An Analytical Study of the Time Variance in Parallelism in Dataflow Architectures

C.R.M.Sundaram and Y.Narahari
Department of Computer Science and Automation
Indian Institute of Science
Banglaore-560 012
India

Abstract

Evaluating the performance of parallel computations on parallel machines is an issue of great importance in the design of parallel processing systems. In this paper, we use an integrated modelling framework comprising Product Form Queueing Networks (PFQN's) and Generalized Stochastic Petri Nets (GSPN's) to model and evaluate the effect of the variance in parallelism in parallel computations. We focus on dataflow computations and present an integrated PFQN-GSPN model for the execution of dataflow computations on the Manchester dataflow machine. We characterize the parallelism in dataflow computations through a 4-parameter characterization comprising the minimum parallelism, maximum parallelism, average parallelism and the variance in parallelism. This model can be efficinetly solved. The numerical results indicate that the average parallelism is a good characterization of the dataflow computations only as long as the variance in parallelism is small. However, siginficant difference in performance measures will result when the variance in parallelism is comparable to or higher than the average parallelism.

1 Introduction

A number of small to medium scale multiprocessing architectures are now in wide use. However, the high cost of building such architectures can be justified only if they are efficiently utilized. Exploiting the parallelism in applications is a viable and effective approach to improve the performance of such systems. For exploiting the parallelism, it is necessary to characterize the parallelism in different applications through some common parameters. Parallelism in applications can be characterized at different levels of detail ranging from the full data dependency graph at one extreme to a single parameter such as the average parallelism, or the variance in parallelism at the other extreme. Detailed characterizations will not only be possible to get due to their possible dependency on data input and partitioning schemes, but also will be difficult to manage in a effective way. So, simple and effective characterizations are needed for this purpose. Such characterizations will be useful for two reasons. First, it will give the user an insight otherwise not clear, into the structure of his algorithm, and will facilitate the user in improving the efficiency of his algorithm. Second, theses characterizations which are simple and independent of different applications and their structures, will help in arriving at efficient resource management strategies and also will give the ability to treat different applications in a uniform way. Moreover, the overhead due to resource scheduling will be very minimal as the amount of information that has to be dealt with by the scheduler is very small.

We characterize the parallelism in applications by a 4-parameter characterization comprising minimum parallelism, maximum parallelism, average parallelism, and the variance in parallelism. Similar characterizations to ours have been considered in [4],[11],[12],[13]. Sevcik [12] analyzes the effects of the variance in parallelism through simulation. But, simulation models are computationally expensive, and for an approximate and quick feedback, analytical models are the preferred tools. Hong, Bhuyan, and Ghosal [11] consider the variation in parallelism(the difference between maximum and minimum parallelism). They make an approximate analytical study by assuming that the instantaneous parallelism of the applications remains constant over sufficiently long intervals of time, before it changes to another value. They analyze the effects of variation in parallelism by repeatedly evaluating an approoximate analytical model for different degrees of parallelism and combine the results using a weighted approach. Eager, Lazowska, and Zahorjan [4] consider a single parameter characterization namely average parallelism and obtain bounds on speedup, and efficiency in terms of the average parallelism and the number of processors. Shikharesh, Eager, and Bunt [13] consider one parameter namely the variability in parallelism (different from the variance in parallelism and the variation in parallelism) in addition to the average parallelism and obtain bounds on speedup, efficiency, etc., which are tighter than the bounds obtained using average parallelism alone.

In this paper, we are interested in performance evaluation of parallel algorithms on Multiprocessing and Dataflow Architectures. We introduce an analytical technique based on Product Form Queueing Networks(PFQN's) and Generalized Stochastic Petri Nets (GSPN's), *to* carry out the above evaluation. We illustrate our technique for the performance evaluation of dataflow computations when executed on the Manchester Dataflow Archiecture. Our technique is quite general, and it could be applied to any multiprocessing architecture, since the performance analysis of mulitprocessing and dataflow architectures can be made in *a* unified way [7],[11]. That is, the same models could be used with appropriate interpretations. We also compare our analytical technique with another technique based on a single parameter characterization namely the average parallelism in applications, considered by Ghosal and Bhuyan [5],[6],[8],[9]. Their models are closed queueing network (CQN) models, where the closed nature is ensured by assuming that the degree of parallelism exhibited by the datafiow graphs can be taken to be a constant, equal to the average parallelism.

