Selectively Patterned Masks: Beyond Structured ASIC

Donkyu Baek, Insup Shin, Seungwhun Paik, and Youngsoo Shin
Department of Electrical Engineering, KAIST
Daejeon 305-701, Korea

***Abstract*—Conventional structured ASIC still suffers from large delay and area due to the use of homogeneous array of tiles. We propose a new lithography method called selectively patterned masks (SPM), which enables more than one type of tiles to be used in structured ASIC. This structured ASIC using mixture of different tiles relaxes regularity. To assess SPM concept, A new structured ASIC is proposed; tile and routing architecture, and routing algorithm are all addressed. Experiment results using 45-nm technology show that proposed concept can push the limit of structured ASIC closer to traditional ASIC.**

I. Introduction

Sky-rocketing of mask cost in the recent technologies is one of the factors that limit sustaining growth of semiconductor business. A full mask set costs about $300K in 130-nm, $1M in 65-nm, and $4M in 45-nm technology [1]. FPGA is one of the solutions, but the gap between ASIC and FPGA is still too large to consider it as a device for volume production. It still suffers from large per-chip cost and large gap of performance (3.4∼4.6 times), area (an average of 35 times), and power consumption (an average of 14 times) compared with ASIC [2]. Gate array is another option to reduce mask cost by pre-fabricating FEOL (front-end-of-line) layers. But, its benefit has diminished in the recent technologies due to continuous increase in the number of BEOL (back-end-of-line) layers.

Structured ASIC is an array of homogeneous programmable cells (called tiles) with programmable routing. As well as active region, masks for most metal layers are already pre-fabricated and can be reused by different designs; only some contact/via masks need to be manufactured for each design, where these contact/via determines the functionality of each tile. However, structured ASIC is not yet mainstream device because of its inferior performance compared with that of ASIC; e.g. 2.0∼6.4 times of delay and 3.0∼7.0 times of area [3] compared to ASIC, even though it is much closer to that of ASIC than that of FPGA is. The limitation of current structured ASIC stems from the fact that it imposes too much regularity and too much generality.

To overcome the limitation of conventional structured ASIC, we propose to use a lithography technique that can selectively pattern tiles from different masks onto a wafer. A prototype of new structured ASIC is presented in Section III. A key component of this prototype is a tile architecture: two types of tiles are proposed to implement combinational gates and one type of tile for flip-flop. Routing architecture, which is another key component since it determines routability and thus circuit delay, is proposed. Experiments are conducted to express acceptability of proposed structured ASIC in 45-nm technology.

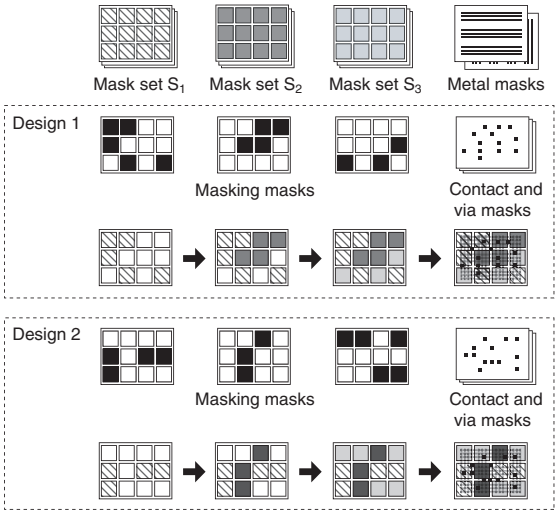

Fig. 1. A concept of SPM and its use in structured ASIC design.

II. Selectively Patterned Masks

In conventional structured ASIC, a mask set contains an array of homogeneous programmable logic (called a tile for brevity), which can implement all kinds of logic; this strict regularity deteriorates the performance of logic gates implemented by tile in terms of both speed and area.

We propose a new concept of lithography called selectively patterned masks (SPM). More than one array of tiles are assumed and different tiles are patterned on a wafer but selectively, i.e. some tiles are patterned while others are not. For this purpose, all tiles must have the same width and the same height even though their internal architectures are totally different. The motivation of SPM is to relax the regularity of conventional structured ASIC by allowing to use mixture of several different tiles, so that its performance can be pushed closer to that of ASIC.

