LABORATORY FOR
COMPUTER SCIENCE

MASSACHUSETTS
INSTITUTE OF
TECHNOLOGY

Analysis of Structures for Packet Sorting Networks

Computation Structures Group Memo 163
July 1978

Robert G. Jacobsen

Thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science at M.I.T.

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

Analysis of Structures for Packet Sorting Networks

by

Robert G. Jacobsen

This thesis examines the underlying theoretical structure of regular, homogeneous packet sorting networks. It addresses itself to the determination of the optimal architecture for a given application of packet sorting techniques.

Results relating the size, speed, cost and structure of optimal networks are obtained for a particular cost function and shown consistent with prior work. Issues of modularity are briefly examined.

Supervisor: David P. Misunas Title: Staff Member, Division of Sponsored Research

Table of Contents

Tables and Figures

Chapter 1 - Introduction

There are many valid ways to partition communication networks. Number of inputs and/or outputs, speed, geographic size and purpose have all been used as characterizations at some point. However, for the purposes of network design, it is often more useful to consider a network in terms of the type of connection made and the number of possible connections.

Each independent connection to a communication network can be either bidirectional or unidirectional. In the case of unidirectional connecting networks, it is generally possible to divide the devices connected to the network into "inputs" and "outputs", where a device either receives data or transmits data at a port, but not both. The basic telephone network is a clear example of bidirectional connections, while FM radio broadcasting is an interesting example of a unidirectional network.

So far, all the described networks have clearly defined physical paths for the transfer of information. Each such path is only used for one communication link. A different organization, similar to the postal service, is also possible. In this

scheme "packets" of information are interchanged, with the actual physical equipment that routes and delivers them shared among a number of packets. This sharing may be in one or more of time, space, frequency or a number of other alternatives. More esoteric examples include the timeslice switching in some modern phone offices and microwave transmission of hundreds of conversations at a time.

With the rapid sophistication in computer technology, many different computer networks have been constructed. Typically, these networks are used for communication between large processors. Networks such as the ARPAnet and the TYMSHARE network both involve relatively few processors and large messages. A significant amount of research has been done on the design and implementation of these structures. In general, the intelligence in the network itself has been distributed throughout, although such interconnection structures as Banyon networks[5] exhibit a central control structure.

At the other end of the spectrum, the data-flow computers under study at the MIT Laboratory for Computer Science and elsewhere contain larger, more distributed networks processing large numbers of relatively small packets [1,2,5]. In a data-flow

computer operations are carried out as data required for them becomes available, resulting in significant increases in parallelism.

The data-flow computer architecture under development at MIT implements data-flow computation through a structure which consists of large routing networks to transfer data between instructions of a program. The operator codes and operands forming an instruction are temporarily accumulated in a memory cell and, when all required data is available, are physically transfered to a functional unit which performs the appropriate calculation (Figure 1). The operation packets which transfer this information are handled by large, uniform networks. Previous work [4,5] has indicated the network performance which can be expected under certain conditions, given a description of the networks structure.

It is desirable to have a set of engineering principles to guide the design of this type of large, seemingly regular packet communication network. In the end an engineer would like a set of tables to consult, where given several known characteristics about the processor and the networks to be made a part of it, he could then immediately determine the best structure. What is known at the start of this design process?