Through extensive numerical experiments of our models, we find that average parallelism is a good characterization of the data flow computations only as long as the variance in parallelism is small compared to it, and a significant difference in performance will result if the variance in parallelism is comparable to the average parallelism. Also, we find that the model proposed here is remarkably accurate.

The rest of the paper is organized as follows. In Section **2,** we present the basic definitions and notation used in this paper. In Section 3, we represent the architecture of a prototype of the Manchester machine, then illustrate the importance of the variance in parallelism in dataflow computations through two simple examples. Then, we develop detailed GSPN models for the execution of dataflow computations on the Manchester machine and point out their state space explosion. In Section **4,** we give a brief introduction to the modelling tool considered in this paper. Then, we develop Integrated queueing network-petri net models and compare their efficiency with exact GSPN models. In Section **5,** we show the results of the detailed numerical experiments carried out on our models and compare them with CQN models proposed by Ghosal and Bhuyan. In Section 6, we preseat the important conclusions of our study.

2 Basic Definitions and Notation

The Average Parallelism of a job denoted by **A** is defined as the average number of busy processors when an unlimited number of them **are** allocated for the execution of the job. Putting in mathematical terms, $A = \sum_{j=m}^{j=M} j * p_j$, where p_j,m,and M are the fraction of time j processors are simultaneously busy, the minimum number of processors that are always busy, the maximum number of processors that are simultaneously busy, respectively,when an unlimited number of processors is available. Average parallelism can also be equivalently defined as $A = \frac{T_1}{T_\infty}$, where T_1 and T_∞ are the total execution times of the job on a single processor and on **an** unlimited number of processors. The variance in parallelism denoted by V is defined **as** $V = \sum_{j=m}^{j=M} (j-A)^2 * p_j$. The fraction of sequential portion of the job denoted by f can be computed using

$$f = \frac{T_\infty * p_1}{T_1} = \frac{p_1}{A}$$

The variability in parallelism denoted by $\omega(A)$ is defined as the ratio of the service time with **A** processors to the service time when the number of processors is unbounded. It can be computed using $\omega(A) = \frac{A}{S(A)}$. The variation in parallelism of a job denoted by V_1 is defined as the difference between the minimum and the maximum parallelisms of the job, i.e., $V_1 = M - m$.

3 The Variance in Parallelism in Dataflow Computations

The closed queueing network models presented by Ghosal and Bhuyan [5],[6],[9], for the execution of dataflow graphs on the ManChester machine assume that the degree of parallelism in the dataflow computations takes a constant value, namely **A,** the average parallelism. However, in an actual execution, the degree of parallelism varies with time, and this variation is found to have **a** significant impact on their performance in our study. In this section, we illustrate the importance of the variation in parallelism in dataflow computations through two simple examples.

First, We give below a brief overview of the Manchester machine. **A** prototype of the Manchester dataflow system is shown in Figure 1. I/O switch acts as an interface between the host unit and the ring. The tokens from the host contain the values to initialize the arcs of the underlying dataflow graph. The tokens from the ring contain both final and intermediate results.The final results are sent to the host whereas the intermediate results are sent to the token queue. The token queue is a FIFO buffer. Its main function is to smoothen the flow of tokens in the ring strcture. The tokens from the token queue are sent to the matching unit. Its main function is to match the incoming tokens from their partner tokens. Unmatched partner tokens are written into the overflow unit. Matched tokens are sent to the node store unit whose function is to form the executable packet containing the instruction and the operands. Processing unit contains a number of functional units. Executable packets formed by the node store unit is executed in one of the functional units. The functional units generate result tokens which are sent to I/O switch for further processing.

