A Highly Flexible Sea-of-Gates Structure for Digital and Analog Applications

PHILIPPE DUCHENE AND MICHEL J. DECLERCQ, MEMBER, IEEE

Abstract—**This paper describes a new sea-of-gates structure optimized for digital random logic applications as well as for regular arrays and analog circuits. Associated with a dedicated design procedure and a systematic metallization strategy, the structure features a full cell-abutment capability and true channelless routing. After reviewing the advantages and limitations of currently available arrays, the main characteristics of the array architecture are presented, and applications to different circuit families are detailed. Design automation tools suited to the structure and design methodology are reviewed. Design results and performance are presented for several macroblocs and are compared with other semicustom approaches. A set of rules is finally presented, which allows an automatic transformation of the sea-of-gates layout into a topologically equivalent full-custom layout, converting semicustom prototypes to full-performance circuits.**

I. INTRODUCTION

GATE ARRAYS, including their recent improvement called sea-of-gates, cover roughly 50 percent of the ASIC market. The key to their success lies in three decisive advantages in comparison with full-custom designs: 1) a reduced turnaround time, due to a lower number of customized wafer processing steps; 2) a reduced fabrication cost, due to the smaller number of customized masks; and 3) a reduced design time, due to the existence of powerful CAD tools. These advantages are particularly meaningful when only a few prototypes or small commercial series must be produced. The limited performance and the relatively low packing density of semicustom arrays generally lead designers to choose a full-custom methodology when large series are planned. However, continuous progress in the semiconductor processing technology, and the associated reduction of the basic device size, still increase their success. Masters containing up to 100K gates and more are announced. To deal with such a complexity, a higher flexibility in circuit design is required, and advanced CAD strategies must be defined.

A. Basic Sea-of-Gates Concepts

The sea-of-gates array, also called continuous gate array, channelless array, or Gate Forest, is derived from the well-known gate array [1]. A brief review of the main gate array features is presented in [2]. Two concepts are at the origin of the so-called second-generation gate array: channelless architecture and gate isolation.

The suppression of predefined channels [3], [4] achieves a better silicon utilization. The routing channels still exist, but they are placed over unused core cells. This results in a higher flexibility in channel width and location. The gate isolation concept [5], [6], as opposed to oxide isolation, means that the basic cell layout includes no gap in the diffusion layer, isolating a cell from the neighboring one. The electrical isolation is performed by a transistor gate connected to the appropriate power (V_{SS} or V_{DD}). Flexibility in cell design, together with a significant reduction of the area per transistor, are the major advantages of this technique. Of course, gate isolation and channelless layout can be used together [7].

Other minor improvements, generally aiming to provide better implementations of some classes of circuits, have been added:

- a third level of metal simplifies routing problems and allows to stack the routing channels over the active cell area [8];
- supplementary transistors can be added to the basic cell or between basic cells in order to increase the RAM packing density [9];
- a part of the master can be reserved for a RAM block [10];
- by increasing the number of n devices, the basic cell can be designed to efficiently support dynamic circuits [2].

Finally, one can observe that currently available sea-of-gates arrays are generally optimized to locally reduce silicon consumption, in order to increase the apparent gate density. However, global layout constraints are not yet taken into account: routing, for example, consumes the biggest part of the active area. This widely decreases the actual gate density.

B. Limitations

Actual sea-of-gates still suffer from several limitations. The most typical ones follow.

1. Cell-Based Designs: The floorplan of array-based designs is composed of successive rows of cells alternating with routing channels. This has two major consequences: first, clustering of functions is not naturally achieved (there

Manuscript received September 19, 1988; revised January 12, 1989.
The authors are with the Electronics Laboratory, Swiss Federal Institute of Technology (EPFL), CH-1015 Lausanne, Switzerland.
IEEE Log Number 8926942.

0018-9200/89/0600-0576$01.00

is only one level of hierarchy); and second, numerous long interconnection wires are often needed, leading to a loss of performance and an area penalty.

2. Fixed Cell Libraries: Basic functional cells are kept in a static and inevitably limited library. In addition to the huge work of maintaining the library and adapting it to technology updates, two drawbacks are to be pointed out: if a function does not belong to the library, it must be implemented by using several cells, thus creating area consuming connections; and transistor sizing is not possible.

3. Analog Capabilities: In spite of some recent efforts [12], the analog world is poorly represented in semicustom environments. Very few analog cells are available. Cell-based CAD tools never give access to the isolated transistor. Moreover, routing channels create coupling capacitances, and are a nonnegligible source of noise. No functional capacitors or resistors are provided.

4. Regular Circuit Implementation: Unlike the previous one, this problem has been widely investigated. Three solutions have been tried to implement regular circuits (especially RAM) in gate arrays or sea-of-gates:

a) Disposing special RAM blocks on the chip [10]—this allows compact and efficient RAM circuitry. On the other hand, the size and location of the circuit is imposed.
b) Distributing a special RAM core cell among the array—but this leads to low densities for the RAM circuit, as well as for the logic. Moreover, the RAM cell is not necessarily suited to other circuits.
c) Using a core cell which supports regular circuit requirements, for example, by increasing the number of n devices. This seems to be the best solution, although these supplementary transistors serve only that purpose.