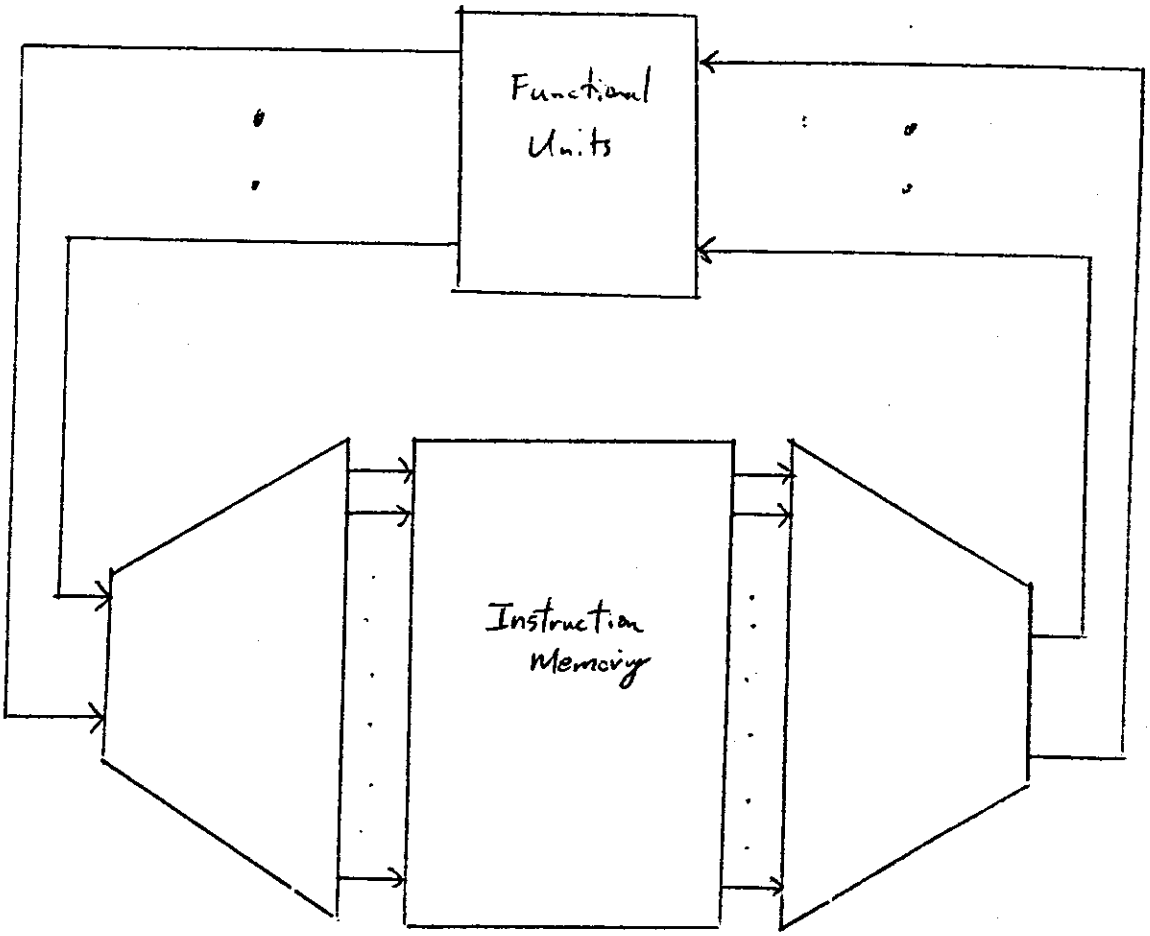


Figure 1 - A Data Flow Processor

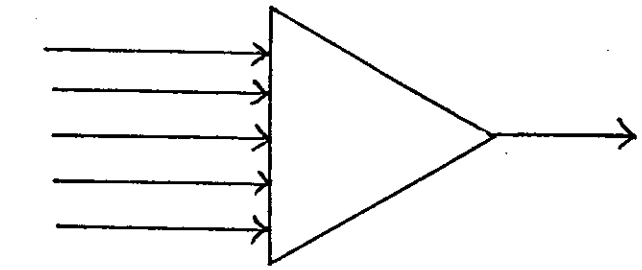
Typically, the type of calculation to be done and the desired rate at which it should take place.

It is beyond the scope of this thesis to discuss the transformation from a program and a speed requirement to an overall design of a data-flow processor. Assuming the engineer has managed to derive the number of memory cells required, the number of functional units and the delay time through the networks, he can now characterize the networks by their number of inputs N, outputs M, desired delay time D and total cost. The next chapter examines the tools currently available to the engineer, presents some of the prior art and discusses some of the assumptions needed for the design.

Chapter 2 - Structural Considerations

In the general case, a network designer knows the number of inputs and outputs (N and M) for the network and an average delay D. The goal, of course, is to start with these various network parameters and from them derive in a mechanistic way the most cost effective practical network design. This involves consideration of both theoretical and technological limitations. In general, there are more problems with the boundaries of current technology than with theoretical considerations.

The particular networks of interest are modeled by basic units called arbiters and switches as shown in Figure 2. An arbiter accepts a multitude of completely asynchronous packets and passes them on to it's output in some rough time order of arrival. A switch unit accepts individual packets at it's input and, based on some internal status of each packet, delivers it at one of the unit's output ports. These modules can be combined in various ways to create packet sorting networks. For example, larger arbiters and switches can be trivially built of smaller units (Figure 3). The bus structure found in many current computers can



