fig. 2. Operations of FIB. (a) Milling. (b) Deposit. (c) Side view of FIB.

When using FIB probing or circuit editing, we must make sure that the metal of other signals above the observation site will not be touched or removed. Otherwise, the probe pad may be connected to some unwanted signals or the original circuit functionality may be changed. Unfortunately, while the technology node continually scales, the resolution of FIB technologies does not scale accordingly. As a result, for advanced process technologies, the circuit layout becomes denser and it is more difficult for FIB probing to pass through all the unwanted metals on top of the observation site without modifying them, which significantly reduce the percentage of signals that can be observed by FIB probing. As mentioned in [8], more than 70% nets of a circuit cannot be observed if a 90 nm or advanced technology is used. [9] proposed an automatic tool to efficiently identify the sites which can be used to perform the desired FIB circuit editing. [8] proposed a framework to maximize the probability that a signal can be observed by FIB probing in the post-layout stage. The proposed method adjusts metals to different layers to increase the FIB observable rate. The *FIB observable rate* means that the percentage of the total nets which can be successfully observed by FIB probing. However, almost 30% nets of a circuit still cannot be observed after a post-layout circuit editing is applied in a 90 nm technology [8]. Therefore, an automatic routing tool is needed to create a layout which is friendly for applying FIB probing or circuit editing.

In this paper, we propose the first design-for-debug routing to increase the FIB observable rate. The proposed algorithm, which adopts three FIB states and costs to enhance the maze routing, keeps at least one candidate region for each net while routing. Experimental results demonstrate that the proposed

method can significantly increase the FIB observable rate under 100% routability.

The remainder of this paper is organized as follows. Section II presents definitions and FIB violations. Details of the routing method are presented in Section III. Experimental results are outlined in Section IV. Finally, conclusions are presented in Section V.

## II. DEFINITION AND FIB VIOLATION

### A. Definition

According to [8], we define the following items:

- FIB observable rate: The percentage of the total nets which can be successfully observed by FIB probing.
- baseline window: The area of the bottom of an FIB-dug hole (usually square).
- baseline window width: The width of the baseline window (as shown in Fig. 2(c)).
- FIB offset: The difference between the window edges on two adjacent layers (as shown in Fig. 2(c)).
- FIB candidate of a net: A site which can be used to observe the net signal using FIB technique.
- FIB region of a net on a layer: An area which is reserved for an FIB candidate.
- Critical region of a net on a layer: An overlapped region of all FIB regions of the net on the layer. To increase the FIB observable rate, other nets must avoid passing through these regions.

As shown in Fig. 3, the baseline window width and the FIB offset of an FIB system are 3 and 1 routing grids, respectively. The FIB regions on Layers 1, 2, and 3 should keep 3 by 3, 5 by 5, and 7 by 7 routing grids, respectively. Gray (blue) boxed in Figs. 3(a) to (c) present the FIB regions of net *A* on Layers 1 to 3, respectively. Net *A* (occupies 4 routing grids on Layer 1) has two FIB candidates. The critical region of net *A* on Layer 1 is shown in the gray (red) box in Fig. 3(d).

### B. FIB Violation

According to the definitions in Section II.A, the FIB violations during routing can be classified into two types:

- Active violation: a routing net (net *A*) passes over a routed net (net *B*), and lets the routed net have no site to apply FIB (as shown in Fig. 4(a)).
- Passive violation: a routing net (net *A*) passes below a routed net (net *B*), and lets the routing net have no site to apply FIB (as shown in Fig. 4(b)).

## III. DESIGN-FOR-DEBUG ROUTING

The main idea of the design-for-debug routing is to keep at least one FIB candidate for each net after routing. The proposed method includes two stages: FIB-driven routing and routability-driven routing. During FIB-driven routing, penalty weights are assigned on specific routing grids to guide the routing to keep FIB candidates for nets. However, if no FIB

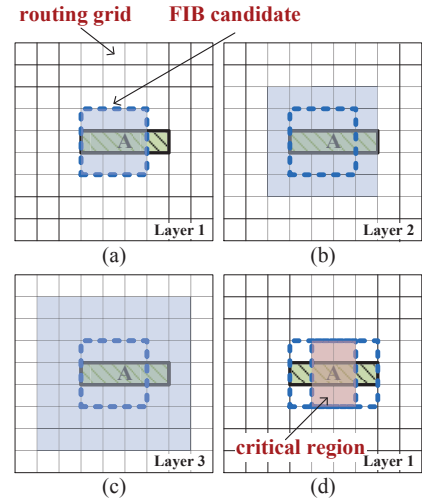


Fig. 3. (a) An FIB candidate and the FIB region of net *A* on Layer 1. (b) The FIB region of net *A* on Layer 2. (c) The FIB region of net *A* on Layer 3. (d) The critical region of net *A* on Layer 1.