5. Mass Production: Actual arrays are still not suited to mass production, where mask cost and fabrication delay have minor importance, but where the highest performance and densities are expected. For this type of production, full-custom design styles are generally preferred. Two possibilities are offered to designers who wish to convert a gate array prototype into a full-custom circuit:

a) converting it to a standard-cell design, if a compatible standard-cell library exists, and if standard cell performance level is sufficient; or
b) making a complete redesign of the circuit, possibly generating new errors.

II. Layout Philosophy of the New Structure

A. Objectives

Most of the limitations described above can be widely balanced by an appropriate layout philosophy. Two aspects will be examined: the basic cell design and the metallization strategy. These have been worked out on the basis of four main objectives.

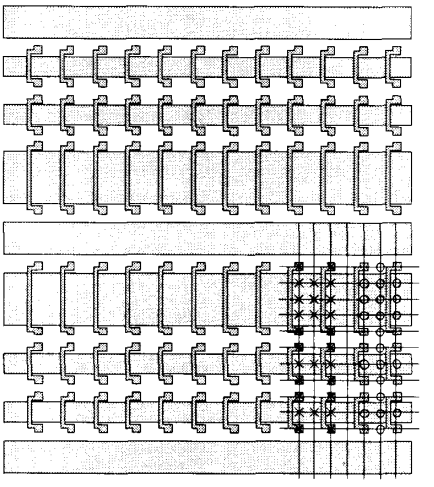


Fig. 1. Array architecture with grid, contact, and via locations.

1. Random Logic Design Philosophy: Instead of a cell-based design, we promote the so-called macrocell, or building-block design style [11]. That means suppressing, or at least widely reducing the number and size of routing channels, by promoting cell abutment, and by using generated cells of arbitrary complexity, instead of choosing them in a library. Two major consequences can be pointed out: first, the array must be specially designed (high level of routability and connectability, cell transparency, etc., and second, suited CAD tools, such as floorplanners, module and cell generators, must be developed. Some kind of compatibility between the array and existing design automation algorithms would help.

2. Regular Circuit Implementation: This should be improved not only for the RAM, but also for other regular circuits: PLA, ROM, or other regular logic arrays (parallel multipliers, register stacks, data paths, etc.). This objective can be achieved by carefully designing the basic core cell.

3. High-Quality Analog Circuits: Although we do not plan to design an "analog sea-of-gates," we should answer the growing demand on analog parts in digital ASIC's. Analog designers main requirements are: the capability of implementing very wide or very long channel transistors, a high level of symmetry, a good routability (to reduce wire length and coupling capacitances), a good matching (achieved by a full regularity), and access to functional capacitors.

4. Sea-of-Gates to Full-Custom Translation: An automatic procedure converting sea-of-gates layouts into correct full custom should be available.

B. Array Architecture

The target technology is a classical two-metal 2-μm CMOS technology. The array architecture has been designed to be easily convertible to other design rules, provided that at least two levels of metal are available.

TABLE I
OPTIMUM N- AND P-TRANSISTOR NUMBER

circuit	# trans. N	# trans. P	optimum N/P ratio
static combinational logic	n	n	1
SRAM	4	2	2
PLA or ROM	1 or 2	0	∞

Fig. 1 presents a detailed view of a part of the array. The following features can be observed.

1. Channelless Array: Obviously, no pre-defined channels are necessary, because routing channels will be avoided as much as possible.

2. Isolation Technique: The gate isolation technique [5] was chosen for its flexibility in the cell design, and the granularity of cell width. A substantial area reduction is also expected. Moreover, the diffusion layer continuity is a decisive requirement for the design of regular and analog circuits.

3. N/P Transistor Count Ratio: An array designed to support only static complementary logic would obviously contain the same number of n and p devices. By contrast, arrays designed to support dynamic circuits [2] are provided with a bigger number of n devices. Dynamic circuits are not particularly addressed by this array, but n/p transistor count contributes to the regular circuit implementation, too. In Table I, the optimal number of n and p devices are given for several common regular cells (isolation transistors are not part of the count). Obviously, a trade-off must be found between complementary logic and regular circuit density. N/P count also modifies the packing density and the routability (number of routing tracks crossing the cell). An optimum of two was found. This number provides the best integration rate for the RAM circuit, but ROM or PLA rates are also satisfying. With the further given transistor sizes, it yields enough routing tracks. Finally, although one n diffusion strip is not used in complementary logic cells, its area is not lost, but serves for routing purpose. Indeed, the need of routing area is much more important in random logic than in PLA or RAM, where only two or three global signal wires cross the cell.

4. Basic Cell Layout: An orthogonal comb structure made of horizontal diffusion strips and vertical poly gates forms the cell layout. This pattern achieves a good transistor density and provides the most natural serial–parallel connections for creating logic trees (random logic) or big W or L transistors (analog circuits). It is also the most regular and symmetrical structure. This feature guarantees the best matching in analog circuits.

