ROYAL MELBOURNE INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMMUNICATION AND ELECTRONIC ENGINEERING

SIMULATION OF NUMERICAL PETRI NETS USING DATA-DRIVEN COMPUTER ARCHITECTURES

Report 3
Contract No. 63228

Dr G.K. Egan
Senior Lecturer, Digital Electronics and Computing Systems

1. Introduction

The original objectives of the program where to evaluate in 1982:

a) what data-driven computer architectures might be appropriate to the simulation of Numerical Petri Net models of tellecommunications protocols, services and systems; and

b) what improvement in efficiency could be expected compared with traditional simulation methods using sequential computer systems.

Subject to a significant order of improvement being evaluated in b., the program would proceed in 1983 to:

c) investigate how the economical design and implementation of a suitable architecture might be conducted;

d) investigate the effectiveness of the proposed architecture by computer simulation; and

e) produce estimates of the development costs of implementing a prototype machine.

Studies in the initial stages of the program indicated that the RMIT architecture, which was designed to directly evaluate functions expressed as directed graphs, could be readily adapted to the evaluate petri nets. The studies also showed however that the more complex firing conditions and memory reference requirements of numerical petri nets may result in substantial overheads in determining enabled transitions.

The man power required to write appropriate translators to convert NPNs to data-flow graphs was outside the level of project funding and prevented extensive studies of what increase in efficiency might be achieved using data-driven architectures. Inspection however indicated that the NPN examples provided exhibited little exploitable concurrency and the gains for these protocols at least would be less than a decimal order of magnitude reduction in computing time using a conventional sequential computer of similar technology. In 1982-83 changes to algorithms used within the NPN Analyser at Telecom resulted in a substantial reduction in the computing time required for the analysis of NPNs; this made the original simulation efficiency goals less pressing.

In consultation with Telecom in December 1983 it was agreed to shift the study from a simulation emphasis, to how timing analysis and tentatively direct execution of protocols might be performed using data-driven architectures. It was agreed that the original design and implementation aspects of the study were to continue as originally proposed.

2. Timing and Direct Execution of Numerical Petri Nets

Having verified that a protocol expressed as an NPN is correct the process of translation by what may be ad hoc. methods to executing code on processors in a network raises questions of

whether the function described by the NPN is accurately reflected by that code. The prospect of direct execution of NPNs by data-driven architectures suggests itself.

Additionally verification of a protocol using the current analysis tools gives little information as to how long a protocol takes to perform its task. It is possible that while the protocol is safe it may not prove practical in terms of speed.

2.1 Timing

The data-flow simulator modelling a 128 processing-element system has been extended to time tag all tokens generated during the simulation. The time tag which is computed as the age of the oldest operand token plus the node evaluation time. This results in the last token generated in a computation having an age equal to that of the last token generated in a perfect system. This age is compared with the real computation time for a graph or net indicates how effectively the available concurrency in the graph or net was exploited and how long a protocol takes to perform its function in the perfect case.

2.2 Direct Execution

Translation of NPNs to data-flow graphs has proved impractical and an alternative approach of direct execution of NPNs is now being explored. This entails the writing of NPN interpreter sections for the data-flow hardware simulator and eventually the pilot data-flow system hardware. The interpreters would accept NPN descriptions directly with minimal translation to internal system representation. Substantial progress has been made on this work with the additional man power available to the project. It is intended that the firing times of transitions be directly proportional to the computational effort required to evaluate firing conditions and fire the transition.

Some of the firing conditions for NPNs specifically those involving the use of memory reference variables wil reduce the efficiency with which NPNs may be executed. It is proposed that the matching block of the pilot system described in section 3.1 would determine which transitions are eligible to fire. The evaluation block would access reference variables local to that block modifying them as necessary. If the matching block requires access to these variables then requests for current values must be directed to the evaluation block with responses being passed back via the local queue. This is not to be recommended and it is suggested that control in the NPNs should be limited to explicit passing of tokens not by 'invisible' control through memory reference variables.

3. Data-flow Computing System

A pilot system of 32 elements is being constructed to support the Numerical Petri Net Study ond other studies. The pilot system is described briefly in the following sections.

