A HIGH-DENSITY SEA-OF-GATES ARCHITECTURE INCORPORATING TESTABILITY SUPPORT

R.J.H. Koopman and H.G. Kerkhoff

MESA Research Institute
University of Twente
P.O.Box 217, 7500 AE Enschede, The Netherlands

Abstract - **In this paper a new CMOS sea-of-gates master array architecture will be presented. In this architecture not only routing considerations are taken into account, but also performance considerations. These routing and performance arguments have both been used to determine the transistor gate widths. In doing so a high-density sea-of-gates architecture has been designed with good routing and performance characteristics. Special hardware is incorporated in the presented architecture to support design for testability.**

1. Introduction

In the past, many sea-of-gates master array architectures have been presented [1,2,3,4]. The basic idea behind these arrays is to provide full-custom performance of VLSI systems while enjoying semi-custom advantages such as a short design and fabrication time. In the process of VLSI systems design it is important to keep the power consumption to a minimum, reduce the propagation delay within critical paths, construct high-density circuit and memory structures and be able to place all the required wiring nets in an efficient manner. The great challenge in the design of sea-of-gates architectures is to find a well-balanced solution for these in some cases conflicting requirements of the VLSI design process.

The following architectural solutions have been used to increase the design flexibility: applying a common-gate architecture to increase the density of memory structures and increasing the transistor gate widths to enlarge the wiring possibilities [1], adding n-channel MOS transistors to support dynamic logic and memory design [2,3], and supporting a two-dimensional design approach [3,4]. These arrays are excellent in the intended application fields, but usually fail to provide optimized designs beyond these fields. If an architecture explicitly supports the design of high-density memory structures, this same architecture results in a low density of non-memory structures. Some parts of the design can be designed using a static logic implementation to reduce the power consumption. If transistors have been added in the array to support dynamic logic, the overall density of these static logic structures will be reduced too.

The sea-of-gates design style is often associated with flexible routing through and over logic cells without using predefined wiring channels. The area occupied by these added or enlarged transistors in the above mentioned arrays, is often justified by the increase of transparency and thereby supporting this design style. Again this built-in transparency reduces the design efficiency. Within some chip areas this available space may not be necessary to solve the routing problem, where at other areas this transparency may not be enough to complete the routing problem. On account of the emphasis on the routing in these arrays, less attention is often given to performance considerations and the determination of the transistor dimensions.

In this paper a sea-of-gates architecture will be described which more suits the demands of VLSI systems design. This architecture is not only based on routing aspects, but also on performance aspects. The key feature of this architecture is the increase of the transistor density through a reduction of the area overhead, the application of small-sized transistors and the exclusion of dedicated transistors or transistor structures. The use of small-sized transistors results in high-density circuit implementations and a reduction of the power consumption. Delay and routing problems can be solved by using additional transistors. To support design for testability special hardware is incorporated in the architecture.

In the next section more will be said about the performance and the routing considerations the architecture has been based on. The third section describes the resulting architecture itself. In the last section some remarks will be made about the incorporated testability support.

2. Performance and Routing Considerations

In this section some design arguments will be discussed concerning performance and routing. These arguments have been used to tune the architecture. The architecture is based on the following two elementary rules:

1. performance rule: do not use large-sized transistors if the same performance can also be obtained with small-sized transistors (performance considerations are mainly concentrated to speed and power because they are often considered as the most important performance exponents),
2. wiring rule: do not provide excessive wiring provisions if they are not needed.

2.1. Performance

Within sea-of-gates arrays all n-channel MOS transistors have the same transistor gate width W_n and all p-channel MOS transistors have the same gate width W_p. These W_n and W_p values are characteristic quantities of the array. The transistors need a sufficient transistor gate width to meet a desired speed level. The speed will be affected by several capacitances. A distinction is made between the following capacitances:

- parasitic transistor capacitance,
- intra-cell wiring capacitance,
- inter-cell wiring capacitance,
- transistor gate capacitance from a fan-out cell.

