VLSI/WSI Designs for Folded Cube-Connected Cycles Architectures

M. P. Sebastian, P. S. Nagendra Rao, Lawrence Jenkins
Electrical Engineering Department, Indian Institute of Science
Bangalore - 560 012, India.
{mpsebas, nagendra, lawrn @ee.iisc.ernet.in}

Abstract

This paper presents VLSI/WSI designs for a recently introduced parallel architecture known as the folded cube-connected cycles (FCCC). We first discuss two layouts for the FCCC, in which there is no component redundancy. Then we incorporate redundancy, and present locally and globally reconfigurable FCCCs. We also discuss the design of universal building blocks for the construction of fault-tolerant FCCCs of various dimensions.

Index Terms- Building block design, layout, fault-tolerance, parallel architecture, reliability, yield.

1 Introduction

The cube-connected cycles (CCC) [7] is a popular parallel architecture, not only because it preserves most of the attractive properties of the hypercube, but also because it has degree boundedness of nodes and an efficient VLSI/WSI implementation [3], [5], [9], and [10]. The folded cube-connected cycles ($FCCC$), discussed in [6] and [8], is more attractive than the CCC, as the former has smaller diameter, lower average distance, and higher fault tolerance. The n-dimensional folded cube-connected cycles, $FCCC_n$, is the composition of an n-dimensional folded hypercube [2], with an $(n+1)$-cycle; i.e., an n-dimensional folded hypercube with each of its vertices replaced by an $(n+1)$-cycle. The new vertices (each of degree 3) are then labeled by a pair (Z, i), where Z is an n-bit vector that corresponds to the original folded hypercube node label, and i is a decimal quantity that takes values from 0 to n. The vertex (Z, i) is adjacent to the vertices $(Z, i+1)$, $(Z, i-1)$, and $(Z^{\bar{i}}, i)$, where (i) $i+1$ and $i-1$ are computed cyclically (i.e., $(i+1) mod(n+1)$ and $(i-1) mod(n+1)$ respectively), and (ii) Z and $Z^{\bar{i}}$ differs only in the i-th bit (with $Z^{\bar{n}} = \bar{Z}$). Figure 1 shows a 3-dimensional $FCCC$, where the nodes labeled with decimal values.

In this paper, we try to establish that the $FCCC$ network is realizable using VLSI/WSI, and that all such designs for the CCC are extendable to the $FCCC$ as well. We assume a rectangular grid model as in [1], which uses two layers of evenly spaced horizontal and vertical wires. Processing Elements (PEs) are placed at grid points. Wires running along the grid lines that meet at a grid point occupied by a processing element (PE), represent the input/output links of that PE. Design efforts are focused on improving the reliability, yield, and fault tolerance. The approach is to incorporate some redundancy in the target structure and employ an appropriate reconfiguration scheme. Since area considerations have an important impact on reliability and yield, the area overhead due to the addition of redundant elements should be as small as possible. Hence, compact layout strategies are highly desirable for both redundant and non-redundant structures.

We present several layouts for the $FCCC$. For each of them, we derive expressions for the area, yield, and reliability. The rest of the paper is organized as follows. In Section 2, we propose two layouts for the $FCCC$ (without redundancy). In Section 3, we discuss reconfigurable $FCCC$ networks, and in Section 4, we present a building block approach.

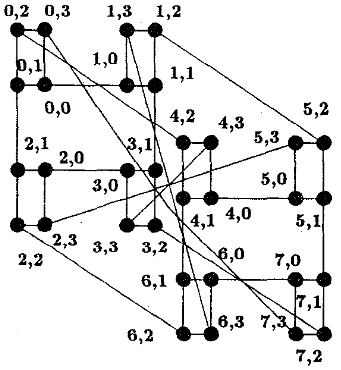

Figure 1: $FCCC_3$.