5. Grid: Unlike gate arrays, no distinction is made between active and routing areas. Metal wires are intended to cover the whole array. It is thus natural to define a "connection grid," fixing the set of possible contact or via locations, and to reference the predefined layout to this grid. The grid spacing is the metal wire pitch. For our technology, the worst case is the via-to-via distance, which fixes the grid spacing to 8 lambdas. Fig. 1 partially shows the grid and possible contact and via locations. The use of such a grid has the side effect of an easy conversion to new design rules: only the new grid spacing needs to be adjusted, then the underlying layout can be redrawn.

6. Transistor Pitch: Most sea-of-gates structures are designed to offer the highest apparent gate density. For that purpose, the transistor pitch is fixed to the minimum, i.e., one grid spacing. In this case, vertical transparency is definitely impossible, and, for relatively complex cells, internal routing becomes a very hard problem, because diffusion and gate contacts are on the same vertical location. If the transistor pitch is fixed to two grid spacings, a full transparency can be achieved (this will be proved further). This choice, while decreasing the apparent gate density by a factor of 2, drastically improves the routability and connectability of the array.

7. Number of Contacts per Gate: The doubling of the gate contacts increases the polysilicon connectability by a factor of 2. It also reduces the vertical routing congestion, by decreasing the average vertical wire length.

8. Relative Transistor Sizes: Since the array is not particularly dedicated to dynamic circuits, nor to NAND or NOR logic, but to general complementary logic, the relative transistor size ratio is fixed to the inverse carrier mobility ratio.

9. Absolute Transistor Sizes: Fixing transistor sizes has effects on two different levels. At the electrical level, the available currents are proportional to the transistor sizes, making the circuit faster. But parasitic capacitances also increase, partially removing this advantage. (Fig. 2 shows the evolution of the simulated gate delay (inverter case) with the fan-out and the transistor size.) Paralleling two or more transistors is a lot more efficient than making them wider. At the geometric level, a trade-off between the number of horizontal routing tracks and a limitation of the basic cell area must be found. Finally, taking all these elements into account, the minimum sizes (as fixed by the connection grid and by the relative p/n transistor sizes) were chosen: one horizontal routing track crosses n transistors, and three cross p transistors. This, respectively, gives a W/L of 5 and 13 for each device. Eleven routing tracks are finally provided on one basic cell height, and speed requirements are satisfied.

10. Unalignment of the Gates: The unalignment of the poly gates from the grid is originally due to a technological reason: no via is permitted over polysilicon. But it has a useful side effect: without any loss of space (the grid does not move), an additional contact to diffusion is available between two successive gates. This doubles the diffusion connectability, and considerably simplifies internal routing problems.

11. Power Bus: The power bus must be wide enough to satisfy two requirements, i.e., avoiding electromigration, and allowing one track shifting even if neighboring loca-

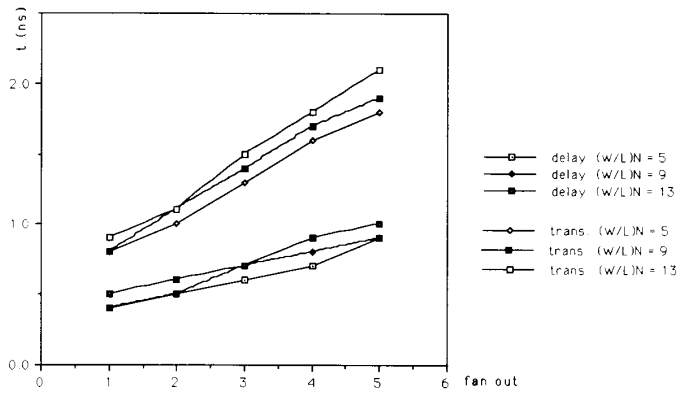


Fig. 2. Evolution of the inverter delay and transition time versus fan-out and transistor size.

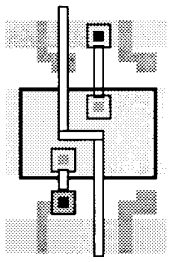


Fig. 3. Track shifting over power lines.

tions are occupied by power connections (Fig. 3). In analog circuits, currents are usually very small but sometimes more than two powers are needed. In that case, it is possible to split the power bus into two thinner lines, resulting in a supplementary power (or global signal).

C. Metallization Philosophy

The metal strategy used in typical sea-of-gates follows the same rules as older gate arrays: metal 1 is exclusively used for horizontal power buses and internal cell routing; it is also the main horizontal routing channel layer. Metal 2 is used for vertical overcell routing, for cell access, and as the second channel routing layer. Eventually a second power distribution can use vertical metal 2 buses [15]. The main consequence of this strategy is that the cells present only vertical transparency for metal 2, but no horizontal transparency at all (that is why channels are needed).

We propose a very systematic use of the two levels of metal: each direction is assigned to one level of metal. While this choice is not always the best for local routing problems, it yields a full transparency in both directions. In the next paragraph, it will be shown that a full abutment capability is then allowed. The global metal 1 direction is vertical and metal 2 direction is horizontal. So, metal 1 directly connects corresponding poly gates and diffusion drains. Power buses are in metal 2. This choice decreases the total contact and via count, as well as the area.

However, some exceptions are allowed to this rule: short local wires (such as a wire connecting two neighboring tracks), or regular circuit cells (since transparency is not a must, these cells being designed to naturally abut). Handcrafted layout may or may not follow it, but in the latter case, no automatic conversion to full custom will be possible (see Section V).