The number of active instructions at any time in an execution of **a** dataflow graph is the degree of parallelism at that time instant. Consider the dataflow graphs shown in Figures **2** and **3.** Figure 2 shows the dataflow graph for the evaluation of the expression $\frac{-b+\sqrt{b^2-4*a*c}}{2*a}$, where as Figure **3** shows the dataflow graph for the evaluation of the following program segment:

$$\begin{aligned} P &= X + Y \\ Q &= \frac{P}{Y} \\ R &= X * P \\ S &= R - Q \\ T &= R * P \\ \text{Result} &= \frac{S}{T} \end{aligned}$$

If such dataflow graphs are executed on the Manchester machine, even **if** all the instructions get executed in more or less the'same time, the degree of parallelism varies due to the structure of the computation. The degree of parallelism can also vary due to the difference in the execution time of instructions and the availability of functional units. Moreover, this variation in parallelism with time can occur not only during the execution of a single dataflow graph, but also due to the presence of dissimilar dataflow graphs in the job queue of Figure 1.

Now, we develop a GSPN model which incorporates this variation in parallelism. The GSPN model presented below is based on the following assumptions: (i) The service times at all the service centers namely, matching unit, node store unit, and processing unit are exponentially distributed. (ii) The matching store is large enough so that overflow unit is not required. (iii) The degree of parallelism in a dataflow graph can be characterized by a 4-parameter charcterization comprising the minimum, maximum, average and the variance in parallelism. (iv) The degree of paralleiism changes in steps of 1 only, i.e., it either increases by 1, or decreases by 1. This assumption is justified since the execution time of functional units is exponentially distributed and the probability of two or more activities with exponentially distributed durations ending simultaneously is zero.

Figure **4** shows the GSPN model for the execution of dataflow graphs on the Manchester machine. The reader is referred to [I]for GSPN details. The interpretation of the places, and the transitions is given in Table 1. Figure 4 becomes self-explanatory with Table 1. The random switch t_8, t_9, t_{10} models the variance in parallelism. The firing of the transition t_9 increases the degree of parallelism by I, and the firing of t_{10} decreases the degree of parallelism by 1. **A** change in parallelism of +1 is due *to* the processing unit generating two result tokens and a change in parallelism of -1 is due to processing unit generating no tokens. In this GSPN model, the loss of tokens is not necessarily equal to the gain of tokens, which makes'the parallelism of the underlying dataflow graph to vary. The degree of parallelism, d(M), in a marking M of the GSPN model is given by

$$d(M) = \sum_{i=1, i\neq 4, i\neq 8}^{10} M(P_i)$$

Consider the execution of a dataflow graph having an average parallelism **A,** a minimum parallelism $A - V_1$, and a maximum parallelism $A + V_1$. Here, V_1 is the variation in parallelism from the average parallelism. For modelling execution of such a dataflow graph, the place P_1 is initialized with **A** tokens. The probability q_1(M) in a marking M is set to zero when d(M) equals $A + V_1$. This eliminates the increase of parallelism above $A + V_1$. The probability q_2(M) in a marking M is set to zero when d(M) equals $A - V_1$. This does not allow the decrease of parallelism below $A - V_1$. Thus the degree of parallelism varies between $A - V_1$ and $A + V_1$.

