A MODULE–SLICED APPROACH FOR HIGH YIELD VLSI/WSI PROCESSORS

Yi-Chieh Chang and Kang G. Shin

Real-Time Computing Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109–2122

ABSTRACT

Low yield is one of the practical difficulties in the design of large VLSI/WSI chips, such as 32-bit microprocessors. Since the yield of a module declines with the increase of its size, one can improve the yield by decomposing a large module into several smaller submodules, or using the module-sliced approach. The submodules will collectively perform the function of the original large module.

The module-sliced approach is realized in a reconfigurable fault-tolerant segmented array processor (RFTSAP). The basic building block of RFTSAP is a node which consists of a processor, local memory, and a programmable I/O unit (PIOU). The PIOU allows any groups of processors to be combined to perform the functions of a large processor module. The yield of a large processor module can be improved 2 to 4 times compared to the approaches without using the module-sliced method.

1 Introduction

Low yield is one of the practical difficulties in the design of large VLSI/WSI chips, such as 32-bit microprocessors. Although introducing redundancy for yield improvement has been reported to be successful in memory systems [1, 2, 3], little has been done on the applicability of this approach to processor modules due mainly to their complexity and irregularity [4]. Since the yield of a module declines with the increase of its size, one can improve the yield by decomposing large modules into smaller modules, or using the *module-sliced approach.* For example, the function of a processor with data width of 32 bits can be obtained by combining 4 small processors each with data width of 8 bits.

The key issue in the module-sliced approach is how to interconnect fault-free submodules to perform the function of the original, large module. The required interconnections should not occupy too much an area and the reconfiguration algorithm should be able to restructure any group of fault-free submodules into the original module. A reconfigurable fault-tolerant segmented array processor (RFTSAP) is proposed as a means of improving the yield of VLSI/WSI processors without inducing the high interconnection overhead. The basic module of RFTSAP is a node which consists of a processor and a programmable I/O unit. Each programmable I/O unit has four I/O ports and six programmable switches. The programmable I/O unit allows the processor on an node to be combined with other processors in order to perform the function of a large processor. Moreover, a faulty processor can be bypassed by activating one of the six programmable switches in the corresponding I/O unit. These switches are controlled by the processor, if not faulty, of the corresponding node, or by one of it's nonfaulty neighboring processors if the node processor is faulty.

The rest of this paper is organized as follows. Section 2 describes the RFTSAP architecture and presents the reconfiguration method. Section 3 analyzes the merit of the module-sliced approach. This paper concludes in section 4.

2 RFTSAP Architecture

A node of the RFTSAP consists of a processing element (PE), a PIOU, and local memory. The structure of a node is shown in Fig. 1. The computing power of each node resides in its PE which is a conventional microprocessor. The key features of RFTSAP are provided by the PIOU in Fig. 2. It has four I/O ports, denoted by U_port, D_port, L_port, and R_port, to which four immediate neighbors located in the up, down, left, and right directions are connected. Local memory is provided at each node to store configuration settings and application programs such that a PE can execute programs and dynamically reconfigure the PIOU at run-time.

In addition to the four I/O ports, each PIOU includes a status register (SR) and a reconfiguration register (RR), six programmable switches $S_1, \ldots, S_6$, and two command

CH2794-6/89/0000/0500$01.00 © 1989 IEEE

decoders. The I/O ports are used for inter-node communications. The SR is used to indicate the current state of the PIOU which would be in one of two possible states: *local_control* and *remote_control*. If a node processor is non-faulty, it will control the corresponding PIOU locally, placing the PIOU in local_control state. If the node processor is faulty, the PIOU will be placed in remote_control state and thus controlled by one of its neighboring nodes. The RR is used to store control signals, $C_1, \ldots, C_6$, generated by a PE to activate one of the six programmable switches. Each programmable switch can be activated to close (open) two I/O ports. For example, when L_port is connected to R_port the signal can propagate through them with a negligible delay such that $PE_{i,j-1}$ is connected to $PE_{i,j+1}$, thus virtually bypassing $PE_{i,j}$. These programmable switches are combined into two groups : $\{S_1, S_2, S_3\}$ and $\{S_4, S_5, S_6\}$. There are two command decoders, each of which is associated with one of these groups. Whenever a switch is closed, the corresponding command decoder will be enabled to decode the message which passes through the switch and will respond to a *global operation* command (e.g., broadcast).

The PIOU and interconnections between nodes are assumed to be fault-free since these circuitries are much simpler than a processor, and, due to their simplicities, redundant circuits can be added in each PIOU to improve the yield. Since faulty processors may randomly distributed on an array of processors, the PIOU allows a faulty processor to be bypassed by properly setting its programmable switches. When a group of processors are connected to perform the functions of a large processor module, one of the processors in the group will control the operation of other processors. This is done by broadcasting a command or instruction to other processors. One example of reconfiguring processors is shown in Fig. 3.