2 Layouts with no redundant components

We now propose two layouts for the $FCCC$. The first layout is a conventional one, and has been evolved using the strategy given in [7]. It is shown in Figure 2. Let all dimensions be specified in terms of a length unit λ. The actual layout area depends on the ratio m of the width of a PE to the width of a link. Assume the width of a link as $w\lambda$. There are 2^n cycles in $FCCC_n$,

1063-9667/95 $04.00 © 1995 IEEE

and each cycle has one vertical track for links and one for PEs. The width of the layout is thus $2^n(1+m)w\lambda$. There are $3 \times 2^{n-1} - 1$ horizontal tracks and each track has the width of a PE. Hence, the height of the layout is $m(3 \times 2^{n-1} - 1)w\lambda$. Therefore, the area is
$A_{FCCC-1} = m(3 \times 2^{n-1} - 1)2^n(1 + m)(w\lambda)^2$.
This is of $O((\frac{N}{\log N})^2)$, where $N = (n + 1)2^n$.
Thus, the layout area is optimal with respect to the measure of complexity in the VLSI grid model of [7].

We propose one more layout (layout-2) for the $FCCC$ network, which can improve the yield, and has the following construction:
Each column in layout-2 will have two cycles, as shown in Figure 3. Two columns together form a pair, and and two pairs form a cluster, such that the layout will have 2^{n-3} clusters. The two cycles in a column are connected by their complementary cube link, and the cross cycles in a pair are connected by cube links in the 0-th dimension. Two pairs are linked by cube links of dimension $n-1$ to form a cluster. The first and second clusters are linked by cube links of dimension $n - 2$, first and third clusters are linked by cube links of dimension $n - 3$, and so on. Likewise all the clusters are placed in the layout. The even numbered cycles are placed in the upper half, whereas the odd numbered ones are placed in the lower half, for each column.

Area: The width of this layout is $(2 + m)2^{n-1}w\lambda$, and the height is $m(3 \times 2^{n-1} - 2)w\lambda$. Therefore, the area is
$A_{FCCC-2} = m(2 + m)(3 \times 2^{n-1} - 2)2^{n-1}(w\lambda)^2$.
This layout is not only optimal but also offers some constant factor improvement in area. For example, for $m = 50$ and $n = 5$, $\frac{A_{FCCC-1}}{A_{FCCC-2}} = 2.00$.

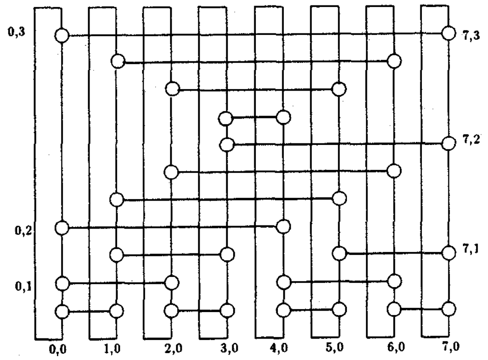

Figure 2: Layout-1 for $FCCC_3$.

Reliability: Though the number of links in both layouts remain the same, the average length of links will be less in layout-2. So the reliability of layout-2 would be higher than that of layout-1.

Yield: The random variable X is used to designate the number of faults per chip. The probability for k faults or failures occurring on a chip will be designated by $P(X = k)$. Then the probability of having k faults in the layout known [5] to be
$P(X \leq k) = \frac{\Gamma(\alpha+k)}{k\,!\,\Gamma(\alpha)} \frac{(AD/\alpha)^k}{(1+AD/\alpha)^{\alpha+k}}$,
and the yield is
$Y = (1 + A\,D/\alpha)^{-\alpha}$,
where α is the clustering parameter for the particular manufacturing process,Γ is the Gamma distribution, A is the area, and D is the defect density. Observe that the yield is higher for layout-2, as it occupies lesser area.

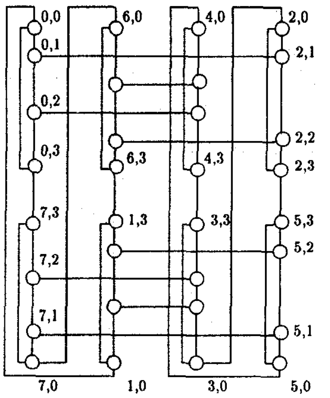

Figure 3: Layout-2 for $FCCC_3$.