III. Circuit Implementation

A. Random Logic

As previously stated, only one of the two n diffusion strips is used for complementary random logic. The unused active area is normally covered by internal or external routing. Of course, these transistors could be used in dynamic circuits to reduce area, or in any circuit to increase the speed (but in this case, electrical symmetry between p and n devices is lost). Fig. 4 presents a typical random logic cell implementation example.

Although a standard cell implementation is always feasible, we focus here on only a full abutment implementation. Cell abutment requires four conditions: through-cell routing in both directions, choice of the aspect ratio, stretching, and choice of the connectors location.

1. Horizontal Through-Cell Routing is naturally allowed by transparency. The choice of horizontal wire location is free, provided that their total number is beyond the limit of 11. It has been shown [13] that ten horizontal tracks are sufficient to implement most CMOS logic circuits without extra routing.

2. Vertical Through-Cell Routing is not so obvious. First notice that it is generally not possible to vertically cross a cell over a gate location. Since the gates are placed, let's say, on odd grid x coordinates, vertical through-cell wires must be placed on even coordinates. If the track is free, the wire can be placed without conflict, thanks to the transparency. If it is not, it must be released. With this end in view, one can shift all transistors placed on the right of the track, one position to the right. The unconnected gates can be wired to a power. To respect electrical connectivity, one must short the source and drain of these transistors. Fi-

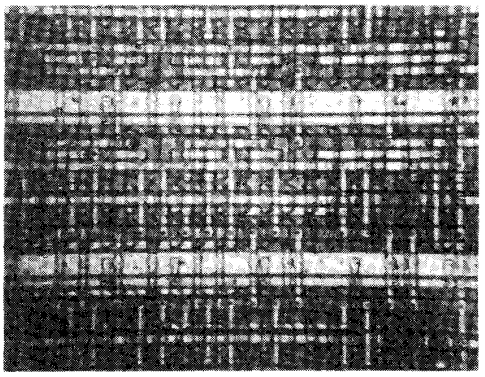


Fig. 4. Microphotograph of a static synchronous counter cell.

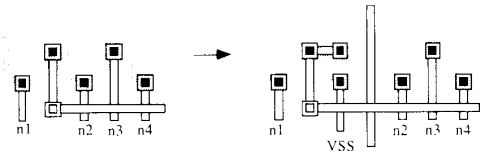


Fig. 5. Track release for vertical through-cell routing.

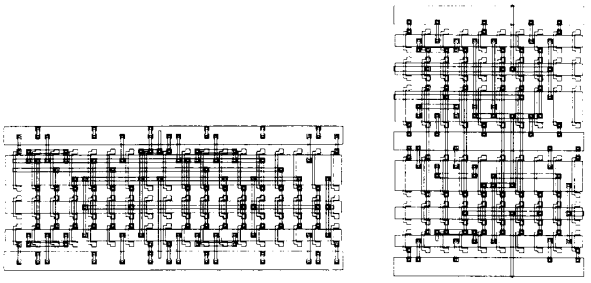


Fig. 6. Different aspect ratios for a full-adder cell.

nally, the desired track is released. This procedure is illustrated on Fig. 5.

3. Aspect Ratio: The height of a cell is not limited to one array slice. By using two or more slices, the cell is made higher and thinner (Fig. 6). The granularity of this process equals the basic cell height.

4. Stretching is obvious: after creating a gap, the electrical connectivity can be restored by shorting the two isolated extremities.

5. Connector Location: Thanks to transparency allowing a complete freedom for routing, any internal cell node can be routed to any location on the cell boundary.

It can therefore be concluded that abutment is possible in most cases, the only exception being horizontal congestion, that is, if the average number of needed horizontal tracks is over 11. In that case, one or more horizontal routing channels are required. An example of a module containing about 300 transistors, and realized without routing channels, is shown in Fig. 7. In this example, the transistor utilization rate (not counting unused diffusion strip) equals 81.5 percent.

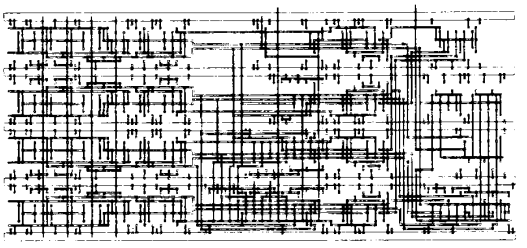


Fig. 7. Four-bit random logic ALU (only metal layers are represented).

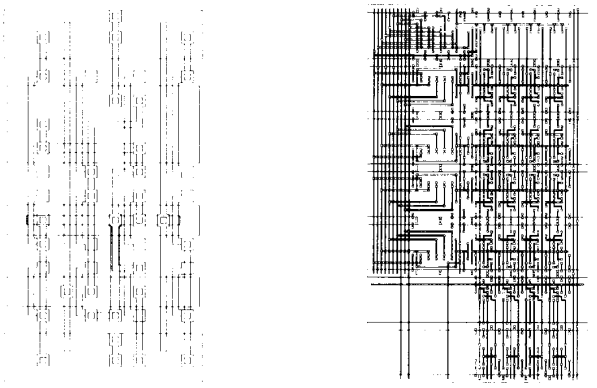


Fig. 8. (a) RAM cell detail. (b) RAM array (only metal layers are represented).