In general, large capacitances require large-sized transistors to charge and discharge. However, without any parasitic and wiring capacitance no speed differences will occur between small-sized and large-sized transistors. This is illustrated in Figure 1 which shows the relation between propagation delay and transistor width if only the

0-7803-0593-0/92 $3.00 1992 IEEE

transistor gate capacitance from fan-out cells is taken into account. As an example a CMOS full adder cell has been used with $W_p = (11.5/6.5)*W_n$. The number of load inverters is used as a parameter in the figure. The transistors of these inverters have the same widths as those used in the full adder.

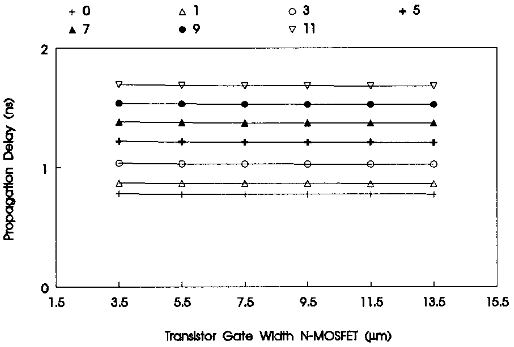

Figure 1. Relation between propagation delay and transistor width if only the load capacitance from fan-out cells is taken into account.

If parasitic transistor, intra-cell wiring and inter-cell wiring capacitances to nearby logic cells are taken into account, large-sized transistors will have a reduced propagation delay compared to small-sized transistors. However, if the transistor sizes are made excessively large, the transistor gate capacitance from fan-out cells would be dominant over the other capacitances. The propagation delay will not be reduced by a further increase in the transistor gate width. This relation is shown in Figure 2. The same CMOS full adder as in Figure 1 has been used, but now all parasitic transistor and intra-cell capacitances have been considered. To obtain inter-cell wiring capacitance, the output has been loaded additionally with 20fF.

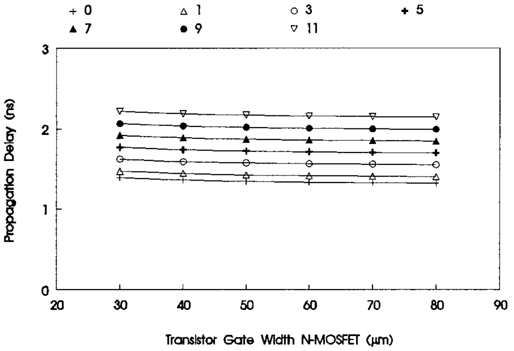

Figure 2. Relation between propagation delay and transistor width if all capacitances are taken into account.

The situation as illustrated in Figure 2 is highly undesirable because the increase of transistor width results in a higher power consumption without a further reduction of the delay. Furthermore, not every logic cell in a design needs to possess the minimum delay and for these cells a reduced power consumption is more important than speed.

In the architecture, the typical logic cell intra-cell wiring capacitance and the typical inter-cell wiring capacitance due to interconnect to nearby logic cells, have both been used to determine the minimal transistor gate width. A further increase of the transistor width will not result in a dramatically reduced delay where as the power consumption for all logic cells will be increased. If a higher load capacitance is encountered than has been assumed in the design of the array based on the assumptions above, two techniques are available to improve the speed. First, a performance-driven micro-cell compiler can be used to reduce the delay of micro-cells [5]. A micro-cell is defined as a small logic cell with one of the most complex being a full adder. By putting transistors in parallel within a micro-cell the fan-out capability is increased. Second, a logic synthesis procedure based on buffer-insertion techniques can be used.

2.2. Wiring

To reduce excessive wiring provisions, small-sized transistors have to be used. On the other hand, the area above the transistors must be sufficiently large to contact the source, drain and gate terminals. Smaller sized transistors would result in very poor connectability while wider transistors would only increase the routing space.