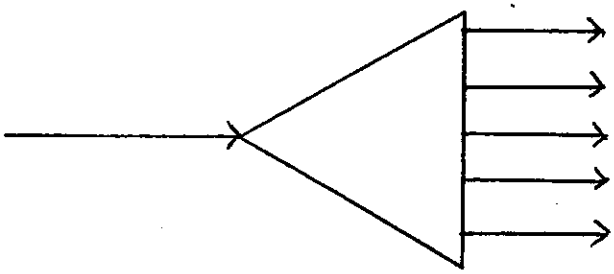


Figure 2 - Arbiter and Switch Units

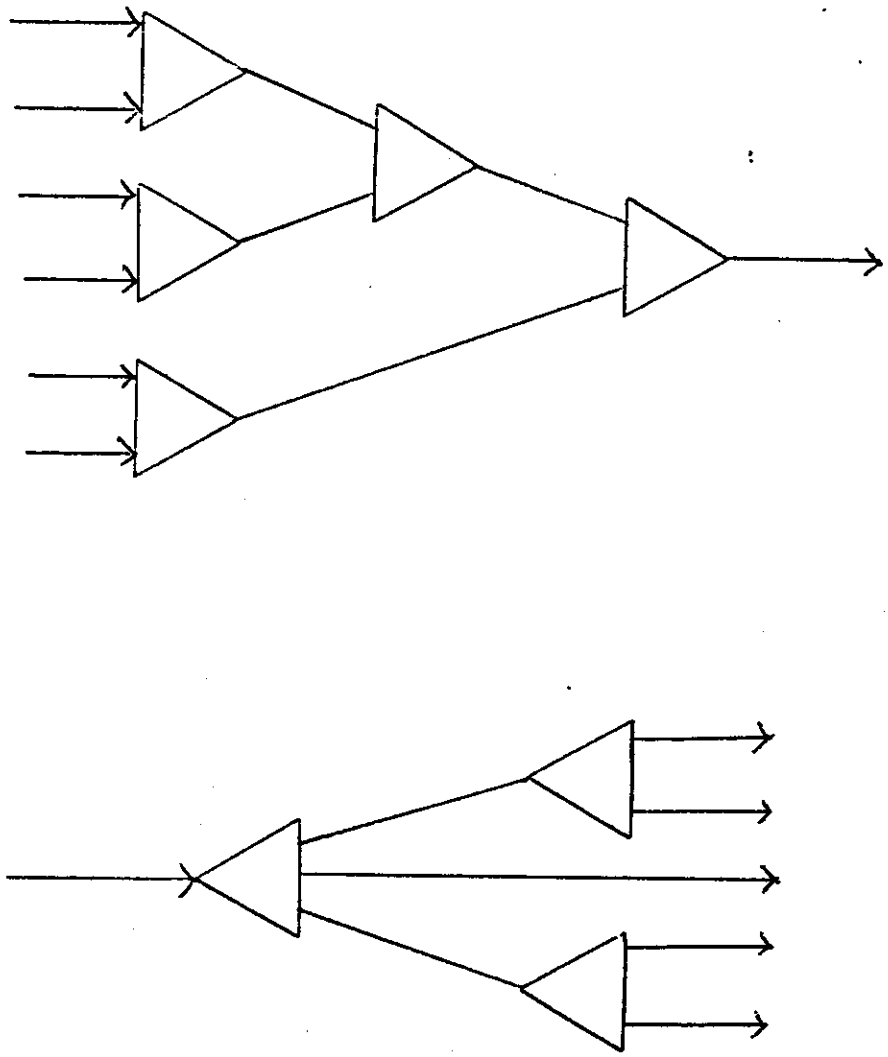


Figure 3 - Combinations of Units

be modeled by a single arbiter and switch (Figure 4). This representation clearly indicates the separation of the bus into two functionally distinct parts - the logic to allocate the communication path and the logic to determine the destination of the communication being done.

On a more complex level, the "crossbar switch" commonly used in telephone switching can be modeled by a larger number of arbiters and switches (Figure 5). This model clearly shows the inherent parallelism of the crossbar geometry and also it's rapid increase in complexity with number of inputs and outputs. Prior work has suggested that, in fact, the cost of the crossbar and the bus structure both grow at the same rate, which is approximately the number of inputs squared. Intuitively, the size of the bus grows as N, the number of inputs, but the speed (and thus roughly the cost) of it's internal structure also grows proportional to N. Conversely, the internal structure of a crossbar always runs at the same speed but it's complexity is of the order of N squared.

For larger networks, multilayer structures have been suggested [5]. By combining alternating layers of arbiters and switches, it is thought that a more cost effective structure can be obtained (Figure 6).