An example of using SPM to design a structured ASIC is shown in Fig. 1. There are three mask sets S_1, S_2 and S_3 containing an array of tile. These masks, for logic function, are programmed by contact masks and metal masks, for routing, are programmed by via masks. To enable selective patterning,

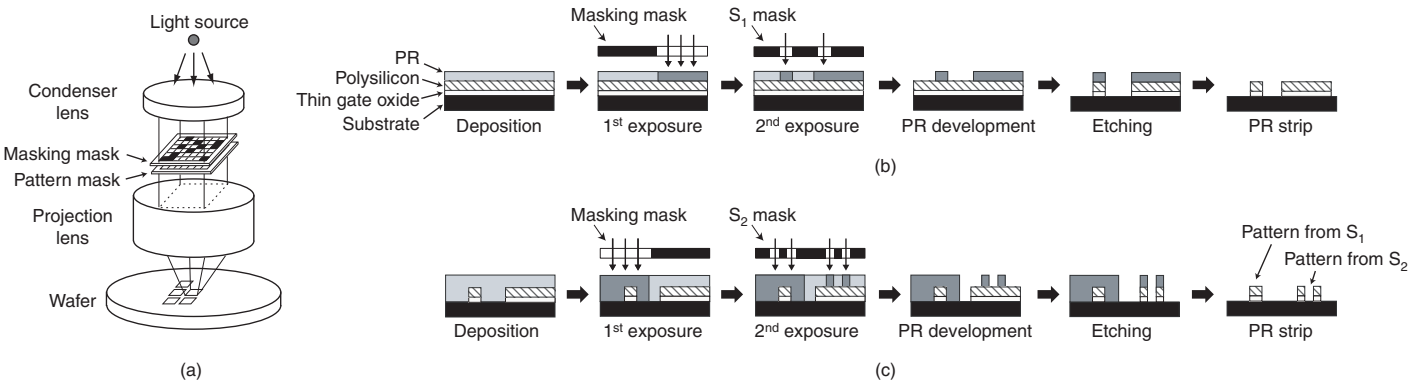

Fig. 2. (a) Conceptual lithograph setting for selective patterning. More practical solution; pattern transfer process of double exposure using (b) the first mask set and (c) the second mask set with their corresponding masking masks.

we use special mask, *masking mask*, to block patterning of unwanted mask region. In this setting, only contact and via masks together with masking masks need to be fabricated for each new design, like Design 1 and Design 2 in Fig. 1.

Conceptually, if a pattern mask, a mask from a mask set shown in Fig. 1, and a masking mask are used together (after they are aligned) during lithography process, selective patterning could be achieved, which is shown in Fig. 2(a). This method, however, is not supported by current lithography equipment. Moreover, the intensity of light will become weaker as it goes through two masks, which makes patterning process more difficult.

The practical solution is to adopt double exposure [4]. As shown in the second step of Fig. 2(b), the photoresist (PR) is first exposed to ultraviolet (UV) light through a masking mask that corresponds to S_1. The PR, in this case, is negative, i.e. the resist does not dissolve when be exposed to UV light. This property enables selective patterning by blocking lithography process on the region exposed through a masking mask. After the second exposure using S_1 mask, which is followed by development, etching, and strip, only the polysilicon pattern on the left remains, even though S_1 mask itself contains two polysilicon patterns. Similar process is repeated using S_2 mask and its corresponding masking mask as shown in Fig. 2(c), which allows two polysilicon patterns on the right to be developed. At the end, a target pattern consists of one polysilicon gate on the left that comes from the first mask set S_1 and two polysilicon gates on the right originating from S_2.

III. Structured ASIC Design with SPM

A. Via-Programmable Tile

We constructed three different tiles, two for combinational gates and one for flip-flops, to assess the benefit of using SPM when designing structured ASICs. For the sake of explanation, the tiles are denoted by T_1, T_2, and T_3 respectively. The width of T_1 and T_2 is a half of that of T_3 and the height of all types of tiles is the same (see Fig. 3). Each of T_1 and T_2 have 10 pins, implemented by metal2 (M2).