When d, the degree of parallelism is between $A - V_1$ and $A + V_1$, the probabilities q_1 and q_2 take positive values. The values taken by them depend on V_1 due to the following reason. The sojourn times are very low in the markings having a degree of parallelism clsoe to $A + V_1$ (because of the high throughputs in these markings). Similary, the sojourn times are very high in the markings having a degree of parallelism close to $A - V_1$ (because of the low throughputs in these markings). So, this makes the systems to spend more time in the states with a low degree of parallelism, than the states with a high degree of parallelism, and hence the average parallelism decreases with the increase in the variation in parallelism. So, we adjust the probabilities q_0, q_1, and q_2 so that the average parallelism **A** remains the same for all the variations. For an average parallelism 10, Table 2 shows the probabilities q_0, q_1, and q_2 so that the average parallelism **A,** remains the same for all the variations in parallelism. The variance in parallelism, V, is given by model can be calculated by

$$V = \sum_{M \in S} (d(M) - A)^2 * SSP(M)$$

where SSP(M) is the steady-state probability of marking M. S is the reachable set of the GSPN model, and d(M) is the degree of

parallelism in the marking M. The values of V are also shown in Table 2 for different values of V_1. As expected, the variance in parallelism increases with the increase in V_1.

The cardinality of the state space of the exact GSPN model for the execution of a dataflow graph with an Average Parallelism A, and a variation V_1 is given by $\sum_{i=A-V_1}^{A+V_1} \frac{(i+2)!}{i!2!}$.

As explained above, in any marking M, the degree of parallelism d(M) varies between $A - V_1$ and A t V_1. When the degree of parallelism takes a value of i, the i tokens can be distributed across the queues of three service centers of the Manchester machine in $\frac{(i+2)!}{i!2!}$ ways. Then, the total number of states is the sum over the number of states of having a parallelism of i, where i varies from the minimum parallelism $(A - V_1)$ to the maximum parallelism $(A + V_1)$.

As the expression shows, studying the effects of the variance in parallelism in dataflow computations through exact GSPN models will not be practically feasible, as the number of states grows exponentially with increase in average parallelism and the variation in parallelism.

4 Integrated Queueing Network-Petri Net (IQP) Models for the Variance in Parallelism

The IQP modelling originated by Balbo,Bruell,and Ghanta [2], [3] is based on the concept of flow-equivalence. The basic idea is to isolate the product form and non-product form portions of a given system, evaluate them independently, and combine them through flow-equivalents. The overall model (.alsocalled the high level model) of a given system will then have some subsytems represented by the flow-equivalents. These flow- equivalents of the subsytems (also called the submodels) are computed by evaluating the throughpnt of these subsytems in isolation for all possible populations. The throughputs thus computed together with associated populations 'constitute the flow-equivalent. The IQP modelling can be of two types: (ij IQP model with PFQN as the high level model and (ii) IQP model with GSPN as the high level model. If the high level model is a PFQN model, then some of its stations will be the flow-equivalent representations (derived using GSPNs) of the submodels that incorporate non-product form features. If the high level model is a GSPN model, then some of its timed transitions will be the flow-equivalent representations (derived using PFQNs) of the submodels that are product form solvable. Since exact represntations can be derived for the flow-equivalents, the only source of error in this technique could be due to the interaction between the product form and the non-product form portions. When the interaction is weak, this technique will produce remarkably accurate results.

The development of IQP models for the execution of the dataflow computations on the Manchester machine are motivated by the following two observations:(i) The non-product form feature namely the variation in parallelism cannot be analyzed through product form queueing networks. GSPNs can elegantly model the variation in parallelism. But, analyzing the GSPN models is practically infeasible as the cardinality of the models grows exponentially. (ii) The service rates of the matching and node store unit are 10 times more than the processing capacity of the functional units. These rates were taken from [10]. Consequently, between two successive interactions between the processing unit and the rest of the system, the subsystem containing the node store unit and the matching unit would have virtually reached a steady-state. In such a case, the interaction is weak, and the integrated technique would give much less error.