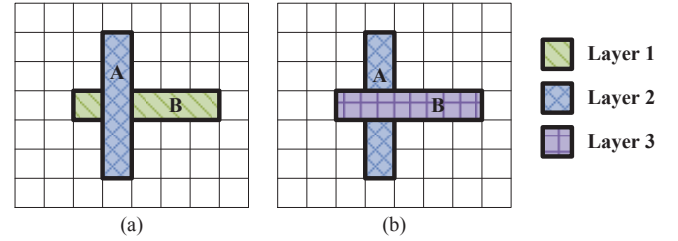


Fig. 4. FIB violations. (a) Active violation. (b) Passive violation.

candidate is kept for a net, the net is reserved to the routability-driven routing stage which will remove the penalty weight on routing grids to increase the routability.

### A. FIB-Driven Routing

Differing with a traditional maze routing, which considers wirelength as the routing cost, FIB-driven routing applies three FIB states, lookup, lookdown, and lookpin states, with different costs on the routing grids. The propagation step of maze routing is also modified to record more information of propagation.

1) *Lookup State*: The purpose of the lookup state is to prevent passive violation. After routing a net, the routing grids on the layers which are lower than the net are marked as the lookup state  $P_U$  to avoid other nets passing through. Fig. 5 presents an example of the lookup state. Net *B* in Fig. 5(a) has no FIB candidate, due to the lookup state is not considered. Routing result on Layers 1 and 2 are presented in Figs. 5(b) and (c), respectively. With considering the lookup state of net *A*, net *B* is detoured to keep FIB candidates for itself, as shown in Fig. 5(d). Figs. 5(e) and (f) present the routing grids with the lookup state after routing net *A*.

2) *Lookdown State*: The purpose of the lookdown state is to prevent active violation. After routing a net, the FIB candidates of the net can be calculated and then the routing grids on the layers which are higher than these FIB candidates are marked as the lookdown state  $P_D$  to avoid other nets

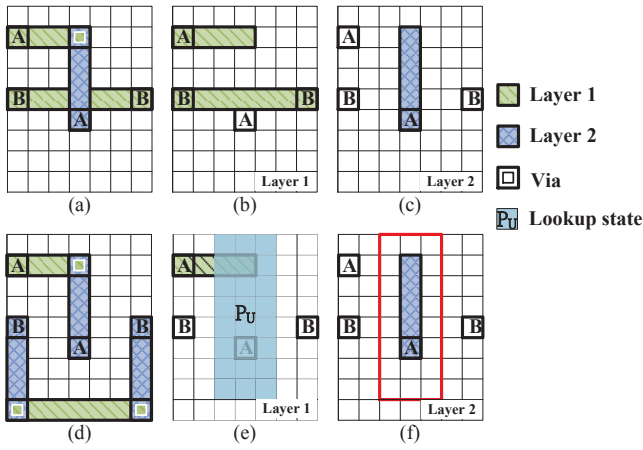


Fig. 5. Lookup state. (a) Routing result without considering the lookup state. (b) Layer 1 of (a). (c) Layer 2 of (a). (d) Routing result with considering the lookup state. (e) Layer 1 of (d). (f) Layer 2 of (d)

passing through. Besides, the routing grids on the layers which are higher than the critical region are marked as blockages  $O$  to prevent other nets passing through. Fig. 6 presents an example of the lockdown state. Fig. 6(a) presents a routing result without considering the lockdown state. Net  $A$  routes directly without keeping FIB region for net  $B$ . Figs. 6(b) and (c) present the FIB candidates of net  $B$  on Layers 1 and 2, respectively. With considering the lockdown state, net  $A$  is detoured to keep FIB candidates for net  $B$ , as shown in Fig. 6(d). Figs. 6(e) and (f) present the routing grids with the lockdown state after routing net  $B$ .