3.1 Pilot System

A prototype processing-element constructed as a final year design project and based on two Motorola M68000 16/32 microprocessors has been commissioned and is executing data-flow graphs successfully. A production version of this processing-element will form the basis of a pilot system. The structure of the processing-element being used in the pilot system is shown in Figure 3.1. The matching section accepts tokens from the Input Queue or Local Queue and determines whether the firing conditions for a data-flow node or petri net transition(s) have been satisfied. The evaluation section evaluates the node function or generates the selected transition output tokens and sends them to either the Local or Output Queues.

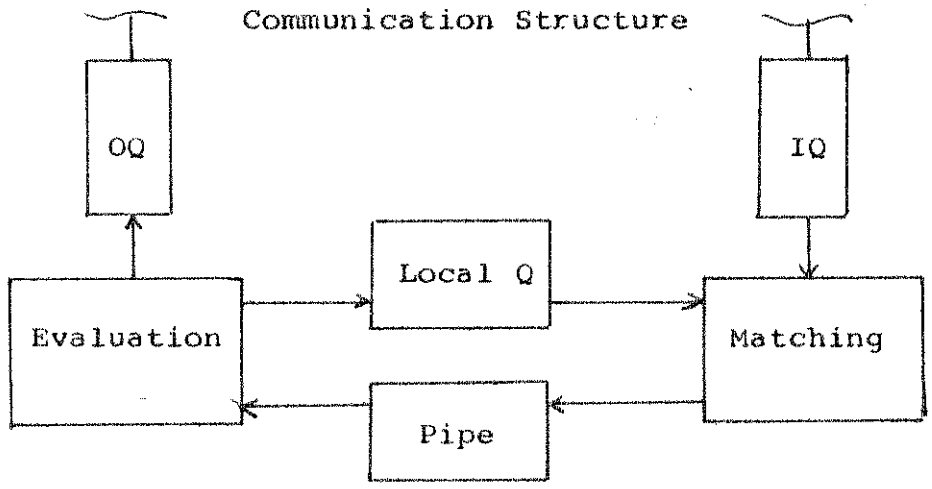

Figure 3.1 Processing-element

The 32 elements in the pilot system will be connected by a 'rectangular' shuffle network communication structure. Rectangular shuffle networks allow tokens or packets to be switched to any of n destination elements address in log2 n levels with destination address being independent of the source element address. Two-input router blocks are the basic components of the data-flow computer systems communication structure; the configurations of the router blocks are shown in figure 3.2. At each level of the network router blocks accept packets of data on their two input highways and direct them one of the two output highways. The packets are sequences of 16-bit words with a sub field of the first word specifying which of the router block's two output data highways the associated packet is to be directed to. The target time to transfer a 16-bit word through one stage in the pilot system is 100 nanoseconds. A network for 8 elements is shown in Figure 3.3.

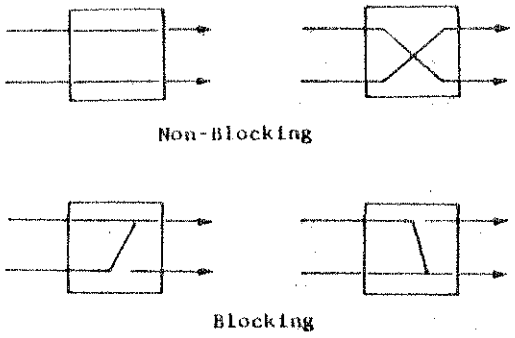

Figure 3.2 Router Block Configurations

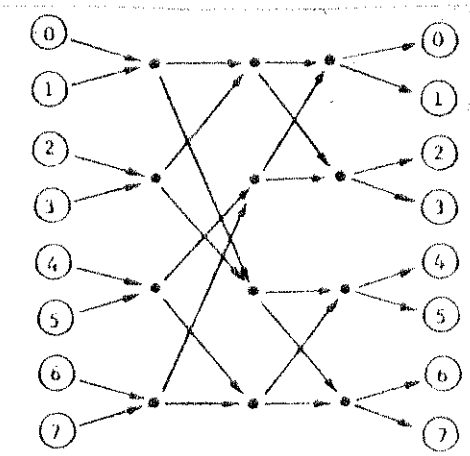

Figure 3.3 Rectangular Shuffle Network (8 elements)

The final mix of element types has not been finalised yet but the initial configuration will be eight processing-elements, four input-output elements and two fast mathematics elements. The mix of element types may be varied dependent on the application.