T_1 is designed to accommodate three logic gates by programming vias: two of them can implement any logic gates

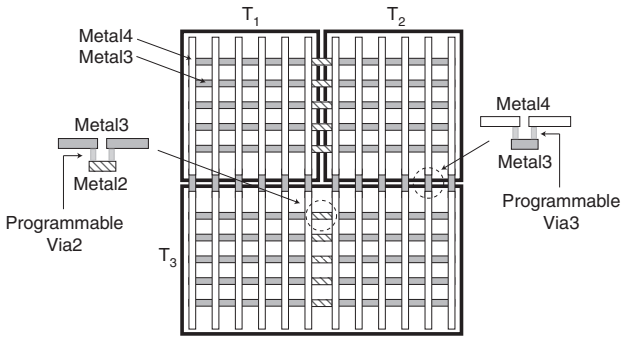

Fig. 3. Four arrays of routing grids on top of T_1, T_2, and T_3.

consisting of up to two polysilicon gates such as INV and NAND2; the remaining one implements a logic gate that requires three polysilicon gates including AOI21(see Fig. 4(a)). Two logic gates can be implemented in T_2; One implements logic gates requiring three polysilicon gates and another implements complex logic gates requiring five polysilicon gates like AOI221. After logic synthesis, we pack logic gates into tiles so that the utilization of tile is maximized. Note that T_1 and T_2 support gate sizing to some extent, e.g. $2\times$ INV can be implemented with 2 polysilicon gates in T_1 and $4\times$ INV can be realized with 4 polysilicon gates in T_2.

B. Via-Programmable Routing Architecture

Routing a net is accomplished by programming vias, where each via connects two metal segments in a routing architecture. A grid of horizontal M3 tracks and vertical metal4 (M4) tracks form the routing architecture, which is shown in Fig. 3 in a simplified form; for the experiment, we assumed 11 horizontal M3 tracks and 12 vertical M4 per each grid. As illustrated in Fig. 3, one grid is required for both T_1 and T_2 and two grids are required for T_3. To connect metal segments between two adjacent routing grids, short metal segments (M2 segments for horizontal and M3 for vertical) are inserted.

Because the routing grids are regularly placed all over the design area regardless of tiles placed underneath, just one mask set of metal segment can be shared by all designs. The real

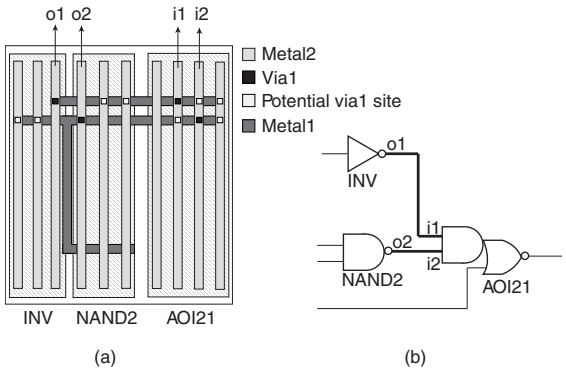

Fig. 4. (a) An example of intra-tile routing in T_1 and (b) corresponding netlist.

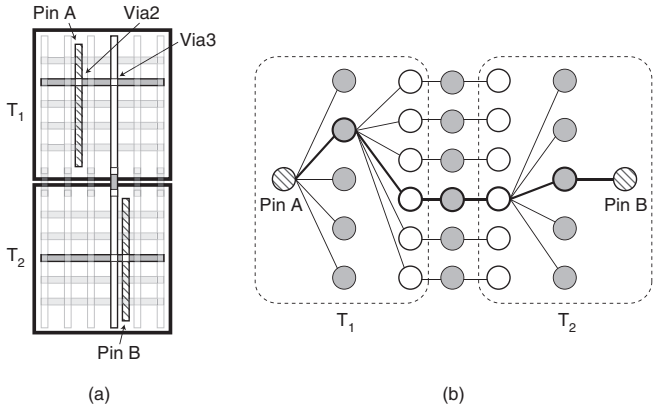

Fig. 5. (a) An example routing of a net and (b) its corresponding RRG.

connection between metal segments is achieved by insertion of programmable vias implemented by via masks as shown in Fig. 1.

IV. Regular Routing

A. Intra-Tile Routing

After packing logic gates into tiles, intra-tile routing is performed within each tile prior to performing inter-tile routing. Intra-tile routing only uses metal1 (M1) to route nets that are confined within each tile; this can reduce the number of nets that need to be routed during inter-tile routing, thereby improving the routability.

Fig. 4(a) shows an example of intra-tile routing, where the outputs of INV and NAND2 are connected to the inputs of AOI21 (see Fig. 4(b)) via horizontal M1. M1 tracks of T_1 and T_2 are designed so that every possible connection within a tile can be routed via intra-tile routing.

B. Inter-Tile Routing

To route all nets in a design, the proposed routing architecture is first transformed into a routing resource graph (RRG). In the RRG, each node corresponds to a metal segment, which can be either a pin or a routing track, and there is an edge between two nodes if they can be connected through a programmable via. Then we resort to a graph traversal algorithm to find paths between nodes, which are pins of nets to be routed.