3 Reconfigurable $FCCC$ layouts

In this section, we present two reconfigurable $FCCC$ network schemes. These are similar to the constructions suggested in [5] for reconfigurable cube-connected cycles. The first scheme utilizes local spare processors and switches, and employs local reconfiguration scheme (LR-FCCC), while the second scheme is based on global spare cycles (columns) and rows of processors and switches, and global reconfiguration (GR-FCCC). Both schemes use layout-1, and are capable of providing significant reliability improvement over the nonredundant $FCCC$ network.

3.1 Locally reconfigurable $FCCC$

In LR-FCCC, spare PEs and switches are included in each cycle of the $FCCC$ network. The number of spare PEs placed in each cycle is flexible, and is dependent on system constraints and reliability requirements. The faulty PEs can be bypassed by a network of simple switches; as these switches route data around faulty PEs. Reconfiguration is performed locally, with this strategy inside each cycle. Whenever a faulty PE is detected, a spare PE in the same cycle is switched in, to replace the faulty PE, and all the switches in that cycle are reprogrammed to reflect the changes. A link fault is treated as a PE fault, and is handled similarly. Figure 4 shows a layout of a locally reconfigurable $FCCC$ with $n = 3$ (the spare PEs are shown as square blocks).

A single spare PE per cycle may not provide sufficient reliability enhancement for large arrays, in which number of PEs in each cycle is large (i.e., for large n). So multiple PEs (G spares/cycle) may be included, by dividing each cycle into $\lceil\frac{n+1}{g}\rceil$ groups, with g PEs and one spare PE in each group. The spare PEs are linked together to form a ring of PEs. The degree of each regular PE is four, and that of a spare PE is $g+4$ (*or* $g+3$), since the spare PE is connected to every PE in the same group, to two other spare PEs in the same cycle, and to two(or one) spare PEs in neighbouring cycles. Now each group can tolerate one PE failure or one non-lateral link failure without performance degradation. Figure 4 shows a layout of a locally reconfigurable $FCCC$, with $n=3$, G=1, and $g=4$.

Area: The area of LR-FCCC layout is
$A_{LR-FCCC} = m(3 \times 2^{n-1} + \lceil\frac{n}{g}\rceil - 1)\ (\lceil\frac{g}{2}\rceil + t(G) + m)2^n(w\lambda)^2$. The value of $t(G)$ is 1 if G is 1. Otherwise, $t(G)$ is 3, because then we have more than one spare PE in a cycle and we need two extra tracks to connect these spare PEs into a ring. Hence the layout complexity of the LR-FCCC is of the same order as the optimal layout of the nonredundant $FCCC$ (i.e., layout-1). This layout also requires two layers only.

Reliability: The reliabilities for all PEs are assumed to be equal and exponentially distributed with failure rate λ_p, i.e., $R_p = e^{-\lambda_p t}$. For the LR-FCCC, each cycle has G spare PEs, one in each group. Hence the reliability of each group [5] is
$R_{group} = (R_p)^{g+1} + (g+1)(R_p)^g(1-R_p)$. The reliability of each cycle is $R_{cycle} = (R_{group})^G$, and the system reliability is $R_{LR-FCCC} = (R_{cycle})^{2^n}$.

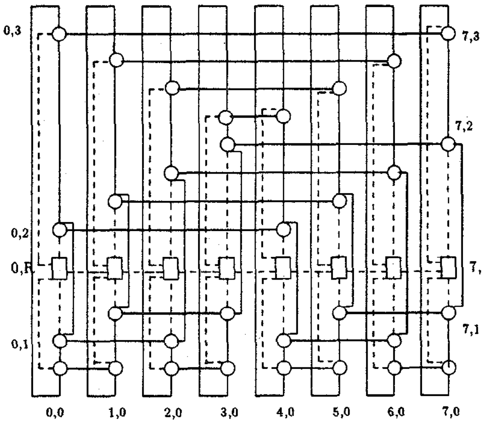

Figure 4: A layout of $LR-FCCC_3$ with one spare PE/cycle.