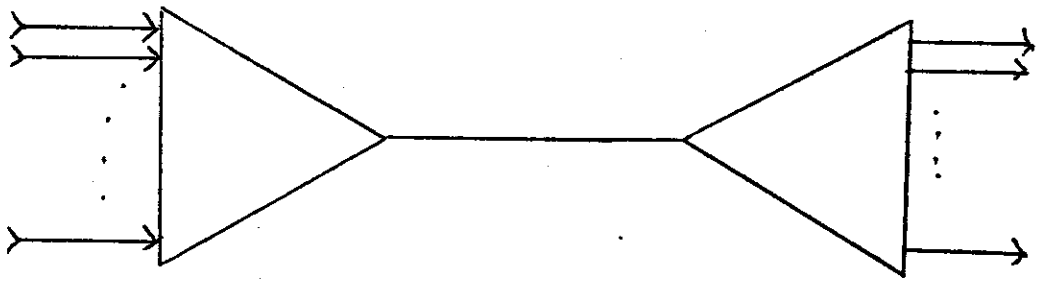


Figure 4 - A Model Bus

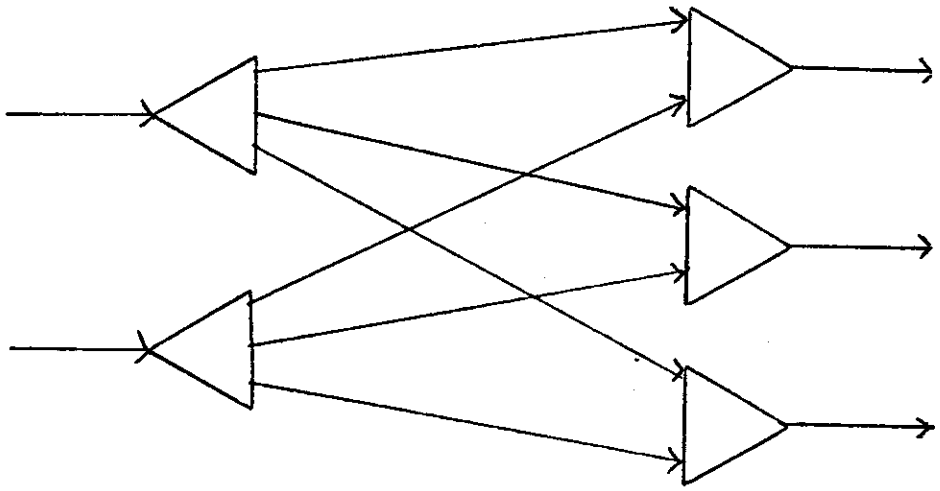


Figure 5 - A Model Crossbar

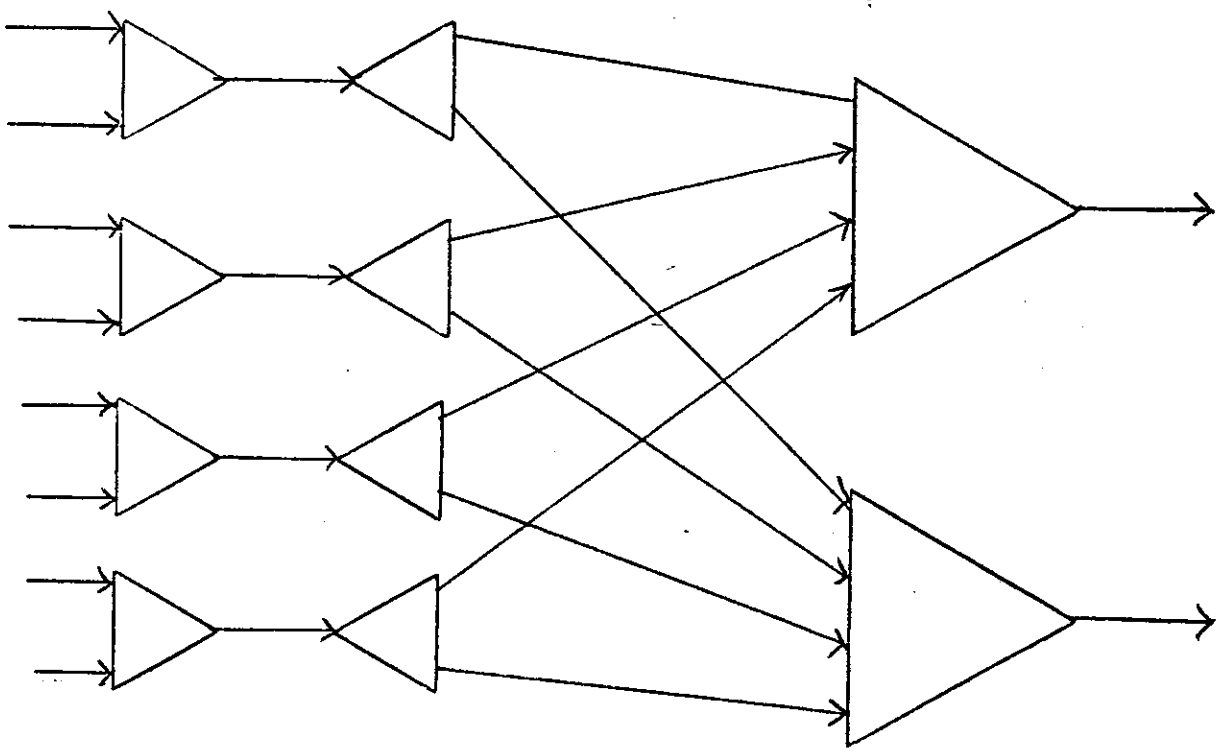


Figure 6 - A Simple Network

Analysis of this idea for the case of regular networks with equal numbers of inputs and outputs [4] indicates that it is indeed possible to make significant gains in this manner. In fact, the number of layers was found to be proportional to the logarithm of the number of inputs and outputs.

For the purposes of the current discussion, rectangular, regular networks will be considered. These networks are packet switched and have M interchangable outputs and N interchangable inputs. Two inputs or outputs will be considered interchangable if their traffic characteristics are identical and there are no dependencies between them - i. e., arrival at one is probabilistically independant of arrival at the other. Together, these limitations are clearly an abstraction from reality, where all the inputs and outputs are actually (complex?) functions of each other, but for large networks in equipment doing complex computation they are a satisfactory approximation.

A second, little discussed approximation involves the relative traffic on each external and internal link in the network. Jacobsen and Misunas [6] point out that for traffic load less than some value arbiters and switches may be viewed as small, single server queues. Above some other level, the analysis of Misunas [5]

predicts delays and throughputs based on bandwidth and "backup" considerations. The combination of these in the midranges of loading is not well understood. In any case, we shall temporarily assume that the loading of the network under consideration obeys the simple linearity constraint observed in both of the above models, namely that system delay is monotonically and continously increasing with increased loading. This assumption, which intuitively seems defensible, underlies much of the arguments to follow concerning efficient allocation of resources in network design.

Some basic geometric arguments can be simply made. First, there must be a high degree of path symmetry within the network, i. e. the path taken by any packet must be topologically equivalent to the path taken by any other. This is trivially true from the exchangibility of the various inputs and outputs. If there did exist an asymmetry between two paths, either one or the other would occasionally be advantageous to one packet over the other. This would violate the previous (and rigid) assumption of complete path symmetry within the network. Note that this disallows cases were one path in a network can be improved without degrading another. In general this would be a desirable change to make, but for now it will not be

considered.

Using the above, it is clear that since each path must go through several arbiters and/or switches, these arbiters and switches must be organized into levels that extend through cross-sections of the network. For example, each input may be sent into an input of a given arbiter, then through a switch, etc. (see Figure 7). Each of these levels must be made of identical units and the connections must be in some sense regular. This regularity of structure leads to a simple taxonomy of systems as described in Jacobsen and Misunas [4].

For the purposes of this discussion, sequential levels of arbiters are compressed to one of the appropriate larger size, and similarly for layers of switches. Although in practice it may be advantageous to construct large arbiters or switches of similar smaller units properly connected, this internal fine structure does not affect the theory presented here. The pipeline delay characteristics of the system are the only external variables effected and consideration of that is left for later.

