1980

A Combinatorial Problem Concerning Processor Interconnection Networks

Michael J. O'Donnell

Carl H. Smith

Report Number:
80-352

A Combinatorial Problem Concerning Processor Interconnection Networks

Michael J. O'Donnell[1]

Carl H. Smith[2]

Department of Computer Sciences

Purdue University

West Lafayette, Indiana 47907

CSD TR 352

In highly parallel machine architectures, an important issue is how to interconnect components so as to pass data between processors. A crossbar switch connecting all the processors is very flexible, allowing any interprocessor connection. Crossbars however have cost proportional to the square of the number of processors they connect. As a consequence of the advances in microcircuitry, systems with thousands of processors are now feasible. Several less costly, and less flexible, processor interconnection networks have been proposed [3]. Of interest are the tradeoffs between the cost and the flexibility of each of the various interconnection schemes. The flexibility of an interconnection system is measured by counting the number of input to output permutations realizable by the network. In this note we count precisely the number of possible connection permutations achieved by the last stage of the Augmented Data Manipulator (ADM) introduced in [4] as a modification of the data manipulator network of [1].

In the final stage of the ADM network, a linear sequence of n components is

1. Supported by NSF grant MCS 7801812.
2. Supported by NSF grant MCS 7903912.

connected in such a way that adjacent positions may be interchanged. The contents of the first component may also be interchanged with the last. Every set of interchanges is possible, as long as a single component does not participate in two different interchanges (else the same datum would go to two places, and the result would not be a permutation). In addition, left and right circular shifts by one position are allowed. Below we calculate precisely $P(n)$, the number of different permutations which an ADM network of length $n \geq 1$ may achieve in a single step.

With each permutation performed by an ADM network of length n there is an associated bit string of length n. Within the bit string, the i^{th} bit is one if and only if the i^{th} and the $i+1^{st}$ inputs are interchanged by the associated permutation (see Figure 1).

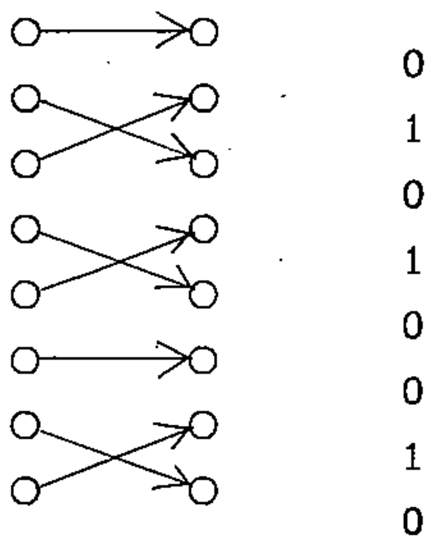

Figure 1. A permutation with n=8 and associated bit string

Thus, each setting of the network, except the two circular shifts, is associated with a string of 0's and 1's with the restriction that adjacent bits may not both be 1. Let $A(n)$ = the number of different strings of n bits in which no two adjacent bits are both 1, and the first and last bits are not both 1. Except for the trivial cases when $n=1$ or 2, $P(n)=A(n)+2$.

PROPOSITION 1

$P(1) = 1$

$P(2) = 2$

$P(n) = A(n) + 2$ for $n \geq 3$

Proof:

$P(1) = 1$ is obvious. $P(2) = 2$ since the only two possibilities are the identity permutation and the exchange. For $n \geq 3$, every different setting of the ADM network of size n produces a different permutation. Consider two different settings, s_1 which interchanges i and $i+1$, s_2 which does not interchange i and $i+1$. s_1 maps i to $i+1$, s_2 maps i to either i or $i-1$. For $n \geq 3$, $i, i+1, i-1$ are all different mod n, so the two permutations are different. By the discussion above, there is a one-to-one correspondence between network settings, excepting the two circular shifts, and the bit strings counted in the definition of $A(n)$.

The function $A(n)$ is easier to analyze using a similar function $B(n)$, which eliminates the restriction that the first and last bits in a string must not both be 1, i.e. $B(n)$ = the number of different strings of n bits in which no two adjacent bits are both 1.

PROPOSITION 2

$A(1) = 1$

$A(2) = 3$

$A(3) = 4$

$A(n) = B(n-1) + B(n-3)$ for $n \geq 4$

Proof:

The first three cases are verified by counting. For $n \geq 4$, consider an arbitrary string of n bits with no two adjacent 1's, including end around adjacency. If the first bit is 0, then the remaining $n-1$ bits may be set in any fashion as long as no two adjacent bits are 1. There are $B(n-1)$ such strings. If the first bit is 1, then the second and last bits must both be 0, but the remaining $n-3$ may be set in any fashion as long as no two adjacent bits are 1. There are $B(n-3)$ such strings.

PROPOSITION 3

$B(n)$ is a shifted Fibonacci series.

$B(1) = 2$

$B(2) = 3$

$B(n) = B(n-1) + B(n-2)$ for $n \geq 3$

Proof:

For $n \geq 3$, consider an arbitrary string of n bits with no two adjacent 1's. If the first bit is 0, then the remaining $n-1$ bits may be set in any fashion as long as no two adjacent bits are 1. There are $B(n-1)$ such strings. If the first bit is 1, then the second bit must be 0, but the remaining $n-2$ bits may be set in any fashion as long as no two adjacent bits are 1. There are $B(n-2)$ such strings.

Using equation (15) in [2 §1.2.8] and the above Propositions, we have that for $n \geq 6$

$$P(n)=[\frac{\varphi^{n+1}}{\sqrt{5}}]+[\frac{\varphi^{n-1}}{\sqrt{5}}]+2$$

where φ is the golden ratio $\frac{1}{2}(1+\sqrt{5})$ and $[x]$ is x rounded to the nearest integer. In [5] it is shown how to count the total number of permutations produced by an ADM in terms of $P(n)$.

We wish to acknowledge H. J. Siegel and G. Adams for bringing the above problem to our attention. Computer time was supplied by the Department of Computer Sciences at Purdue University.

References

1. FENG, T., "Data manipulating functions in parallel processors and their implementations," *IEEE Transactions on Computers* **C-23**(3) pp. 309-318 (1974).
2. KNUTH, D., *The Art of Computer Programming: Fundamental Algorithms*, Addison-Wesley , Reading, Mass. (1968).
3. SIEGEL, H. J., "Interconnection networks for SIMD machines," *Computer* **12**(6) pp. 57-65 (1979).
4. SIEGEL, H. J. AND SMITH, S. D., "Study of multistage SIMD interconnection networks," *Proceedings of the Fifth Annual Symposium on Computer Architecture*, pp. 223-229 (1978).
5. SMITH, S. D., SIEGEL, H. J., MCMILLEN, H. J., AND ADAMS, G. B., "Use of augmented data manipulator multistage networks for SIMD machines," *Proceedings of the 1980 International Conference on Parallel Processing*, (1980).