When designing complex micro-cells or dealing with complex routing situations, routing space can be allocated above a transistor row. However, if the allocated space above transistors is very large and only a few tracks are required, silicon area is wasted. Hence, the array architecture must provide a small track increment. This small track increment can be realized with small-sized transistors and this corresponds with the performance as well as the wiring rule.

3. The Sea-of-Gates Architecture

In this section the resulting sea-of-gates architecture will be described. A distinction will be made between the core-cell (local) architecture and the master array (global) architecture. A core-cell is the smallest repetitive unit of the sea-of-gates array, and consists of p-channel and n-channel MOS transistors. The master array embeds the overall arrangement and placement of the core-cells.

The core-cell consists of one pMOS and one nMOS transistor. Figure 3 shows the arrangement of four core-cells. Adjacent core-cells (transistors) share common diffusion areas. To reduce area overhead every pMOS transistor has a polysilicon gate contact under the first metal V_{dd} rail, and every nMOS transistor has a polysilicon gate contact under the first metal V_{ss} rail. These squares can only be used to connect the transistor gates to the V_{dd} rail and the V_{ss} rail respectively.

From both the performance rule and the wiring rule, transistor gate widths are derived. These two values are used to determine the final transistor gate width. In this way, the optimum transistor width can be found with respect to speed, power consumption, connectability and density under the given assumptions.

To be able to make contact between the first metal layer and the diffusion area of a transistor, at least space for one contact has to be available above the transistor diffusion area. To switch from the first metal layer to the second metal layer without using stacked contacts, space for a via between these two layers must also be available above the diffusion area. If there is no available area for this via, the via must be placed above the polysilicon transistor gate contact or the diffusion area of a nearby transistor. Both options are not desirable because they block the usage of a transistor. So the transistor gate has to be at least as wide as the space needed for both these two contacts. To adjust for the electrical difference between a p-channel and an n-channel transistor, the width of the pMOS has been made one contact position larger. The n-well and substrate contacts are situated under the V_{dd} and V_{ss} rail.

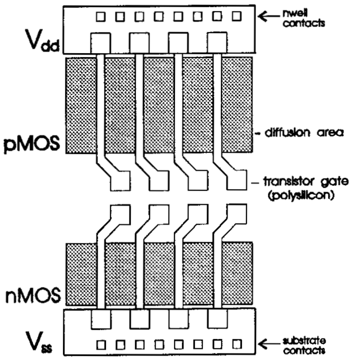

Figure 3. Four core-cells.

Table 1 shows some core-cell characteristics. To estimate the logic density, a 2-input NAND gate has been used. This NAND gate occupies 2.66 core-cells including the waste of transistors used for transistor gate isolation.

core-cell dimensions	39.75μm x 5μm
core-cell area	198.75μm^2
width pMOS	W=11.5μm
length pMOS	L=1.0μm
width nMOS	W=6.5μm
length nMOS	L=1.0μm
W_p / W_n	1.77
logic density	1.89k gates/mm^2
ROM density	6.7k cells/mm^2
sRAM density	628 cells/mm^2

Table 1. Core-cell characteristics.

To minimize parasitic transistor capacitances, the horizontal distance between two transistor gates is kept at a minimum by using common source/drain areas. The transistor row height is decreased by sharing first metal common power and ground rails between rows. A further reduction of the transistor row height has been achieved by placing a pMOS gate contact under the power rail and an nMOS gate contact under the ground rail. Another reduction of the core-cell dimensions has been achieved by not allowing a via above the polysilicon transistor gate contact if this via needs an extension of that polysilicon gate contact. Most polysilicon gate contacts will be used for the positioning of a contact between the first metal layer and the polysilicon layer and will not need an extended contact. Doing so also minimizes area overhead and increases the density.