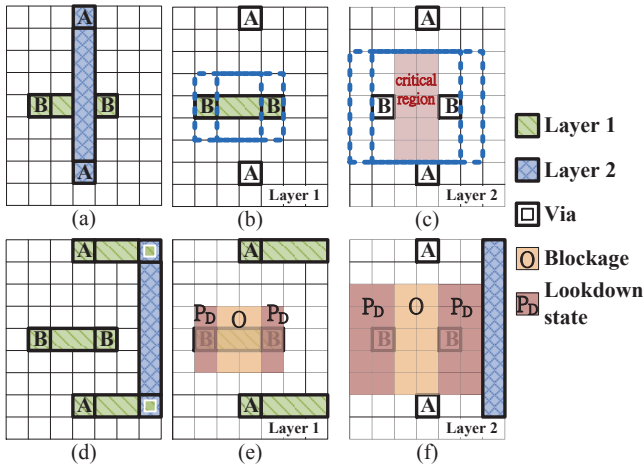


Fig. 6. Lookdown state. (a) Routing result without considering the lockdown state. (b) Layer 1 of (a). (c) Layer 2 of (a). (d) Routing result with considering the lockdown state. (e) Layer 1 of (d). (f) Layer 1 of (d).

3) *Lookpin State*: The purpose of the lookpin state is to improve the routability. A net cannot be successfully routed if one of its signal pin is covered by a blockage. Before routing a net, the routing grids on the layers which are higher than signal pins are marked as the lookpin state  $P_P$  to avoid other nets passing through. Fig. 7 presents an example of the lookpin state. In Fig. 7(a), the routing path of net  $A$  on Layer 1 is too close to one signal pin of net  $B$  since the lookpin state is not considered. Therefore, the signal pin of net  $B$  is covered by

blockages induced by net  $A$  on Layers 2 and 3, as shown in Figs. 7(b) and (c), respectively. With considering the lookpin state, net  $A$  is detoured to keep the routing resource of the signal pin of net  $B$ . Figs. 7(d) to (f) present routing net  $A$  considering  $P_P$  of net  $B$  on Layers 1 to 3, respectively.

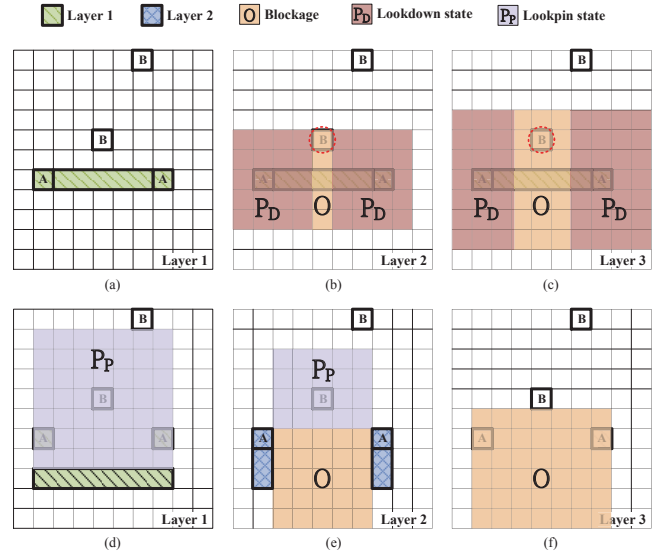


Fig. 7. Lookpin state. (a) Routing result without considering the lookpin state on Layer 1. (b) States of routing grids on Layer 2. (c) States of routing grids on Layer 3. (d) Routing result with considering the lookpin state on Layer 1. (e) States of routing grids on Layer 2. (f) States of routing grids on Layer 3.

4) *Propagation Modification*: Traditional maze routing cannot predict whether a routing path keeps at least one FIB candidate while the backtracking stage. To overcome this issue, the number of FIB candidates should be propagated with maze routing. We define  $F_a$  to record the number of FIB candidates on each routing grid in the routing path from the source ( $S$ ) to the target ( $T$ ). If  $F_a$  of  $T$  is equal to or larger than one, there is at least one FIB candidate on this path and this path can be kept. While propagating on non-lookup-state grids which comprise continued length is longer than the baseline window width,  $F_a$  becomes one from zero. After  $F_a$  becomes one,  $F_a$  will increase one if propagating to another non-lookup-state grid; however, if propagating to a lookup-state grid,  $F_a$  keeps its original value.