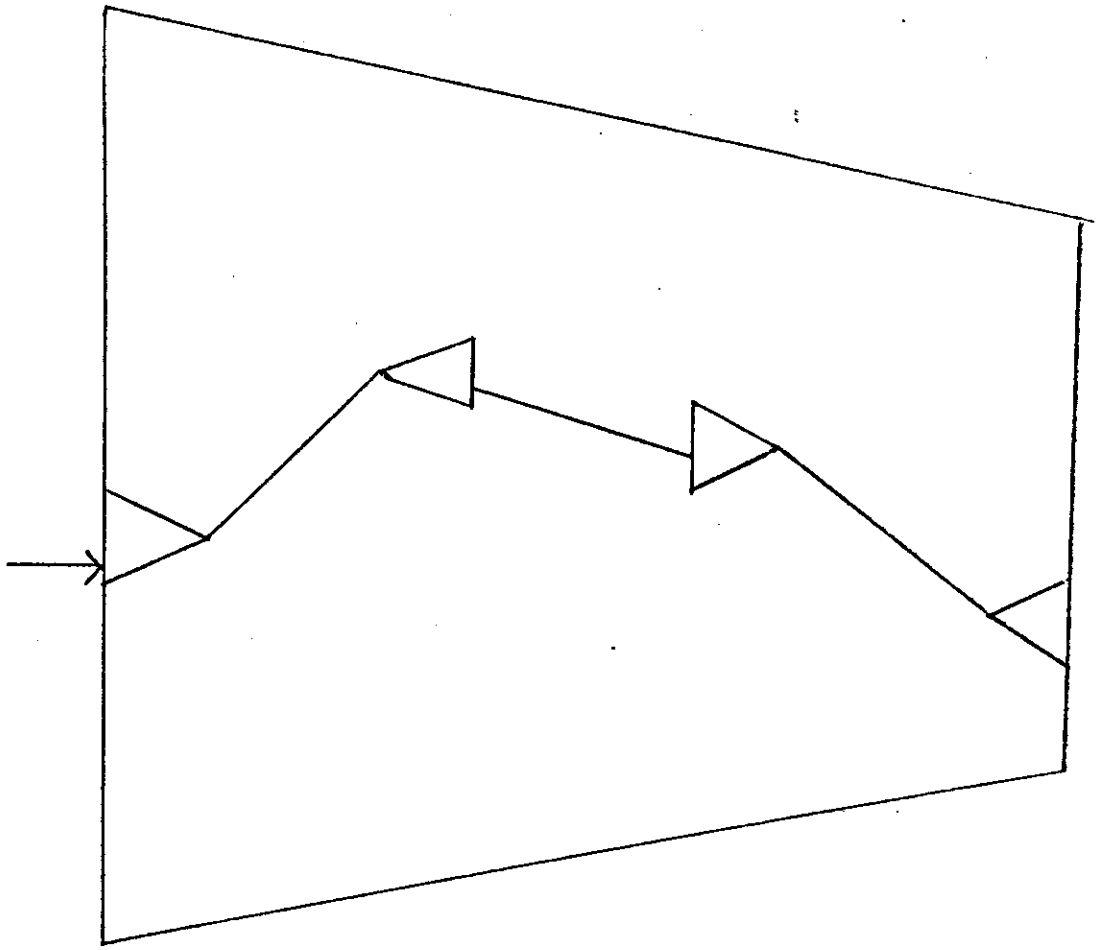


Figure 7 - A Typical Packet Routing

Chapter 3 - Analytical Results

The network modeled in Figure 8 has been isomorphically reduced to alternating layers of similar arbiters and switches. Each layer can be described by a compaction factor C, which is numerically equal to the number of inputs divided by the number of outputs. This compaction factor then applies to either a whole layer or the individual units within it. Similarly, the compaction factor for any unit in a given layer is identical to that of every other unit in that layer. The C for an arbitration layer is greater than one, while it is less than one for switch layers. In general, it is true that

$$\prod_{\forall i} C_i = \frac{N}{M} \qquad 3.1$$

where L is the number of layers in the network, N is the number of inputs to the network and M is the number of outputs. Since each input must have enough possible paths to traverse to reach any output, it is also