In these models the non-product form feature namely the variation in parallelism is captured by a GSPN model and the rest of the product form features by a PFQN model. The two models interact through a flow-equivalent server. The GSPN part is the high level model and the PFQN part is the submodel. The PFQN part is represented in the GSPN part by a marking dependent timed transition. Figure 5(a) shows the high level GSPN model. Figure 5(b) shows the PFQN part of the model. As in the exact GSPN model of Figure 4, the random switch t_2, t_3, t_4 captures the variation in parallelism. In the GSPN high level model, the sub PFQN model is represented by the marking dependent timed transition "t_5". The firing rate of this transition will vary from marking to marking. The rate of the transition 't_5' in a marking M is obtained by solving the PFQN model of Figure 5(b) with a population of $M(P_4)$ customers. As the number of tokens in the place 'P_4' can vary from $A - V_1$ to $A + V_1$, $2 * V_1$ evaluations of the PFQN part will be required. These $2 * V_1$ evaluations together with the associated populations constitute the flow-equivalent. Along the same lines for the exact GSPN model, the cardinality of the GSPN high level model for execution of a dataflow graph with an average parallelism A, and a variation V_1 is given by $\sum_{A-V_1}^{A+V_1}(i+1)$. Comparing this expression with that for the exact GSPN model, one can easily observe that there will be a drastic reduction in the size of the state space. Effectively we have reduced an evaluation of a GSPN model into $2 * V_1$ evaluations of a 2-node PFQN, for whose performance measures a closed form expression can be obtained, followed by an evaluation of a GSPN whose cardinality of the state space is much smaller than that of the exact GSPN model. Now, it would be very easy to study the effect of the variation in parallelism as the experimentation is required on a small-sized GSPN model.

5 Numerical Results

In this section, we study the effects of the variance in parallelism in dataflow computations. For this purpose, we evaluated the IQP models for the values of 5, 10, and 20 for A. In each case, V_1, the variation in Parallelism, is varied over a wide range of values.

5.1 Processing Power

Processing power of the Manchester machine (i.e., the average number of busy functional units) is an important performance measure. Table 3 gives the processing power of the machine for A = 20 and various values of V_1 ranging from 3 to 11. The variance in parallelism, V, is also presented for various values of V_1. The third entry in the table is obtained by the evaluation of CQN models proposed by Ghosal and Bhuyan[5]. We observe from this table that, there is a close agreement in the processing power when the variance in parallelism is small. However, there is a significant decrease in the processing power when the value of V increases. Similar results are shown by Table 4, which shows the processing power for a value 5 for A, and V_1 from 1 to 5. But, this table shows much more significant decrease than in the previous table, since the average parallelism itself has a low value. For example, when V_1 =4, the variance in parallelism is approximately 1.45 times the value of the average parallelism. Even for such a low ratio of the variance in parallelism to average parallelism, we get 16.3% difference in the processing powers predicted by the average parallelism model and our model. Table 5 shows the processing power of the Manchester machine for a value of 10 for A, and various values of V_1 from 1 to 7, and for various number of functional units. The decrease in the processing power is more pronounced when the number of functional units is high, than the case when the number of functional units is low. The detailed explanations on why the processing power decreases with the increase in the variance in parallelism can be found in [14]. They are not presented here due to space constraints.

So, there will be a very high decrease in the processing power when the average parallelism is low, the ratio of the variance in parallelism to the average parallelism is high, and the number of functional units is also high. This decrease will be much more than 18.1%which is the percentage of differencein the processing powers predicted by the average parallelism model and our model, for A = 20, V = 160.7, and 10 functional units (Table 3).

5.2 Mean Queue Lengths

In this section, we study the effects of the variance in parallelism on the mean queue lengths at various service centers of the Manchester machine. Table 6 shows the change in the mean queue lengths at the Matching unit, the node store unit, and the processing units for A =10, and various values of V_1, and 5 functional units.

The variation in the mean queue length of the processing unit is due to the following reason. When V_1 takes the values 1,2, or 3, the decrease in the mean queuelength when the system visits states having a degree of parallelism 9,8, or 7, is more than the increase obtained when the system visits having a degree of parallelism 11,12, or 13. So, on the whole, there is a net decrease in the mean queue length. However, when V_1 takes values 4,5,6,or 7, there is no decrease in the mean queuelength due to the system visiting states having a degree of parallelism 3,4,5,or 6, as there are 5 functional units. But, there is an increase in the mean queuelength due to the system visiting states having a degree of parallelism 14,15,16 or 17. So, there is a net increase in the mean queuelength when V_1 takes values above 3.