3 Merit of Module-Sliced Approach

The yield is reported to decline exponentially as the size of module increases, though there is a recent study showing that the yield falls less than exponential [5]. Generally, the yield can be estimated by Eq. (3.1) when it is less than 30% [6], or by Eq. (3.2) when it is greater than 30% [7].

$$Y = e^{-\sqrt{AD}} \tag{3.1}$$

$$Y = \left(\frac{1 - e^{-AD}}{AD}\right)^2 \tag{3.2}$$

where A is the module's area and D the defect density which is around 4 defects/cm^2 [8].

Since the yield of a module declines with the increase of its size, one can improve the yield by decomposing large modules into smaller modules, or using the module-sliced approach. For example, the module-sliced approach can be applied to VLSI/WSI array processors, since the function of a processor with data width of 32 bits can be obtained by combining 4 small processors each with data width of 8 bits. When a module is divided into several submodule, a certain number of interconnections must be provided for intermodule communications. (They must communicate with each other and collectively perform the function of the original module.) These interconnections can be expressed as an area overhead of the module-sliced approach. The merit of the module-sliced approach can be measured by the *gain factor* (GF) which is the ratio of the number of equivalent fault-free processors that can be obtained with the module-sliced approach to that obtainable from an approach without module decomposition.

To derive the optimal module decomposition, we need to analyze the relation between the yield and the module size. Let A_o (Y_o) and A_{sm} (Y_{sm}) be the areas (yield) of the original module and a submodule, respectively. Assuming the total wafer area to be A_w, the number of fault-free modules on the wafer that can be obtained is:

$$\begin{aligned} N_o &= \frac{A_w}{A_o} \times Y_o \\ N_{sm} &= \frac{A_w}{A_{sm}} \times Y_{sm}. \end{aligned} \tag{3.3}$$

Since the original module is divided into several submodules, A_{sm} can be expressed as:

$$A_{sm} = \frac{A_o}{\alpha} + A_i, \tag{3.4}$$

where A_i is the area required for interconnecting submodules and α the granularity factor. (A larger α means a finer granularity). Let N_o and N_{sm} be the number of fault-free modules and submodules on a wafer, respectively. Then, GF can be derived as:

$$GF = \frac{N_{sm}}{N_o \, \alpha} = \frac{A_o}{A_{sm} \, \alpha} \frac{Y_{sm}}{Y_o}. \tag{3.5}$$

The number of equivalent fault-free processors that can be obtained with the module-sliced approach is GF times that obtainable from an approach without module decomposition. To calculate GF, it is necessary to determine A_i, the only unknown variable in A_{sm}. Since A_i is the area required for interconnecting submodules, it can be approximately expressed as:

$$A_i = \beta \, A_{sm} + k, \tag{3.6}$$

where β and k are some constants. The first term represents the part of A_i which depends on the granularity of submodule and is proportional to the total area of submodule for such components as data paths and buses. The second term is a constant area needed no matter how small a submodule is, e.g., control paths and other logic circuits. Then, we have

$$A_{sm} = \frac{1}{\alpha(1-\beta)} A_o + \frac{k}{1-\beta}. \qquad (3.7)$$

The values of β and k reflect the degree of interconnections required by the module-sliced approach; larger values imply larger interconnection overheads. Given β and k, one can calculate A_{sm} by using Eq. (3.7), and GF can then be computed by Eq. (3.5) with various values of α. Since the values of β and k depend on actual design, the relation between GF and α is plotted for several different values of β and k in Figs. 3 and 4. As shown in Fig. 3, the larger α, the higher GF results. This agrees well with our intuition that a smaller interconnection overhead favors a finer granularity. However, in most curves, the maximum GF occurs at a specific value of α, implying the existence of an optimal decomposition. Since β and k are design parameters, we decided to find ranges of their values that result in high GFs. Assuming A_o to be 1.0 cm^2 which is about the size of a modern 32-bit microprocessor, A_i is compared with A_{sm} and the interconnection overhead $((A_{sm} - A_i)/A_i)$ is found to be 20 to 30% for $\beta \leq 0.1$ and $k = 5mm^2$, 35 to 50% for $\beta \leq 0.1$ and $k = 10mm^2$, and over 50% for $\beta > 0.2$, $k > 5mm^2$, for a wide range of α. However, GF is found to be nearly unity (i.e., no improvement), if the interconnection overhead is over 50%. This result indicates that a higher overhead of interconnections in an array processor will not improve the yield of processors by the module-sliced approach. In the proposed RFTSAP, only one I/O link is required between any two processors in each direction; the interconnection overhead is less than 30%, and thus, GF ranges from 2 to 4 can be achieved.

4 Conclusion

A module-sliced approach is proposed to improve the yield of large VLSI/WSI processors. This approach is realized in a RFTSAP structure. The programmable switches and segmented links of RFTSAP allow faulty processors to be bypassed without requiring extra I/O interconnections and multiplexers. Since a small amount of overhead is introduced in RFTSAP, the merit of module-sliced approach ranges from 2 to 3.