3.2 Sub-Projects for 1984

The Data-flow project team in 1984 is comprised of two academic staff members, two master of engineering and eight undergraduate students. The sub-projects being undertaken by student members of the project team are described briefly in the following sections.

3.2.1 Fast Processing Element Structures

Development of the processing-element to be used in the pilot system has suggested several areas for further research. Matching tokens on data-flow nodes or determining whether petri net transitions are enabled can be computationally time consuming;

this particularly so with numerical petri nets. The study of appropriate harwdware structures for this matching project is one aspect of the master of engineering program being undertaken by Mark Rawling.

3.2.2 Packet Communication Structures

Issues of reliability, size and speed of the Ø.5n log2 n (8Ø) router blocks which will be used to construct the communication structure of the 32 element pilot system are vital if problems associated with the communication structure are not to dominate system construction. Allan Young is examining these issues as part of his master of engineering program.

3.2.3 Communication System Two Input Router

Two students will assist Allan Young (3.2.2) in developing alternative router block implementations. The final year students responsible for this project are Chris Baharis and Stephen Dalloste.

3.2.4 Input/Output Processing-element

The existing processing elements for the data-flow multiprocessor are optimised for computation. This project will develop a general purpose input/output processing element. As with the current computational processing-element the input/output processing-element will be based on the Motorola 68ØØØ 16/32 bit microprocessor. The final year student responsible for this project is Devinder Singh.

3.2.5 Graphical Data-Flow Compiler

A graphical compiler was written in 1983. The project this year is to complete the development of this compiler to allow full top down design of data-flow programs or graphs. The compiler input language is then to be re-configured to accept at least one notation common in signal processing and control e.g. Petri Net Diagrams. The third year student undertaking this project is Steven Mogg.

3.2.6 Fast Mathematics Processing-element

A processing-element optimised for polynomial evaluation is being constructed to enhance the computational capability of the data-flow system. The mathematics processing-element will accept evaluation tokens comprising destination address, function code and operand(s) transmitted to it by processing-elements with limited or no mathematics capability. The mathematics processing-element will perform arithmetic, trigonometric and hyperbolic functions with a target time for floating point multiplication of 2 microseconds. The final year students responsible for this project are K.K. Lim and S.K. Chan.

3.2.7 Optical Communication System

Optical beams will be used to implement a high bandwidth point to point communication system. This project is speculative and if successful will solve many problems associated with data-flow systems with large (>1000) numbers of processing elements. The final year students responsible for this project are John Gallant and Mike Walkington.

References and Bibliography

Bearman, M., et al., 'Experience with Formal Specification and Analysis of the OSI Class Ø Transport Protocol', Draft Document, Switching and Signalling Branch, Telecom Australia, Dec. 1983.

Billington, J. et al., 'Modelling and Analysis of Communication Protocols Part 1 and 2', Proceedings of IREECON'81 International, Melbourne, 1981.

Cain, G.J. et al., 'Computer-Aided Chill Code Generation', Report 1Ø, Telecom Contract 53Ø1, March 1983.

Campos, I.M. and Estrin, G., 'SARA aided design of software for concurrent systems', AFIPS Conference Proceedings, Vol.47, -.

Dennis, J.B., 'A Preliminary Architecture for a Basic Data-Flow Processor', Computation Structures Group Memo 1Ø2, Massachusetts Institute of Technology, August 1974.

Egan, G.K., 'Data-flow: Its Application to Decentralised Control', Ph.D. Thesis, Department of Computer Science, Victoria University of Manchester, 1979.

Egan, G.K., 'FLO: A Decentralised Data-flow System Part 2', Internal document, Deptartment of Computer Science, Victoria University of Manchester, Jan. 198Ø.