Yield: For the LR-FCCC, each group in a cycle can tolerate only one faulty PE. Hence the yield of a group [5] is
$Y_{group} = P(X \leq 1) = \sum_{k=0}^{1} \frac{\Gamma(\alpha+k)}{k\,!\,\Gamma(\alpha)} \frac{(A_{group}D/\alpha)^k}{(1+A_{group}D/\alpha)^{\alpha+k}}$, where A_{group} is the average area of a group. Thus the yield of a cycle is $Y_{cycle} = (Y_{group})^G$, and the yield of the LR-FCCC is $Y_{LR-FCCC} = (Y_{cycle})^{2^n}$.

3.2 Globally reconfigurable *FCCC*

The GR-FCCC is based on spare cycles of processors laid out as columns, and spare rows of processors. The $FCCC$ network is embedded in a switch-processor lattice. The switch lattice is a regular structure, formed from programmable switches connected by data paths. The PEs are not directly connected to each other, but are connected at regular intervals to the switch lattice. A configuration setting enables the switch to establish a direct, static connection between two or more of its incident data paths. The degree of each PE or switch is fixed at 4. The reconfiguration strategy is to replace a row or column containing one or more faulty PEs with a spare row or column, respectively. The allocation of spare rows/columns may be done as in [4]. Figure 5 shows an embedding of $FCCC_3$ in a switch-processor lattice.

Area: The area of the GR-FCCC is
$A_{GR-FCC} = [(m+1)(3 \times 2^{n-1} + r) - m](m+2)(2^n + c)(w\lambda)^2$. Here we assume the width of a switch in the GR-FCCC scheme is the same as that of a link. In general, the LR-FCCC scheme requires lower layout area compared to that of GR-FCCC.

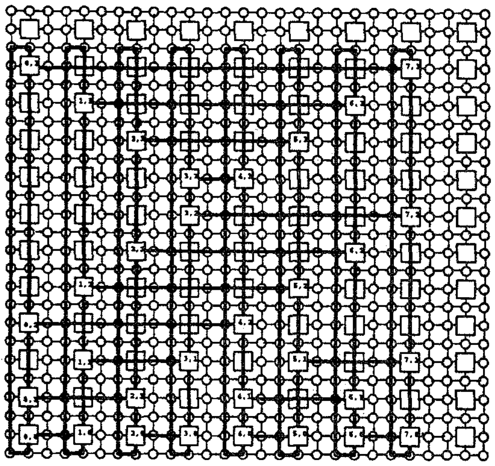

Figure 5: A $GR-FCCC_3$ with one spare row and column of processors.

Reliability: For the GR-FCCC with r spare rows and c spare columns, the system is working if no more than $r+c$ PEs are faulty, because one spare row or column will cover at least one faulty PE, although more faulty PEs can be tolerated if some faulty PEs lie in the same row or column. Therefore, the reliability of GR-FCCC, $R_{GR-FCCC}$, [5] is equal to
$\sum_{k=0}^{r+c} \left(\begin{array}{c} (2^n+c)(n+1+r) \\ k \end{array} \right) R_p^{(2^n+c)(n+1+r)-k}\ (1-R_p)^k$. Both LR-FCCC and GR-FCCC are much more reliable than the nonredundant FCCC. In general, LR-FCCC has higher reliability than GR-FCCC.

Yield: For the GR-FCCC with r spare rows and c spare columns, the circuit is repairable if the number

of defects is no more than $r + c$. Hence the yield, $Y_{GR-FCCC}$, [5] is at least
$= P(X \leq r+c) = \sum_{k=0}^{r+c} \frac{\Gamma(\alpha+k)}{k!\Gamma(\alpha)} \frac{(A_{GR-FCCC}D/\alpha)^k}{(1+A_{GR-FCCC}D/\alpha)^{\alpha+k}}$.
The actual value of $Y_{GR-FCCC}$ could be higher than the above, because a spare line can bypass more than one defect if they are on the same line. The redundant FCCCs give better yield than the nonredundant FCCC.

4 Building block approach

In this section, we propose layouts of universal building blocks [9], which could be used to build *FCCC* networks of any size and with/without redundancy. Each building block comprises one full cycle, or part of a cycle, and hence it requires at least 2^n blocks for building a nonredundant $FCCC_n$. The major advantage of this approach is the flexibility to construct *FCCC* network of any size. Another advantage is the facility to design fault-tolerant *FCCC*. The main drawback of the building block approach is that it increases the number of lengthy lateral links. Another drawback is the introduction of some critical components such as swithes into the structure, although the probability of failures of those critical components is relatively low.