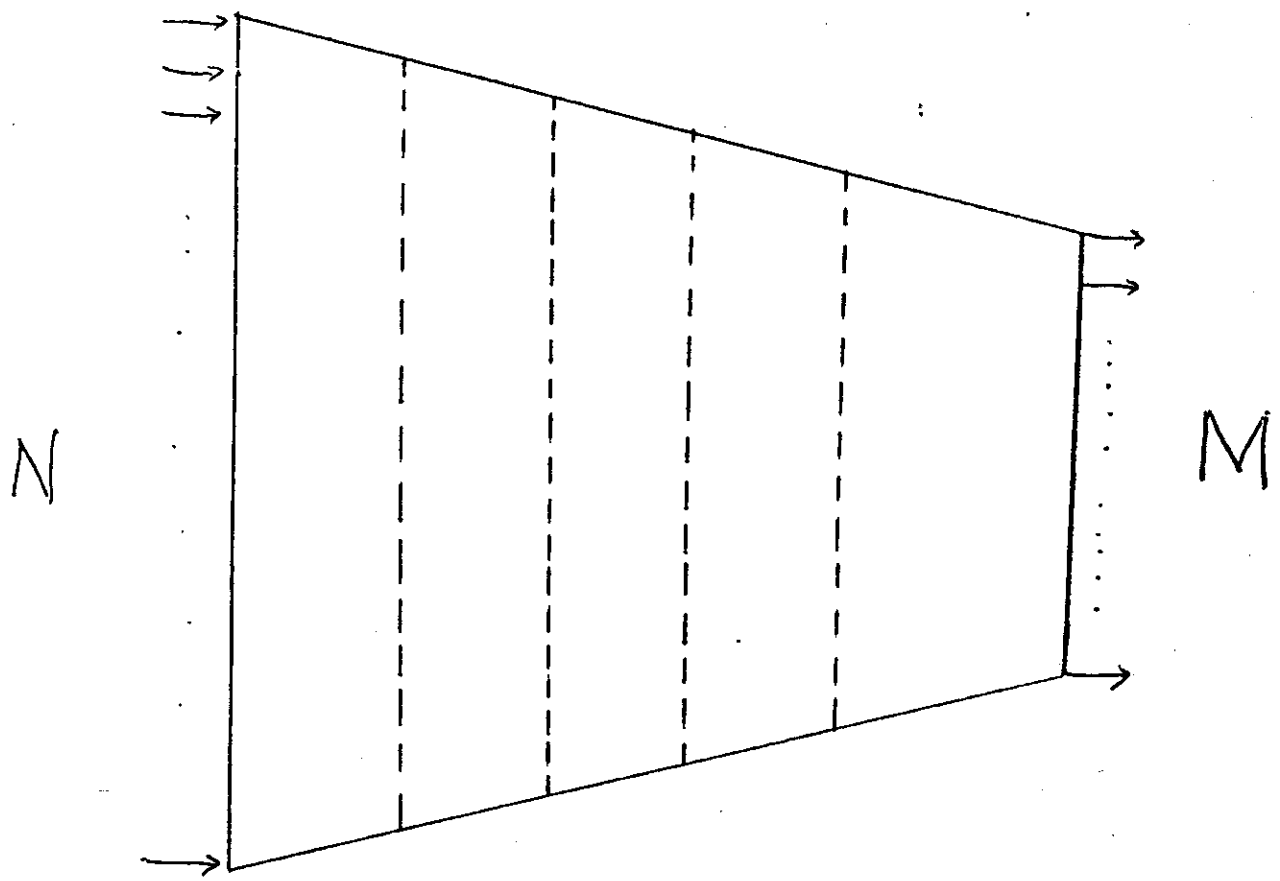


Figure 8 - Network Structure

true that

$$\prod_{\forall i \text{ s.t. } C_i > 1} C_i = N \qquad 3.2$$

Similarly, each output must have a path from each input, leading to the relation

$$\prod_{\forall i \text{ s.t. } C_i < 1} 1/C_i = M \qquad 3.3$$

These are minimums - a network exceeding the connectivity implied by either 3.2 or 3.3 is capable of handling redundant paths although proper operation or completeness is not guaranteed by the above criteria. Similarly, a rectangular NxM network meeting the above net connectivity criteria may still be connected incorrectly.

It is now possible to define a new unit, called a tie, which accepts n inputs and switches them to m outputs. (Figure 9). The tie can be modeled as a arbiter followed by a switch (bus type), by a number of switches feeding a number of arbiter(crossbar type) or by something internally more complex. The compaction factor of this tie is n divided by m. Proper selection of n and m allows a layer of ties to exactly replace a switch and arbiter layer. For example, choosing all

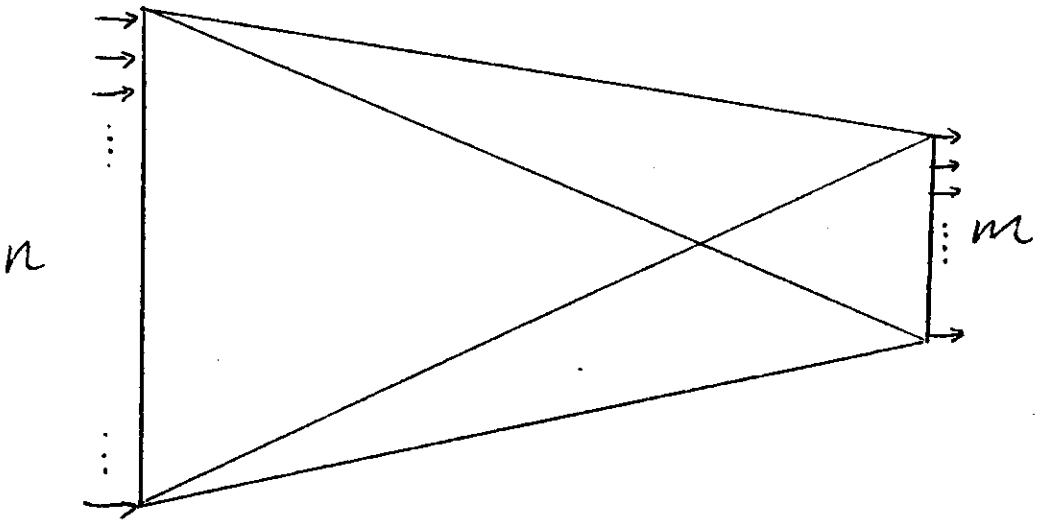


Figure 9 - A Tie Unit

rigidly assures the three criteria of 3.1,3.2 and 3.3.

At this point, it is necesary to relate the cost and value of one of these ties to real technology. As a compromise between strict accuracy and usefulness, the cost of a tie will be defined to be proportional to it's speed (1/delay) in arbitrary units times it's complexity times it's size. Size is easily determined - for a fixed compaction ratio, it is proportional to n. Strictly speaking, cost proportional to speed is a theoretically nice but clearly impractical approximation. The cost of modern electronics is very widely affected by speed, but for small variations, conversion from to serial to various widths of parallel follows a linear cost curve. In any case, more accurate cost functions of speed are presently not feasible. Similarly, complexity is a vague term at this point. Analogies from line switching theory suggest that the complexity should be approximatly the square root of nxm. This is also known as the geometric mean of n and m.

$$COST = n\sqrt{nm}\left(\text{relative speed}\right)$$

For the case of n=m, this reduces to the cost function assumed in the prior work [4]. For the sake of

simplicity, the absolute speed of the network will be left out of much of the following discussion. It should be considered normalized to unity, and following references will be to relative speed inside the network.