Egan, G.K., 'A Decentralised Computing System Based on Data-Flow', Proceedings of the IECI'8Ø Conference, March 198Ø.

Egan, G.K., 'A Data-Flow Computing System for Decentralised Control and Advanced Automata Applications', Proceedings of IREECON'81 International, AUG. 1981.

Egan, G.K. and Richardson, C.P., 'Object Recognition Using a Data-Flow Computing System', Microprocessing and Microprogramming 7, North-Holland, 1981.

Egan, G.K. and Richardson, C.P., 'Manipulator Control Using a Data-driven Multiprocessor Computer System', National Conference and Exhibition on Robotics, Melbourne, August 1984.

Egan, G.K., 'Simulation of Numerical Petri Nets Using Data-driven Computer Architectures', Report 1&2, Telecom Australia, Contract No.63228, 1983.

Estrin, G., 'A Methodology for design of digital systems-Supported by SARA at the age of one', AFIPS Conference Proceedings, Vol.47, -.

Genrich, H.J. and Lautenbach, K., 'System Modelling with High-Level Petri Nets', Theoretical Computer Science 13, North Holland Publishing Company, 1981.

Jensen, K., 'High-Level Petri Nets', 3rd. European Workshop on Application and Theory of Petri Nets, Villa Monastero, Italy, Sept. 1982.

Mazurkiewicz, A., 'Invariants of Concurrent Programs', IFIP INFOPOL'76, North Holland, 1977.

Misunas, D.P., 'Deadlock Avoidance in a Data-Flow Architecture', Computation Structures Group Memo 116, Massachusetts Institute of Technology, February 1975.

Rawling, M.W. and Zuk, E.A., 'Data-Flow Processing Element', Design 3 Manual, Department of Communication and Electronic Engineering, Royal Melbourne Institute of Technology, Nov. 1982.

Razouk, R.R. and Estrin, G., 'Modelling and Verification of Communication Protocols in SARA: The X.21 Interface', IEEE Transactions on Computers, Vol C-29 No.12, Dec. 1980.

Razouk, R.R. and Phelps, C.V., 'Performance Analysis Using Timed Petri Nets', Technical Report #206, Department of Information and Computer Science, University of California, Irvine, Aug. 1983.

Richardson, C.P., 'Object Recognition using a Dataflow Machine: Algorithms for a Laser Range-finder', M.Sc. dissertation, Department of Computer Science, Victoria University of Manchester, 1979.

Richardson, C.P., 'Manipulator Control Using a Data-Flow Computing System', Ph.D. Thesis, Department of Computer Science, Victoria University of Machester, 1981.

Rumbaugh, J. 'A Data Flow Multiprocessor', IEEE Transactions on Computers', February 1977.

Symons, F.J.W., 'The Application of Petri Nets and Numerical Petri Nets', Research Laboratories Report 7520, Telecom Australia, 1982.

Walkington, M., 'A Data-Flow Graphical Compiler', Design 2 Report, Department of Communication and Electronic Engineering, Royal Melbourne Institute of Technology, Nov. 1983.

Wathanasin, S., 'Proposed Language for the Data-Flow Multiprocessor', Internal document, Department of Computer Science, Victoria University of Manchester, 1978.

Weng, K.S., 'Stream-oriented Computation in Recursive Data-flow Schemas', Technical memo 68, Laboratory for Computer Science, Massachusetts Institute of Technology, Oct. 1975.

Wheeler, G., 'Numerical Petri Nets - A Tutorial', Draft Document, Switching and Signalling Branch, Telecom Australia, Dec. 1983.

Wilbur-Ham, M.C., et al., 'Protean User's Manual', Telecom Research Laboratories, Switching and Signalling Branch, Oct. 1983.

Whitelock, P.J., 'A Conventional Language for Data-Flow Computing', M.Sc. Thesis, Department of Computer Science, Victoria University of Manchester, October 1978.

Young, S., 'The Application of Computer Graphics to the Study of Numerical Petri Nets', 4th. Year Report, University of Melbourne, Nov. 1983.