B. Regular Circuits

Two types of regular circuits can be distinguished. The first category contains RAM's, ROM's, and PLA's, whose basic cells are very specialized. The second category (multipliers, registers stacks, FIFO's) contains assemblies of logic cells specially designed for a good integration with neighboring cells. The latter sets the same problems as random logic, with more restrictive constraints on connector location. Here we focus on the implementation of the first category.

As already stated, the same basic array is used as for the random logic: this provides a better integration of the subcircuits at the system level, and suppresses any constraints on the size and location of the regular circuit. The continuity of the array architecture, the n/p transistor count ratio, and the transparency implied by the metallization strategy, considerably improve regular circuit implementation. Relatively high densities can be reached. The best example is the RAM cell (Fig. 8(a)), whose transistor utilization rate is 100 percent. PLA and ROM implementation rate reaches 66 percent, depending on circuit configuration. This lower number is due to the unused p transistor. On the other hand, one basic array cell implements two basic PLA or ROM cells. The regular circuit floorplans obey the same rules as their full-custom equivalents (Fig. 8(b)), and similar generation tools can be used for full-custom and sea-of-gates layouts.

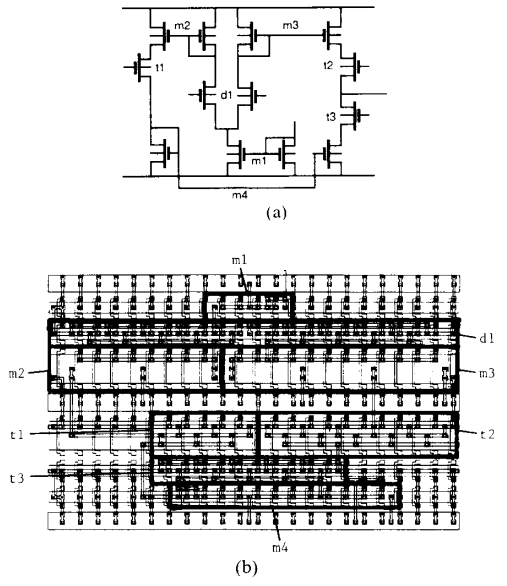


Fig. 9. (a) OTA schematics and (b) layout.

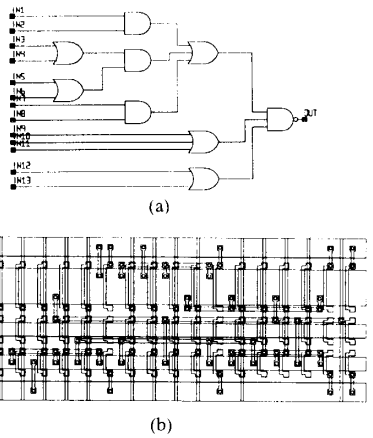


Fig. 10. (a) Complex AOI gate logic diagram and (b) layout.

C. Analog Circuits

Implementing analog circuits on a predefined array is a challenge, because lots of analog requirements seem incompatible with array features: analog circuits need big variations in the transistor size, a careful design of connections, several passive elements (mainly capacitances), etc. Our aim was not to design an analog array, but to find out how high-quality analog circuits could be implemented on a digital array.

Most analog circuits use a comb structure, because this pattern provides a high level of symmetry, and is an easy way of obtaining very large transistors. It has already been pointed out [12] that the regular comb structure offered by most sea-of-gates arrays was suited to the design of large (parallel association) or long (serial association) transistor channels. Moreover, the continuity of the diffusion layer suppresses all constraints on the size of the analog elementary structures (current mirrors, differential pairs, etc.). The embodied symmetry and regularity yield a good matching if the design is carefully done. The high level of routability helps limiting wire length and wire-to-wire capacitances. Experience has shown that analog circuits respecting the same floorplan rules as their full-custom equivalent were easy to design. Fig. 9 shows the typical example of an operational transconductance amplifier (OTA), in a cascoded version. A full 8-bit A/D converter was also realized.

The I/O pad ring can be used to implement analog circuit parts in two ways: first analog structures requiring very big W/L could use the I/O buffers transistors, and second, capacitors can be drawn on some unused I/O pad locations. For that purpose, polysilicon is deposited under each pad. A deposit of metal 1, and eventually of metal 2 over the poly creates functional capacitances ranging from 0 to 1.6 pF.

The remaining limitations to analog circuit implementation is the fixed device size, the relatively high source/drain diffusion capacitances, slowing down the circuit, and the lack of precise resistances.

IV. Design Environment

Although standard place-and-route CAD tools can be used with a standard cell library, a full use of the capabilities of the new structure requires application of new design automation strategies. Some tools more suited to the new layout philosophy are presented here. Several of these tools are already realized and commonly used; others are under realization.

As far as we know, random logic layout generation tools dealing together with semicustom array constraints (fixed device size and location, limited connectability and routability), and true channelless layout are not available yet. Since very specific algorithms could be developed, we believe that a good solution is to adapt existing full-custom design automation algorithms of the sea-of-gates layout template. As a more general principle, full-custom layout strategies (design hierarchy, top-down and bottom-up designs, etc.) should be more widely applied to semicustom arrays. For the tool review, three levels will be distinguished.