An example of routing a net from pin A to pin B in the routing architecture is shown in Fig. 5(a); the corresponding RRG is shown in Fig. 5(b). Routing a net is basically finding a shortest path that connects two nodes, which corresponds to pins, in RRG; once some nodes and edges are used by the net, they cannot be used by other nets. The shortest path to route the net from pin A to pin B in Fig. 5(b), which is highlighted in a bold line, goes through 5 nodes and 6 edges. This corresponds to three M3 segments, two M4 segments, and six vias.

The goal of our regular routing is to successfully route all nets in a design while minimizing the sum of their wirelength; the wirelength of a net is approximated as the number of nodes used by the net in RRG. For this purpose, we used a maze routing algorithm with negotiation-based ripup and reroute [5]. We allow multiple nets to share a node at first, and increase the penalty of using nodes that are used by more than one net; this results in less critical nets, which can still be routed via other path, to detour and eventually each node is used by only one net after several iterations. The algorithm terminates in success when all nets are routed and nodes are exclusively used by nets.

V. Experimental Results

Experiments were performed to compare the proposed structured ASIC design (denoted by SPM) with standard ASIC design (denoted by ASIC) in terms of area and delay for 45-nm technology; several circuits were taken from ISCAS, ITC, and opencores [6] benchmarks. For ASIC, each circuit was synthesized [7] with a open library [8], which consist of 134 logic gates, to obtain a netlist; each netlist was then placed and routed using a commercial tool [9]. To obtain netlist for SPM, the same circuits were synthesized [7] using a library, which consists of 31 logic gates, that is characterized from T_1, T_2, and T_3 by SPICE simulation [10]; the netlists were placed using the commercial tool [9], where the packing of logic gates into tiles as well as intra-tile routing were performed via Tcl scripts. Then the routing was performed using a stand-alone tool, which was implemented in C language.

A. Assessment of Area and Delay

To assess the area overhead of SPM, logic synthesis was performed to minimize area (without regard to delay) in both ASIC and SPM. We forced about 70% of the placement region to be occupied by cells or tiles during physical design, which is a tight placement. Columns 2–4 in Table I shows a result. The circuits from SPM occupy, on average, 2.18× of the area of circuits from ASIC. There are several factors that contribute to the area overhead of SPM. First of all, the height of tile is 49% larger than that of stanard cells in ASIC; this is the result of effort to guarantee routability by increasing the number of horizontal M3 tracks. Another factor is under-utilization of tiles. Some logic gates use less number of polysilicons or

TABLE I
COMPARISON OF AREA (WITHOUT REGARD TO DELAY) AND DELAY (WITHOUT REGARD TO AREA) BETWEEN ASIC AND STRUCTURED ASIC WITH SPM

Bench-mark	Min. area (μm^2)			Min. delay (ns)		
	ASIC	SPM	SPM/ASIC	ASIC	SPM	SPM/ASIC
s820	263	660	2.51	1.40	1.97	1.41
s1196	530	1181	2.23	1.39	1.55	1.11
s1238	536	1130	2.11	1.26	1.63	1.29
s1423	879	1746	1.99	2.21	2.92	1.32
s5378	2034	4096	2.01	1.66	2.58	1.56
s9234	1594	3355	2.10	2.33	2.56	1.10
s13207	5801	11080	1.91	2.63	3.14	1.20
s15850	5736	11502	2.01	2.81	3.47	1.24
b04	1091	2787	2.56	1.81	2.03	1.12
b07	680	1628	2.39	1.54	2.09	1.36
b10	270	584	2.16	1.57	2.02	1.28
b11	802	2067	2.58	1.87	2.54	1.36
b12	1794	3766	2.10	1.93	2.51	1.31
b13	656	1269	1.94	1.38	1.70	1.23
pcm	823	1807	2.20	1.33	1.78	1.34
pci	971	2084	2.15	2.18	2.53	1.16
usb	973	2130	2.19	1.24	1.68	1.36
sasc	1211	2701	2.23	1.73	2.09	1.21
i2c	1656	3346	2.02	1.57	2.02	1.29
des	3351	7530	2.25	3.27	4.16	1.27
Average			2.18			1.28

metals than that are allocated when mapped to tiles; e.g. INV uses only one polysilicon even though twos are available, thus one is wasted. The proposed method, however, has advantage when compared to conventional structured ASIC: 3×–7× [3] and 4× of area [11] compared to ASIC design.