The master array architecture is build by repeating and abutting the core-cells in the horizontal and vertical direction, whereby every core-cell is surrounded by four other core-cells. In the horizontal direction (to the left and right), the adjacent core-cells are identical copies. In the vertical direction (above and below), the core-cells are mirrored around the V_{dd} and V_{ss} rails. In this way a row-based master array floorplan is formed, with the transistors arranged within these rows.

The routing grid can be considered as a signature of the core-cell layout. Figure 4 shows this routing grid structure of the presented architecture.

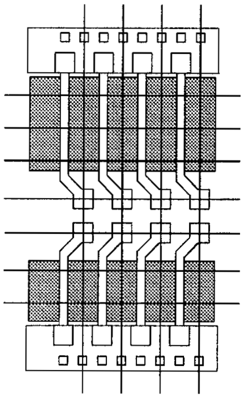

Figure 4. Routing grid structure of the presented architecture.

This grid structure shows that above a core-cell, seven horizontal grid lines and one vertical grid line exist. At every grid point a contact from the first metal layer to the polysilicon or diffusion layer can be placed without blocking the surrounding grid points or grid lines. As already mentioned, the placing of a via will be forbidden on top of the polysilicon grid points as this will require an extended polysilicon contact.

The main advantage of the architecture is that under typical conditions high-density micro-cells, memory and other regular circuits can be designed. As an example of a micro-cell design a full adder has been designed. Figure 5 shows the layout of this full adder.

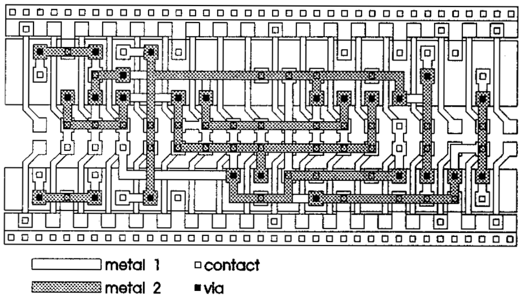

Figure 5. Layout of a full adder.

Under non-typical conditions speed problems can easily be solved by invoking parallel transistors and buffers, and routing problems can be solved by allocating the needed space above core-cell rows.

The architecture has been designed to be process independent. In this way, the architecture is portable between different processes and is also scalable within a process. This is very important because finished chip designs can easily be transformed to a down-scaled process. Also for second-sourcing it is very important to be process independent. In the past some experiments have been carried out with non-row-based sea-of-gates architectures [4]. These core-cell layouts proved to be very process dependent.

The architecture has been realized in two different dual-layer metal CMOS processes, one being a 1.0 μm industrial process (see Table 1), the other being the 2.5 μm MESA process. The MESA process allows the usage of stacked contacts by using selective tungsten for the first contact opening [6]. Figure 6 shows a chip photograph of a part of a full adder designed in the MESA process using stacked contacts.

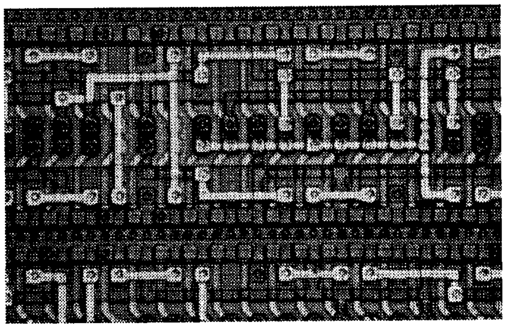

Figure 6. Chip photograph of a full adder layout. .

4. Testability Support

During the design of a VLSI system it is very important to be able to test the chips, not only after production, but also after having been placed on a printed-circuit board. Therefore the sea-of-gates design style should enforce testability during the design of VLSI systems, and hence sufficient support has to be incorporated in the sea-of-gates architecture.

The testability support is based on the IEEE Standard Test Access Port and Boundary-Scan Architecture [7,8]. Hardware related to this standard can be divided into four parts:

- the Test Access Port (TAP),
- the TAP controller,
- a scannable instruction register,
- a group of scannable test-data registers.