The mean queuelengths for the matching and node store units exhibit a reverse variation. This is because, the mean queuelengths in the matching unit, the node store unit, are complementary in nature. When the mean queuelength in the processing unit increases, the matching and node store units exhibit a decrease and vice-versa.

5.3 Accuracy of Results

Since we have used IQP models for all our experiments, we investigated the accuracy of this technique. Table 7 shows the variance in parallelism calculated by the integrated models and the exact GSPN models at various levels of variation in parallelism. Since there is such close agreement in the values of the variance in parallelism between these two models, all the performance measures such as processing power, throughput, mean queuelengths at various service centeres computed by the integrated models are found be very accurate.

6 Conclusions

In this paper, we have made a performance analysis of dataflow computations when executed on the Manchester machine. The parallelism in dataflow computations has been characterized using a 4-parameter characterization. An analytical technique based on queueing networks and Petri nets has been introduced to carry out the performance analysis. This technique is applicable for the performance analysis of parallel computations on any general multiprocessor architecture, as the same models could be used with appropriate interpretations in the case of multiprocessors. By comparing the complexity of analysis with that of exact GSPN models, we have found it to be very efficient. The performance measures computed using the integrated models were also found to be very accurate. From the numerical results and their interpretations presented in the previous section, we conclude that the variance in parallelism affects the performance measures such as the processing power, mean queuelengths, when it is comparable to or higher than the average parallelism. We believe that this observation is also true for parallel computations. Eventhough we have conducted experiments for a small set of values, our interpretations of the results indicate that similiar behaviour will be exhibited at all levels of the average parallelism. These investigations on the effects of the variance in parallelism in the dataflow computations on their performance have been possible, because the integrated models provided us efficient solutions having an accuracy comparable with that of exact GSPN modelling approach.

References

[1] M.Ajmone Marsan, G.Balbo and G.Conte, A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems, **ACM Transactions of Computer Systems,** Vol.2, No.2, May 1984, pp.93-122.

[2] G.Balbo, S.C.Bruell, and S.Ghanta, Combining Queueing Network and Generalized Stochastic Petri Net Models for the Analysis of Software Blocking phenomena, **IEEE Transactions on Software Engineering,** Vol.12, No.4, April 1986,pp.561-576.

[3] G.Balbo, S.C.Bruell and S Ghanta, Combining Queueing Networks and Generalized Stochastic Petri Nets for the Solution of Complex Models of Computer Systems, **IEEE Transactions on Computers,** Vol.17, No.10, October 1988, pp.1251-1268.

[4] D.L.Eager, J.Zahorjan and E.D.Lazowska, Speedup Versus Efficiency in Parallel Systems, **XEEE Transactions on Computers,** Vol.38, No.3, March 1989, pp.408-423.

[5] D.Ghosal and L.N.Bhuyan, Analytical Modelling And Architectural Modifications of a Dataflow Architecture, **Proceedings** of **the Fourteenth Annual Symposium on Computer Architecture,** June 1987, pp.81-89.

[6] D.Ghosal and L.N.Bhuyan, Performance Analysis of the MIT Tagged Token Dataflow Architecture, **Proceddings** of **the Seventeenth International Conference on** Parallel **Processing,** 1988, pp.680-682.

[7] D.Ghosal, **A Unified Approach to Performance Evaluation of Dataflow and Multiprocessing Architectures,** Ph.D.Dissertation, The Centre for Advanced Computer Studies, University of Southwestern Lousiana, 1988.

[8] D.Ghosal, S.K.Tripathi, L.N.Bhuyan and H.jiang, Analysis of Computation-Communication Issues in Dynamic Dataflow Architectures, **Proceddings** of **the Sixteenth Annual Symposium on Computer Architecture,** 1989, pp.325-333.