In case of delay comparison, logic synthesis was performed to minimize delay. Columns 5–7 in Table I shows a result. The average delay of SPM is 1.28× of that of ASIC design, which is very promising; conventional structured ASIC is reported to be slower than ASIC by 2.0×–6.4× [3]. There are two reasons of delay increase in SPM circuits: lack of logic gates in SPM library and use of regular routing architecture. The lack of library gates (31 gates) seems to be the main reason since they are slow even before physical design. The routing architecture does not affect the delay too much.

B. Approaches of Improving Routability

Lack of M3 tracks is the main cause of routing failure in SPM; each pin occupies at least one M3 track to be routed, which leaves only small number of M3 tracks available for routing other nets. For example, T_1 can have at most 10 pins while the number of M3 tracks is 11, which leaves only one track for connection nets passing the tile. There are two approaches to increase the number of available M3 tracks; one is to increase the tile height (thereby increasing the number of M3 tracks in a tile) and the other is to perform intra-tile routing (thereby reducing the number of nets that may occupy M3 tracks).

To assess these approaches, we took four circuits and performed packing very densely so that routing would fail, i.e. some metal segments are occupied by more than one net (see Section. IV-B). To trade-off the runtime with the routability,

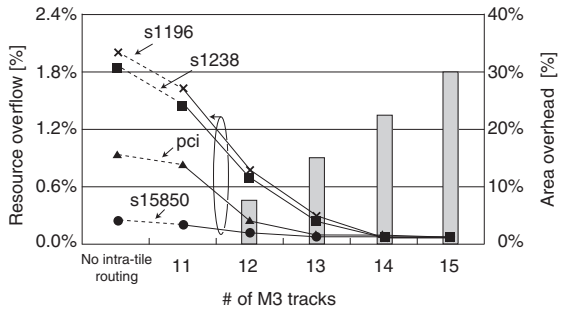

Fig. 6. Trend of resource overflow and area overhead as the number of M3 tracks within a tile is increased.

the rip and reroute was performed up to 25 iterations. In Fig. 6, we plotted the change in the proportion of resource overflow as we vary the number of M3 tracks; resource overflow means the number of metal tracks that are used by more than one net. The area overhead due to increased tile height is also depicted in gray bars. We also compared the case when intra-tile routing was not performed to see its impact on the routability.

In four circuits, all nets were routed successfully when the number of M3 track was increased to 15.

VI. CONCLUSION

The use of selectively patterned masks (SPM) allows more than one type of regular tiles to be used in structured ASIC design. The motivation of SPM is to relax the strict regularity of conventional structured ASIC so that its performance (area and delay) is pushed closer to ASIC design. New tile architecture and routing algorithm are also developed to asses performance of prototype of new structured ASIC. Experiments show 2.2× of area and 1.3× of delay compared to ASIC design, which are very promising compared to conventional structured ASIC.

There is still a possibility of improvement, which is left for future investigation. Tile packing could be performed in technology mapping stage to maximize the number of intra-tile nets. Routing algorithm could be improved for routability and circuit delay. A clock network design, which is not addressed in this paper, needs investigation.

REFERENCES

[1] M. Ooishi, "TSMC takes lead in 45nm IC mass production," *Nikkei Electronics Asia*, July 2007.
[2] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," *IEEE Trans. on CAD*, vol. 26, no. 2, pp. 203–215, Feb. 2007.
[3] N. V. Shenoy, J. Kawa, and R. Camposano, "Design automation for mask programmable fabrics," in *Proc. DAC.*, June 2004, pp. 192–197.
[4] M. Fritze, B. Tyrrell, T. Fedynyshyn, and M. Rothschild, "High-throughput hybrid optical maskless lithogrphy: all-optical 32-nm node imaging," in *Proc. SPIE*, Mar. 2005, pp. 2743–2748.
[5] V. Betz and J. Rose, "VPR: A new packing, placement and routing tool for FPGA research," in *Proc. Int. Workshop. on Field-Programmable Logic and Applications*, Sept. 1997, pp. 213–222.
[6] "OpenCores benchmarks," http://www.opencores.org/.
[7] Synopsys, "Design Compiler User Guide," Sept. 2008.
[8] "Nangate cell library," http://www.nangate.com/.
[9] Cadence, "SoC Encounter User Guide," Nov. 2007.
[10] Synopsys, "HSPICE User Guide," Sept. 2009.
[11] Y. Ran and M. Marek-Sadowska, "An integrated design flow for a via-configurable gate array," in *Proc. ICCAD*, Oct. 2004, pp. 582–589.