## B. Routability-Driven Routing

The purpose of routability-driven routing is to route the nets which cannot be successfully routed during FIB-driven routing. The reason of the nets which are failed during FIB-driven routing is that  $P_D$  and  $O$  restrict the routing resource. Therefore, we sequentially release  $P_D$  and  $O$  on the routing grids to make routing have more resource in the routability-driven routing stage. In Fig. 8(a), nets  $A$  and  $B$  are unrouted nets which are restricted by  $P_D$  of net  $C$  and  $O$  of net  $D$  during FIB-driven routing. Figs. 8(b) and (c) present  $P_D$  of net  $C$  on Layer 1 and  $O$  of net  $D$  on Layer 2, respectively. In Fig. 8(d),  $P_D$  of net  $C$  are released first, and therefore net  $A$  can be routed. Although net  $A$  can be routed after removing  $P_D$ , net  $B$  still cannot be routed because of  $O$  of net  $D$ . In Fig. 8(e),  $O$  of net  $D$  are modified to  $P_D$ , and then net  $B$  can

be routed without restricting from  $O$ . After routability-driven routing, the routing result can achieve maximum routability.

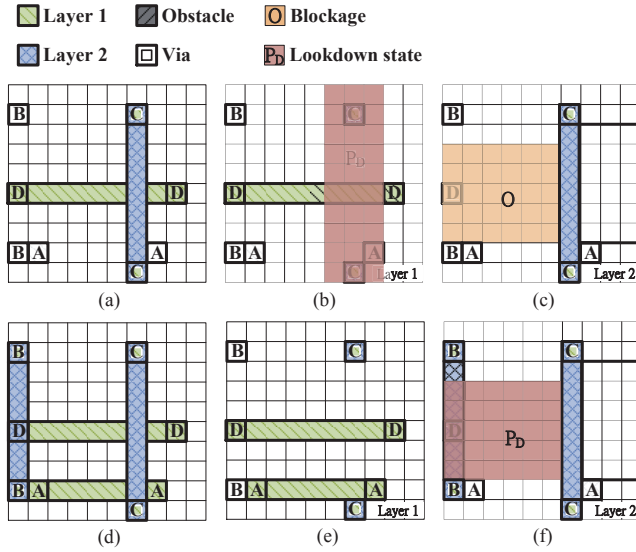


Fig. 8. Routability-driven routing. (a) Unrouted nets  $A$  and  $B$  and routed nets  $C$  and  $D$ . (b)  $P_D$  of net  $C$  on Layer 1. (c)  $O$  of net  $D$  on Layer 2. (d) Routing result of nets  $A$  and  $B$ . (e) Release  $P_D$  of net  $C$  on Layer 1 with routing net  $A$ . (f) Modify  $O$  to  $P_D$  of net  $D$  on Layer 2 with routing net  $B$ .

#### IV. EXPERIMENTAL RESULTS

The proposed design-for-debug routing was implemented in C++ language on a Linux workstation with a 2.4 GHz Intel Xeon CPU. Two sets of benchmarks, MCNC [10] and Faraday [11], are used to test the proposed algorithm. Detailed information of benchmarks are listed in TABLE I.

TABLE I. BENCHMARK INFORMATION

Circuit	Chip region ( $\mu m^2$ )	Layer	#Total Nets
s5378	124 × 70	3	1694
s9234	116 × 66	3	1486
s13207	186 × 104	3	3781
s15850	198 × 112	3	4472
s38417	320 × 172	3	11309
s38584	474 × 190	3	14754
dma	204.4 × 204.4	6	12611
dsp1	353.2 × 353.2	6	28399
dsp2	321.6 × 321.6	6	28383
risc1	502 × 502	6	33761
risc2	480 × 480	6	33761

TABLE II compares the FIB observation rates. In TABLE II, "Traditional Maze Routing" refers to a router implemented by the traditional maze routing, "Design-for-Debug Routing" refers to the proposed routing considering the FIB observable rate, "FIB O.R. (%)" refers to the FIB observable rate, "WL ( $\mu m$ )" refers to the wirelength in micro-meter unit, and "R.T. (m)" refers to the executing time in minute unit. Both methods can achieve 100% routability. However, the traditional routing can only achieve 41% in the average FIB observable rate. The proposed design-for-debug routing can achieve 100% FIB observable rate for nine benchmarks with 6% wirelength increasing.