Several additional, weaker criteria are satified by this cost function. First, it is symmetrical front to back. If allowance is made for the change in line loading due to the differing number of inputs and outputs, the cost of a tie is the same from either the input or output side. This is not strictly intuitively needed, but seems to bear out common examples. Secondly, the partial derivatives of this cost function with respect to n and m give very satisfying marginal costs

$$m.\ cost\ of\ n = \frac{total\ cost}{n}$$

$$m.\ cost\ of\ m = \frac{total\ cost}{m}$$

The cost symmetry of a tie also leads to a cost symmetry among networks. If an NxM network of minimal cost for fixed performance costs D to build, it's inverse MxN network will cost exactly D for the same performance.

Prior work has suggested that the bandwidth (equal to the number of lines into a level times that level's speed) should be equal at every layer [5]. It is now possible to prove this rigorously. The full details are presented in the Appendix, but it is useful to sketch the proof here. It is desirable to maximize the overall speed of the network subject to a fixed cost constraint. Since the performance of a network is inversely proportional to the total packet delay through it, and the total delay is just the sum of each layer's delay, we wish to minimize

$$\sum_{i=1}^{L} D_i$$

where D is the delay at level i. Since the speed of a level is directly proportional to the cost of the level, this can be restated in proper units as

$$\min \sum_{i=1}^{L} D_i$$

subject to

$$\text{CONSTANT} = \sum_{j=1}^{L} 1/D_i$$

where L is the number of layers in the network, which is independantly fixed at an optimal level, and D is the delay in the ith level. Using the Langrangian, it can be shown that this is done when the resources are

spread to ensure equal bandwidth. Intuitively, this can be seen from the general equivalence of the layers - no layer is loaded by any more packets than any other, so there is no reason to have asymmetries in the topology of the network structure.

Similarly, a relation can be established among the compaction factor of the ties in the various levels. It has been previously shown that irrespective of the cost function of ties, each layer should be made of a single size. Using the cost function established above

$$COST = n\sqrt{n\,m}\,(speed)$$

it is possible to find a relation between the levels. Again, the best solution arises when the cost is minimized for a given performance. Leaving the speed component fixed, the network cost equals the sum of the costs of the layers. Each layer has a cost

$$COST = n_i\sqrt{n_i m_i}\,(number\ ties)\,R$$

where R is the relative speed of the layer. This term arises because as the network becomes more and more compact, each tie must run faster to maintain the same delay. This can be more precisely stated as

$$COST = N\sqrt{n_i m_i}$$

taking advantage of the fact that the bandwidths of all layers are equal. The minimum of this must satisfy the condition that the product of all the compaction factors equals the overall compaction factor of the network. Symbolized

$$\text{CONSTANT} = \prod_{1}^{L} \frac{n_i}{m_i}$$

Again using Langrangian techniques, this is found to be at a minimum when the compaction factors are all equal. Intuitively this is very satisfying, for the "compaction load" , which consists of the amount of delay inherent in the arbitration process, should be uniformly spread over the structure.

The only significant remaining variable to be considered is the number of layers constituting the network. Once this number is found, the designer can find the compaction factor and required speed of each layer and the number of ties required. Again the cost function must be used to find the minimum cost design for fixed performance, which is the most useful set of dependent and independent variables for engineering applications.

The cost of the entire network is equal to the sum

of the layer costs. Each layer consists of a given number of identical individual units. Therefore the cost of the network is given by

$$COST = L \sum_{1}^{L} N \sqrt{n_i m_i} R$$

where L is the number of layers in the network. It appears in the equation to compensate for the fact that as more layers are added to a network, each must run faster to maintain a constant delay. However, since the bandwidth of each layer was previously shown to be equal, this can be expressed more precisely as

$$COST = L^2 N \sqrt{n_i m_i}$$

where the terms correspond to those in the preceeding equation and the speed at the inputs to the network has been normalized to unity.

This can be expressed in terms of the overall size of the network since n and m are just the Lth roots of N and M. Therefore

$$cost = L^2 N^{1+1/L} M^{1/L}$$

This cost equation can be minimized with respect to L by differentiating and setting to 0, becoming

$$0 = \left(2L - \tfrac{1}{2}\left(\ln N + \ln M\right)\right) N^{1+\frac{1}{2}L} M^{1+\frac{1}{2}L}$$

Solving for L

$$0 = 2L - \frac{1}{2}\left(\ln N + \ln M\right)$$

$$L = \frac{1}{4}\left(\ln N + \ln M\right)$$

Some numeric examples of this are seen in Table 1. Note that with N=M this reduces to the previous work of Jacobsen and Misunas [4].