1. Cell Level: Our first sea-of-gates prototypes were manually designed. For the handcrafted layout design, a specialized symbolic editor has been developed. But automatic tools are also available. A good example of a sea-of-gates compatible layout generation algorithm is the Uehara–Van Cleemput algorithm for functional cell generation [14]. Fig. 10 presents the schematics and corresponding layout designed using this algorithm. Relaxation algorithms allow the generation of more complex cells and are also compatible with the array.

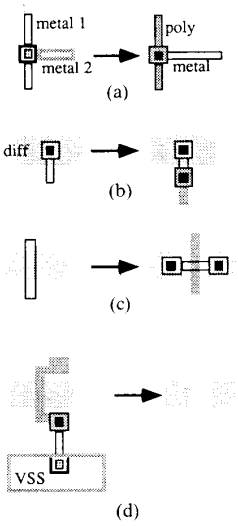


Fig. 11. Some sea-of-gates to full-custom transformation rules.

2. Module Level: Manual assembly by using a hierarchical graphical editor is currently used. A full-custom module generator [16], performing context-dependent cell generation and full cell abutment, has been adapted to the sea-of-gates array.

3. Chip Level: At this level, standard full-custom placement and routing strategies can be applied. Full-custom tools with only slight modifications will produce satisfying results.

Concerning regular circuits, specialized generators for RAM, ROM, and PLA, derived from full-custom generators, have been developed.

For analog circuits, a full-custom generation tool, for example [17], can also be adapted to the sea-of-gates structure. Only low-level layout generation procedures (transistor, differential pair, current mirror) need to be modified. But, in the common case of a mainly digital ASIC using a few analog functions, the number of different analog cells needed is limited. A cell library could be sufficient to cover most of the designer's needs.

V. Sea-of-Gates to Full-Custom Translation

Gate arrays or sea-of-gates designs often serve for testing and validating prototypes. When this is done, the problem of translating the design to a full-custom version compatible with mass production requirements arises. A full layout redesign is often necessary, and this is generally the source of new errors. A better solution would be the definition of an automatic procedure converting the semi-custom layout into a full-custom one.

Layouts designed on the described sea-of-gates array, and respecting the metallization rules given in Section II-C, allow such an automatic conversion. The procedure is composed of three steps.

1) First the sea-of-gates layout is decomposed into elementary symbolic elements (transistors, contacts, wires). Thanks to the metallization rules, only a restricted disposition of these elements will appear: for example, there will be no vertical metal 2 wires.

2) Then the set of rules given in Fig. 11 is applied; the result is a topologically similar symbolic layout.

3) Finally, this symbolic layout is compacted to produce a final full-custom version of the same circuit.

The target full-custom technology is a single-metal CMOS technology (however, the set of rules can be enhanced to use the second metal too). Global layer allocation is vertical for poly, and horizontal for metal. Transistors and diffusion wires keep their orientation. Thus vertical metal 1 wires are translated into vertical poly wires, and horizontal metal 2 wires are translated into

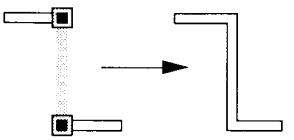


Fig. 12. Simplification rule.

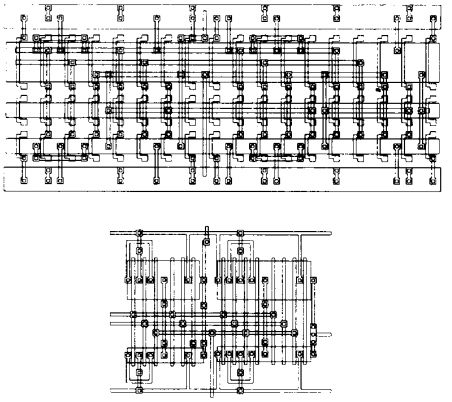


Fig. 13. Sea-of-gates to full-custom conversion example.

TABLE II
Evolution of Transistor Density with Cell Complexity

	Number of functional transistors	Area per transistor (λ^2)
Inverter / pass gate	2	1728
2 bits PLA cell	2	1296
2 inputs NAND	4	1296
1 bit SRAM cell	6	864
2 inputs XOR	6	1440
13 inputs complex AOI gate	26	1130
full-adder	28	987
multiplier cell	34	1067
4 bits carry-lookahed adder	128	1458
4 bits carry-lookahed ALU	284	1463

TABLE III
SURFACE COMPARISON BETWEEN DESIGN STYLES

	Symbolically generated layout	Sea of gates design	Gate Array design	Standard cell design	Full custom derived from sea of gates
Carry lookahead adder	100 %	213 %	261 %	249 %	107 %
Synchronous counter	100 %	182 %	258 %	229 %	119 %
Parallel multiplier	100 %	193 %	271 %	217 %	107 %
LIFO stack	100 %	219 %	-	243 %	111 %