References

[1] R. P. C. et al., "A fault-tolerant 64k dynamic random-access memory," *IEE Trans. Electron Devices*, vol. ED-26, pp. 853-860, June 1979.

[2] B. F. Fitzgerald and E. P. Thoma, "Circuit implementation of fusible redundant address on rams for productivity enhancement," *IBM J. Res. Devlop.*, vol. 24, pp. 291-298, May 1980.

[3] Y. Ueoka, C. Minagawa, M. Oka, and A. Ishimoto, "A defect-tolerant design for full-wafer memory lsi," *IEEE J. Solid-State Circuits*, vol. SC-19, pp. 319-324, June 1984.

[4] I. Koren and D. K. Pradhan, "Yield and performance enhancement through redundancy in vlsi and wsi multiprocessor systems," *Proc. IEEE*, vol. 74, no. 5, pp. 699-711, May 1986.

[5] J. E. Price, "A new look at yield of integrated circuits," *Proc. IEEE*, vol. 58, no. 8, pp. 1290-1291, August 1970.

[6] B. T. Murphy, "Cost-size optima of monolithic integrated circuits," *Proc. IEEE*, vol. 52, no. 12, pp. 1537-1545, December 1964.

[7] R. B. Seeds, "Yield and cost analysis of bipolar lsi," *Proc. IEEE Int. Electron Devices Meeting*, p. Paper 1.1, October 1967.

[8] N. Weste and K. Eshraghian, *Principles of CMOS VLSI design a systems perspective*, Addison Wesley, 1985.

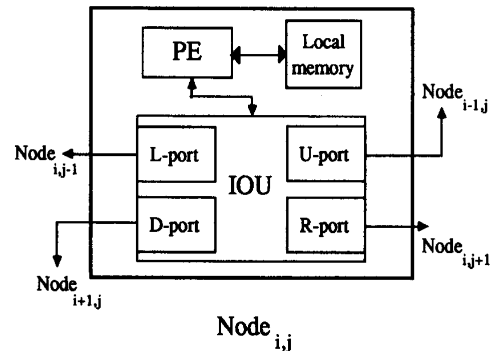

Figure 1: Block diagram of node (i, j).

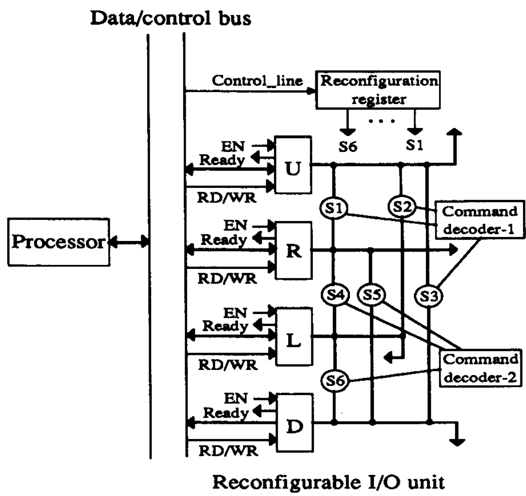

Figure 2: The block diagram of an IOU.

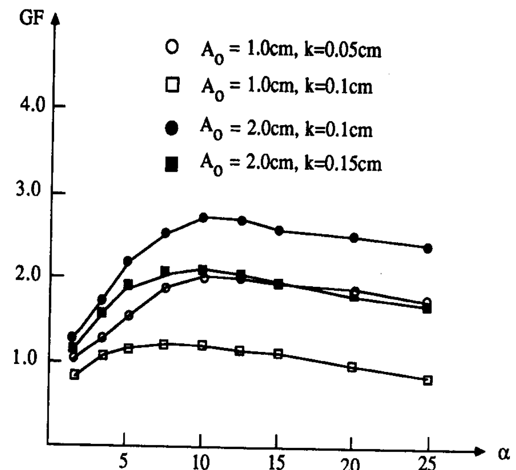

Figure 4: Plot of Gain factor ($\beta = 0.2$).

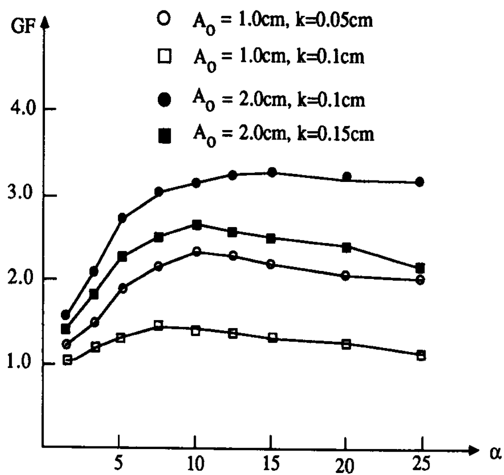

Figure 3: Plot of Gain factor ($\beta = 0.1$).