TABLE II. COMPARISON OF THE FIB OBSERVATION RATES. "TRADITIONAL MAZE ROUTING" REFERS TO A ROUTER IMPLEMENTED BY THE TRADITIONAL MAZE ROUTING, "DESIGN-FOR-DEBUG ROUTING" REFERS TO THE PROPOSED ROUTING CONSIDERING THE FIB OBSERVABLE RATE, "FIB O.R. (%)" REFERS TO THE FIB OBSERVABLE RATE, "WL ( $\mu m$ )" REFERS TO THE WIRELENGTH IN MICRO-METER UNIT, AND "R.T. (M)" REFERS TO THE EXECUTING TIME IN MINUTE UNIT.

Circuit	Traditional Maze Routing			Design-for-Debug Routing (Ours)		
	FIB O.R. (%)	WL ( $\mu m$ )	R.T. (m)	FIB O.R. (%)	WL ( $\mu m$ )	R.T. (m)
s5378	50.89	21875	1.2	100.00	23590	3.0
s9234	55.82	16410	1.0	100.00	17818	2.1
s13207	53.76	51844	5.6	100.00	55791	12.1
s15850	51.76	64209	7.5	100.00	68804	23.4
s38417	54.30	141492	40.2	100.00	152623	62.1
s38584	47.78	196093	87.2	100.00	210230	154.0
dma	26.37	283755	94.1	100.00	295568	178.5
dsp1	26.58	628661	429.0	100.00	653576	907.3
dsp2	27.80	558743	370.6	99.99	582190	624.0
risc1	28.20	1073640	1258.7	100.00	1112640	3444.9
risc2	29.16	977084	1165.7	99.99	1015910	2359.8
<b>Comp.</b>	<b>0.41</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1.06</b>	<b>2.15</b>

#### V. CONCLUSION

This paper proposed the first design-for-debug routing to increase the FIB observable rate. By adopting three FIB states and costs to enhance the maze routing, each net can be kept at least one FIB candidate while routing. Experimental results demonstrated that the proposed method can significantly increase the FIB observable rate under 100% routability.

#### REFERENCES

- [1] M. Cho, K. Yuan, Y. Ban, and D. Z. Pan, "ELIAD: Efficient lithography aware detailed router with compact post-OPC printability prediction," in *Proc. DAC*, 2008, pp. 504–509.
- [2] K. H. Chang, I. L. Markov, and V. Bertacco, "Reap what you sow spare cells for post-silicon metal fix," in *Proc. ISPD*, 2008, pp. 103–110.
- [3] T. Malik, R. Jain, R. Nicholson, and T. Lundquist, "Role of circuit edit in post-silicon debug and diagnostics," in *IEEE Silicon Debug & Diagnosis*, 2005.
- [4] C. G. Talbot, M. Park, N. Richardson, P. Alto, and D. Masnaghetti, "IC modification with focused ion beam system," U.S. Patent NO. 5,140,164, 1992.
- [5] J. Giacobbe, "Physical design for debug: Insurance policy for IC's," in *Proc. ISPD*, 2013, pp. 122–122.
- [6] C. Rue, *Circuit Edit Basic*, FEI Company, 2008.
- [7] R. Desplats, T. Dargnies, J.-C. Courrage, P. Perdu, and J.-L. Noullet, "Calculation of the optimal FIB milling and deposition operations for easier and faster circuit reconfiguration," in *Proc. ISTFA*, 2000, pp. 407–414.
- [8] K.-A. Chen, T.-W. Chang, M.-C. Wu, M. C.-T. Chao, J.-Y. Jou, and S. Chen, "Design-for-debug layout adjustment for FIB probing and circuit editing," in *Proc. ITC*, 2011, pp. 1–9.
- [9] Y.-R. Wu, S.-Y. Kao, and S.-A. Hwang, "Minimizing eco routing for fib," in *Proc. VLSI-DAT*, 2010, pp. 351–354.
- [10] J. Cong, J. Fang, M. Xie, and Y. Zhang, "MARS—a multilevel full-chip gridless routing system," *IEEE Trans. on Computer-Aided Design*, vol. 24, no. 3, pp. 382–394, March 2005.
- [11] J. A. Roy, S. N. Adya, D. A. Papa, and I. L. Markov, "Min-cut floorplacement," *IEEE Trans. Computer-Aided Design*, vol. 25, no. 7, pp. 1313–1326, July 2006.