TABLE IV
ELECTRICAL PERFORMANCE

	unit	sea-of-gates	full-custom	standard cells
DIGITAL				
basic gate delay (ring oscillator)	ns	0.5	-	-
adder 1 bit addition time	ns	3.5	-	8.8 - 13.4
adder carry propagation time	ns	2.8	-	1.9 - 6.0
PLA max. frequency	MHz	50	-	-
4 bits sync. counter max frequency	MHz	75	-	70
RAM access time : 16 bits	ns	3.8	-	-
RAM access time : 2K bits	ns	22.0	-	-
ANALOG				
OTA transconductance	uA/V	108.1±4	96.5±3.9	-
OTA offset voltage	mV	4.4±3.1	2.1±1.3	-
OTA linear range	mV	103	100	-

horizontal metal wires. The metal 1–metal 2 vias and metal 1–poly contacts are similarly converted (Fig. 11(a)). For metal 1–diffusion contacts, an additional short metal wire is required (Fig. 11(b)). When a metal 1 wire crosses a diffusion wire, a horizontal metal "bridge" is required (Fig. 11(c)). Of course, isolation transistors and unused transistors are removed (Fig. 11(d)). After application of these rules, some simplifications are performed. For example, a vertical poly wire connecting two metal contacts, and crossing no other wire, can be replaced by a simple metal wire (Fig. 12).

Fig. 13 presents an example of this automatic conversion. The resulting area reduction is generally about 50 percent. The area of the generated full-custom layout is about 10–20 percent more than a symbolically generated layout using the same topology.

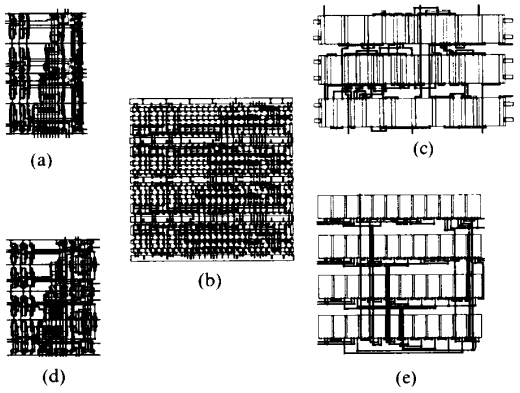


Fig. 14. Implementation of a 4-bit carry-lookahead adder by using different design styles: (a) symbolic full custom, (b) sea-of-gates, (c) standard cells, (d) symbolic full custom automatically generated from sea-of-gates, and (e) gate array.

VI. RESULTS AND PERFORMANCE

Table II shows resulting densities obtained by implementing different functions on the array. For low-level complexity cells, density is significantly lower than other arrays. On the other hand, one can observe that densities do not decrease much with complexity. This results from the absence of routing channels. Over a certain level of complexity, a density higher than that of classical arrays is expected.

Table III presents a comparison of silicon area consumption for different design styles and for several cells. In this table, a symbolically generated full-custom layout is taken as a reference, and its area is fixed to 100 percent. Other areas are given in percents. One can see that area consumption of the presented sea-of-gates is about twice that of the symbolic implementation. On the other hand, sea-of-gates area is generally smaller than the full-custom standard cell implementation, and of course than the gate array, too. Full custom derived from a sea-of-gates layout by using the above procedure give densities close to the reference layout (10–20 percent higher). Fig. 14 shows different design style implementations of a 4-bit carry-lookahead adder.

A test chip (Fig. 15) was integrated to check and validate the structure, as well as to perform some comparative measurements between sea-of-gates and full-custom circuits. Main performance results are summarized in Table IV. In spite of a relatively high source/drain parasitic capacitance, the basic gate delay is quite small. Performance of typical digital circuits is comparable to performance obtained with a standard cell design style. When comparing full-custom and sea-of-gates analog circuits, most characteristics are kept constant, except the offset voltage. This one is multiplied by a factor of 2, but stays in an acceptable range for most applications.

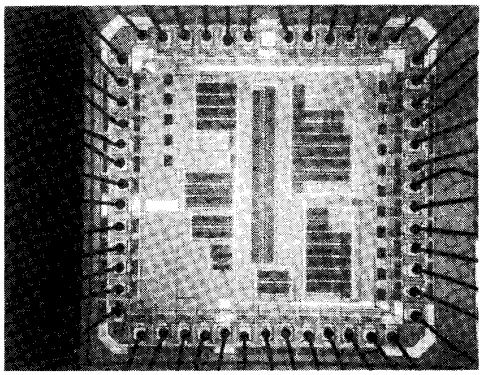


Fig. 15. Microphotograph of the test chip.

VII. Conclusion

Starting from a classical sea-of-gates structure, and adapting it to several new requirements, has lead to the definition of an original array. Its main distinctive features are: a larger transistor pitch to allow for vertical transparency, an optimized n/p transistor count ratio, the unalignment of transistor gates, and a systematic metallization strategy. Although the resulting structure can locally yield lower packing densities than other arrays, it drastically increases the integration rate of different classes of circuits. Powerful design styles, which were previously reserved to full-custom designs, are thereby applicable. Moreover, advanced design concepts, such as hierarchy, should be applied to take full advantage of the capabilities of the new structure. Implementation of regular and analog circuits was also examined. Thanks to the optimized n/p count, regular circuits yield satisfying densities. High-quality analog circuits have also been designed. Indeed, most of the analog requirements (symmetry, matching, capacitances) are met. Concerning design automation, several well-known algorithms, widely used in full-custom automated design, were found to be applicable to this array architecture. This will considerably ease the development of specific CAD tools. Finally, an automatic sea-of-gates to full-custom translation procedure was presented. While keeping the same circuit connectivity and topology, it converts a sea-of-gates design respecting the metallization rules into a compact full-custom version.