The TAP is a general-purpose port which provides access to the test logic of an integrated circuit. The TAP controller generates clock and control signals for the instruction and test-data registers and other parts of the boundary-scan architecture. The instruction register allows a particular instruction to be shifted into the chip under test. This instruction is used to select the test to be performed and/or the test-data register to be accessed. The group of test-data registers must include a bypass and a boundary-scan register. It may also include an optional device identification register and further optional user defined test-data registers.

To be able to control and observe a primary chip input or output, a boundary-scan register cell is added to every input and output. All these cells together form the boundary-scan register. The boundary-scan cells are incorporated in the floorplan as fixed hardware blocks. In addition, every boundary-scan implementation always requires the TAP, the TAP controller and at least two instruction register cells and the bypass register. This boundary-scan logic is also incorporated in the floorplan.

Layouts of an instruction register cell and a device identification register cell are available in the library. These layouts can be used during the design to enlarge the instruction register and to incorporate a device identification register.

The main advantage of the testability support is the availability of basic test hardware on-chip. Layouts of necessary test logic has been provided in the library to implement on-chip tests. Furthermore, the chip will be printed-circuit board testable according the IEEE boundary-scan standard. Drawbacks are the same as with other test approaches such as area overhead, reduced system performance, complicated routing and the use of extra I/O pins.

5. Conclusion

A new sea-of-gates architecture has been presented which is based on routing arguments as well as on performance aspects. This results in an architecture where specific speed and routing problems can be solved without reducing the overall circuit density. The key feature of the architecture is a high transistor density, without dedicated transistors or transistor structures. In this way more flexibility has been incorporated in the architecture. Therefore the presented architecture is more dedicated to VLSI design than existing sea-of-gates architectures. Furthermore, the proposed architecture supports testability according to the IEEE 1149.1 boundary-scan architecture.

Acknowledgments

The authors would like to thank Y. Xing for his valuable suggestions and discussions. This work has been carried out with the support of the Dutch Innovative Research Program (IOP IC-TEL 88.038).

References

[1] H.J.M. Veendrick, D.A.J.M. van den Elshout, D.W. Harberts and T. Brand, "An Efficient and Flexible Architecture for High-Density Gate Arrays", IEEE Journal of Solid-State Circuits, vol.25, no.5, October 1990, pp.1153-1157.

[2] P. Duchene, M.J. Declercq, "A Highly Flexible Sea-of-Gates Structure for Digital and Analog Applications", IEEE Journal of Solid-State Circuits, vol.24, no.3, June 1989, pp.576-584.

[3] M.A. Beunder, "Design and Analysis of Semi-Custom Architectures", Ph.D. Thesis, Institute for Microelectronics Stuttgart, West-Germany, December 1989.

[4] R.J.H. Koopman, R. Peset LLopis and H.G. Kerkhoff, "Digital CMOS Sea-of-Gates Core Cells and Master Images", European Solid-State Circuit Conference, Grenoble, France, September 1990, pp.245-248.

[5] R. Peset LLopis, R.J.H. Koopman, H.G. Kerkhoff and J.A. Braat, "A Performance Analysis Tool for Performance-Driven Micro-Cell Generation", European Design Automation Conference, Amsterdam, The Netherlands, February 1991, pp.576-580.

[6] A. Hasper, Ph.D. Thesis, MESA Research Institute, Enschede, The Netherlands, June 1992.

[7] IEEE Standard 1149.1-1990, IEEE Standard Test Access Port and Boundary-Scan Architecture, IEEE Standards Board, 345 East 47th Street, New York, May 1990.

[8] R.P. van Riessen, H.G. Kerkhoff and A. Kloppenburg, "Designing and Implementing an Architecture with Boundary Scan", IEEE Design and Test of Computers, vol.7, no.2, February 1990, pp.9-19.