[9] D.Ghosal and L.N.Bhuyan, Performance Evaluation of a Dataflow Architecture, **IEEE Transactions on Computers,** Vol.39, No.5, May 1990. pp.615-627.

[10] J.R.Gurd, I.Watson, and C.C.Kirkham, The Manchester Prototype Dataflow Computer, **Communications of ACM,** Vol.28, No.1, January 1985, pp.34-52.

[11] Hong Jiang, L.N.Bhuyan, and D.Ghosal, Approximate Analysis of Multiprocessing Task Graphs, **Proceeding;** of **the nineteenth Parallel Processing Conference,** August 1990.

[12] K.C.Sevcik, Characterizations of Parallelism in Applications and Their Use in Scheduling, **Performance Evaluation Review,** Vol.17, No.1, May 1989, pp.171-180.

[13] Shikharesh Majumdar, D.Eager, and R.B.Bunt, Characterisation of Programs for Scheduling **in** Multiprogrammed Parallel Systems, to appear in **Performance Evaluation.**

[14] C.R.Meenakshi Sundaram, **Integrated Analytical Models for Parallel and Distributed Computing Systems,** Department of Computer Science and Automation, Indian Institute of Science, October 1990.

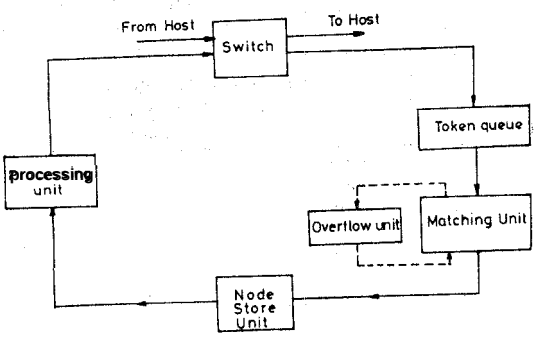

Fig. 1 The Manchester Dataflow Computer

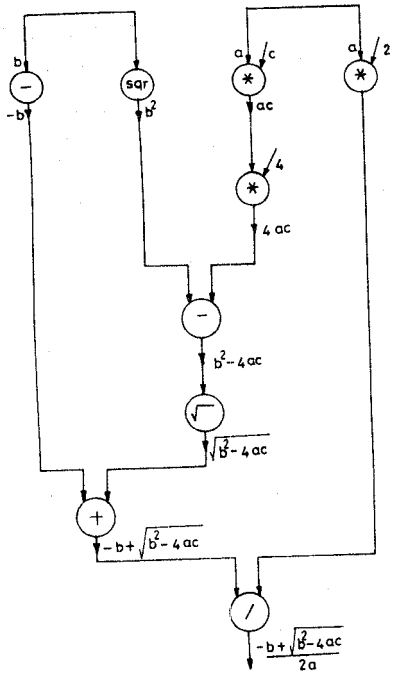

Fig. 2 Example Data Flow Graph 1

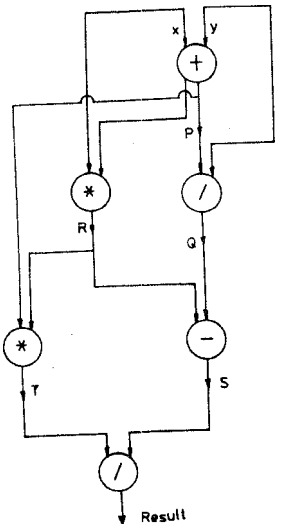

Fig. 3 Example Data Flow Graph 2

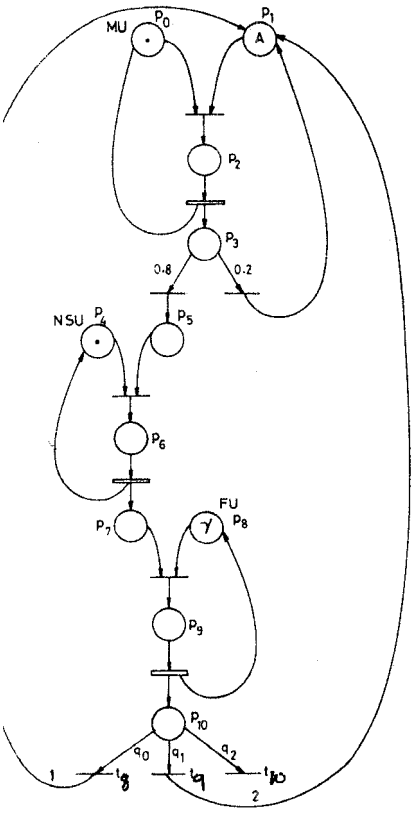

Fig. 4 GSPN Model of the prototype incorporating time Variance of Parallelism

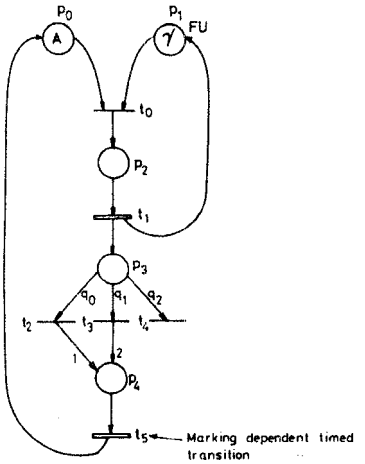

Fig. 5 (a) GSPN Port of the Integrated Analytical Models

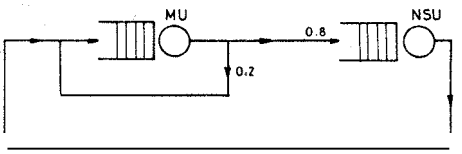

Fig. 5 (b) PFQN Part for the Basic Machine

Places:
P_0, P_4, P_8: availability of MU,NSU,and PU.
P_1, P_5, P_7: Tokens waiting for MU,NSU,and PU.