References

[1] E. E. Hollis, *Design of VLSI Gate Array IC's*. Englewood Cliffs, NJ: Prentice Hall, 1985.

[2] M. A. Beunder, J. P. Kernhof, and B. Hoefflinger, "The CMOS Gate Forest: An efficient and flexible high-performance ASIC design environment," *IEEE J. Solid-State Circuits*, vol. 23, no. 2, pp. 387–399, Apr. 1988.

[3] H. Fukuda, H. Yoshimura, and T. Adashi, "A CMOS pair transistor array masterslice," in *Proc. Symp. VLSI Technology*, 1982, pp. 16–17.

[4] A. Hui *et al.*, "A 4.1K double metal HCMOS sea-of-gates array," in *Proc. CICC*, May 1985, pp. 15–17.

[5] I. Ohkura *et al.*, "Gate isolation—A novel basic cell configuration for CMOS gate arrays," in *Proc. CICC*, May 1982, pp. 307–310.

[6] K. Sakoshita *et al.*, "A 10K-gate CMOS gate array based on a gate isolation structure," *IEEE J. Solid-State Circuits*, vol. SC-20, no. 1, pp. 413–418, Feb. 1985.

[7] W.-M. Van Noije and G. J. Declerck, "Advanced CMOS gate array architecture combining gate isolation and programmable routing channels," *IEEE J. Solid-State Circuits*, vol. SC-20, no. 2, pp. 469–480, Apr. 1985.

[8] F. Anderson and J. Ford, "A 0.5 micron 150K channelless gate array," in *Proc. CICC*, May 1987, pp. 35–38.

[9] H. Kubosawa *et al.*, "Layout approaches to high density channelless masterslice," in *Proc. CICC*, May 1987, pp. 48–51.

[10] T. Chan *et al.*, "Advanced structured arrays combine high density memories with channel-free logic array," in *Proc. CICC*, May 1987, pp. 39–43.

[11] K. Ueda, R. Kasai, and T. Sudo, "Layout strategy, standardisation, and CAD tools," in *Layout Design and Verification*, T. Ohtsuki, Ed. Amsterdam: North-Holland, 1986.

[12] S. Masuda *et al.*, "A CMOS analog and digital masterslice LSI," in *ISSCC Dig. Tech. Papers*, 1987, pp. 146–147.

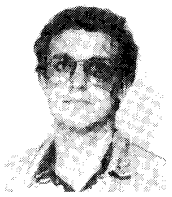
[13] R. Reis, R. Gomes, and M. Lubaszewski, "An efficient design methodology for standard cell circuits," in *Proc. ISCAS*, June 1988, pp. 1213–1216.

[14] T. Uehara and W. M. Van Clemput, "Optimal layout of CMOS functional arrays," *IEEE Trans. Computers*, vol. C-30, no. 5, pp. 305–311, May 1981.

[15] R. Blumberg and C. Waggoner, "Aluminium screen gives superior power distribution in a sea-of-gates array," *VLSI Syst. Des.*, vol. 9, no. 2, pp. 16–19, Feb. 1988.

[16] H. Heeb and W. Fichtner, "GRAPES: A module generator based on graph planarity," in *Proc. ICCAD*, Nov. 1987, pp. 428–431.

[17] M. Kayal, S. Piguet, M. Declercq, and B. Hochet, "SALIM: A layout generation tool for analog IC," in *Proc. CICC*, May 1988, pp. 7.5.1–7.5.4.



Philippe Duchene was born in Brussels, Belgium, on January 28, 1964. He received the M.S. degree in electrical engineering from the Université Catholique de Louvain, Louvain-la-Neuve, Belgium, in 1985.

From 1985 to 1987 he worked in the field of integrated circuit design in the Microelectronics Laboratory at the Université Catholique de Louvain. Since 1987 he has been a Research Assistant at the Electronics Laboratory of the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. His current interests are CMOS and BICMOS semicustom arrays and VLSI layout design automation.

Michel J. Declercq (S'70–M'72) was born in Brussels, Belgium, on January 28, 1945. He received the Doctor degree in electrical engineering from the Catholic University of Louvain, Louvain-la-Neuve, Belgium in 1971.

From 1971 to 1973 he was Professor at the University of Zaire, Kinshasa, where he was teaching electronic circuits and semiconductors physics. During the academic year 1973–1974, he was awarded a Senior Fulbright Fellowship, and was Research Associate at Stanford University, Microelectronics Laboratories, Stanford, CA. From 1974 to 1978 he was Research Associate at the Catholic University of Louvain, Belgium. In 1978 he joined Tractebel (Société Générale de Belgique) in Brussels, Belgium, where he was Group Leader of the Electronic Systems team. In 1985 he joined the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, where he is currently Professor and Director of the Electronics Laboratory.