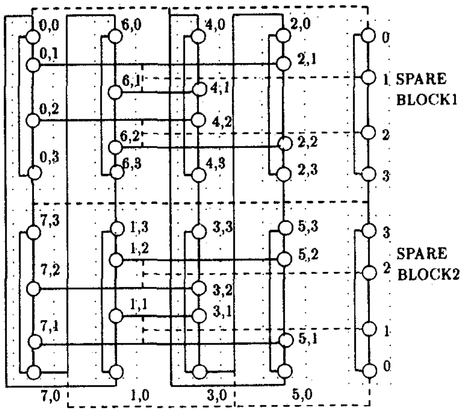

Figure 6: A globally reconfigurable building block $FCCC_3$.

There are three ways of incorporating redundancy using the building block approach. One may add redundant elements to each building block and use a local reconfiguration scheme. A second possibility is to avoid internal redundancy, but to add spare blocks, requiring a global reconfiguration scheme. This may require additional external switches and links to interconnect blocks. The third possibility is to incorporate both local and global redundancy. Here, the reconfiguration process consists of two steps: first, a local reconfiguration is performed in each block if necessary, and then, after executing a diagnosis, the spare blocks can replace malfunctioning blocks. We use layout-2 for the building block approach. For block level redundancy construction, though spare PEs, links, and switches are added to the building block, the degree of each PE remains four. Figure 6 shows a globally reconfigurable 3-dimensional building block *FCCC* with two spare blocks, constructed using blocks of size four with $4(2^3 + 2)$ PEs (no internal redundancy). If we add redundant PEs to each building block of Figure 6, then we get a locally and globally reconfigurable building block *FCCC*.

5 Conclusions

In this paper, we show that the *FCCC* network is realizable using VLSI/WSI. Two types of layouts were proposed, and evaluated for reliability and yield. The proposed reconfigurable *FCCC*s provide both reliability and yield improvement, and also fault tolerance without performance degradation. The universal building block approach allows the construction of fault tolerant *FCCC* networks of various dimensions.

References

[1] P. Banerjee, "The cubical ring connected cycles: a fault-tolerant parallel computation network," *IEEE Trans. on Computers*, Vol. 37, No. 5, pp.632-636, May 1988.

[2] A. El-Amawy and S. Latifi, " Properties and performance of folded hypercubes," *IEEE Trans. on Parallel and Distributed Systems*, Vol.2, No.1, pp.31-41, Jan.1991.

[3] S. Horiguchi and S. Fukuda, "Yield enhancement architecture of WSI cube-connected. cycle," *Proc. IEEE Int'l Conf. on Wafer Scale Integration*, pp.61-68, 1994.

[4] S. Y. Kuo and W. K. Fuchs, "Efficient spare allocation for reconfigurable arrays," *IEEE Design and Test* 4, pp. 24-31, Feb. 1987.

[5] S. Y. Kuo and W. K. Fuchs, "Reconfigurable cube-connected cycles architectures," *Journal of Parallel and Distributed computing*, 9, 1-10, 1990.

[6] F. J. Meyer and D. K. Pradhan, "Flip trees: fault-tolerant graphs with wide containers," *IEEE Trans. on Computers*, Vol.37, No.4, pp.472-478, April 1988.

[7] F. P. Preparata and J. Vuillemin, "The cube-connected cycles: a versatile network for parallel computation," *Communication of the ACM*, pp. 300-309, May 1981.

[8] M. P. Sebastian, P. S. N. Rao, and L. Jenkins, "Properties and performance of folded cube-connected cycles," Communicated to *Microprocessing and Microprogramming*, 1995.

[9] J. Shen and I. Koren, "Yield enhancement designs for WSI cube-connected cycles," *Proc. IEEE Int'l Conf. on Wafer Scale Integration*, pp. 289-298, 1989.

[10] N. F. Tzeng, "A reconfigurable cube-connected cycles architecture for wafer scale integration," *Proc. IEEE Int'l Conf. on Wafer Scale Integration*, pp.33-39, 1991.