To gain a feel for the growth rate of total cost as N and M increase, L can be resubstituted to obtain

$$\text{cost} \propto \left(\ln N + \ln M\right)^2 N^{1+\frac{2}{\ln N}+\frac{2}{\ln M}} M^{1+\frac{2}{\ln N}+\frac{2}{\ln M}}$$

which can be reduced to approximately

$$\text{cost} \propto \left(\ln N + \ln M\right)^2 N$$

Also interesting is the size of the individual ties that make up a network. Although they are operating at different speeds (except in the case of N=M), they are all the same size, which seems to indicate that

N	M	Optimal Number of Layers
1	1	0
8	8	1
60	60	2
400	400	3
2500	2500	4
16000	16000	5
60	2500	3
8	16000	4

Table 1 - Some Numeric Results

properly designed ties may be simply stacked in parallel to form the network.

This analysis has in no way considered pipelining. The characteristics of the equipment connected to each end of the network will determine the need for and significance of pipelining in the network. In general, the addition of pipeline requirements should not affect the foregoing analysis extensively. The internal design of the ties, particularly those in the first and last layers, determines the pipeline delay of the network independent of the overall delay.

Chapter 4 - Conclusions

This thesis has examined the issues involved in the design of uniform routing networks. The theoretical optimum values developed for various network parameters are meant to form starting values for the practicing engineer in his optimization of a particular practical design. As such, they do not include consideration of many technological issues, such as the actual relationship between cost and speed for a given network architecture. The linear assumption used extensively in this work breaks down for large changes in speed, but it is not clear exactly what it's replacement should be.

Similarly, the foregoing analysis has not considered the devices that are communicating through the network. Their design is intimately related to the design of the network, since tradeoffs between the speed and cost of various components of the overall system may change it's performance radically. Eventually, analysis of the interaction between components of large systems, where a network would be viewed as a fundamental unit with various parameters, may be feasible.

At the other end of the spectrum, further consideration should be given to the design of the ties that make up the network. Along with their actual design, information on a possible basic set which can be easily combined to form networks of a large variety of sizes and speeds would greatly advance the art.

In summary, it seems that the modular design of large regular networks is starting to come within the realm of understood, commonly used hardware for computation.

Appendix - Derivation of Selected Results

The proof of the statement of chapter two concerning the equal bandwidth proof involves finding the minimum of

$$D_1 + D_2 + \ldots + D_L$$

given

$$\frac{1}{D_1} + \frac{1}{D_2} + \ldots + \frac{1}{D_L} = \text{CONSTANT}$$

To find this, the Lagrangian

$$0 = D_1 + D_2 + \ldots + D_L + \lambda\left(\frac{1}{D_1} + \frac{1}{D_2} + \cdots + \frac{1}{D_L} - C\right)$$

is formed and it's partial derivatives

$$\frac{\partial L}{\partial D_i} = 1 - \lambda\frac{1}{D_i^2} = 0 \qquad \frac{\partial L}{\partial \lambda} = \frac{1}{D_1} + \frac{1}{D_2} + \cdots + \frac{1}{D_L} - C = 0$$

are found. From this L+1 equations are generated and solved to obtain

$$\lambda = \frac{N^2}{C^2}$$

and thus

$$D_i = D_j \quad \forall i, j$$

which thus enforces the equality of bandwidths.

Similarly, the equal compaction factor proof involves finding the minimum of

$$\sum \sqrt{n_i m_i}$$

given that

$$\prod \left(\frac{n_i}{m_i}\right) - \frac{N}{M} = 0$$

Forming the Lagrangian again,

$$\sum \sqrt{n_i m_i} - \lambda\left(\prod \left(\frac{n_i}{m_i}\right) - \frac{N}{M}\right) = 0$$

Taking all partial derivatives and setting equal to zero,

$$\frac{\partial L}{\partial n_i} = 1/2 \frac{\sqrt{m_i}}{\sqrt{n_i}} + \frac{\lambda N}{n_i M} = 0$$

$$\frac{\partial L}{\partial m_i} = 1/2 \frac{\sqrt{n_i}}{\sqrt{m_i}} - \frac{\lambda N}{m_i M} = 0$$

It is now possible to generate 2*L+1 equations, of which L+1 are independent. Of the form

$$\frac{1}{2} \sqrt{\frac{n_i}{m_i}} + \lambda \frac{N}{M} = 0$$

these show that

$$\frac{n_i}{m_i} = \frac{n_j}{m_j} \quad \forall i, j$$

and therefore the compaction factors of each level must be equal.

Bibliography

1) Dennis, J. B. and D. P. Misunas, "A Computer Architecture for Highly Parallel Signal Processing", Proceedings of the ACM National Conference, ACM, New York, Nov 1974, pp. 402-409.

2) Dennis, J. B. and D. P. Misunas, " A Preliminary Architecture for a Basic Data-Flow Processor", Proceedings of the Second Annual Symposium on Computer Architecture, IEEE, New York, Jan. 1975, p 126-132.

3) Goke, L. R. Connecting Networks for Partitioning Polymorphic Systems, Doctoral Dissertation, University of Florida, 1977

4) Jacobsen, R. G. and D. P. Misunas "Analysis of Structures for Packet Communication", Proceedings of the 1977 International Conference on Parallel Processing, Aug 1977

5) Misunas, D. P. "Performance of an Elementary Data-Flow Processor", MIT Laboratory for Computer Science, Computation Structures Group Memo 115, Cambridge, Ma, Feb. 1976

Analysis of Structures for Packet Sorting Networks

by

Robert G. Jacobsen

Submitted in Partial Fulfillment

of the Requirements for the

Degree of Bachelor of Science

at the

Massachusetts Institute of Technology

May 1978

Signature of Author.....
Department of Electrical Engineering and
Computer Science, May 11,1978

Certified by.....
Thesis Supervisor

Accepted by.....
Chairman, Departmental Committee on Theses