P_2, P_6, P_9: Tokens getting executed in MU,NSU,and PU.
P_3, P_{10} : Matching and Processing operations have just finished.

Immediate Transitions:

t_0, t_4, t_6: starting of the operations of MU,NSU,and PU.
t_2:Matching operation is successful.
t_3:Matching operation is unsuccessful.
t_8, t_9, t_{10}:Procesing unit generates no tokens, exactly one token, and two tokens.

Exponential Transitions:

t_1, t_5, t_7: Execution of tokens by MU,NSU,and PU.

Random Switch: Dynamic

t_8-q_0; t_9-q_1; t_{10}-q_2;

Initial Marking:

P_0-1; P_1-A; P_4-1; P_8-r;
where r is the number of functional units.

Table 1.
Legend for GSPN Model in Figure 1

V_1 (Variation in parallelism	q_1 (Probability that parallelism increases)	q_2 (Probability that parallelism decreases)	A (Average parallelism)	V (Variance in parallelism)
10±7	0.2543	0.2457	10.00	19.2958
10±6	0.2530	0.2470	9.99	13.8154
10±5	0.2521	0.2479	10.00	9.5059
10±4	0.2514	0.2486	10.00	6.1745
10±3	0.2509	0.2491	9.99	3.6245
10±2	0.2505	0.2495	9.99	1.7782
10±1	0.2500	0.2500	9.99	0.6000

Table 2:The values of the probabilities for different values of the variations in parallelism.

Average Parallelism = 20;
Number of Functional Units = 10;

Variation in parallelism	Variance in parallelism	Average parallelism model	Variance model (integrated analytical model)	%difference in processing power
20±3	3.6173	8.774	8.743	0.353
20±7	17.747	8.774	8.659	1.311
20±11	43.666	8.774	**8.442**	3.784
20±15	85.273	8.774	8.025	8.537
20±19	160.672	8.774	7.190	18.053

Table 3: Processing power of the Manchester machine.

Average Parallelism = 5;
Number of Functional Units = 3;

Variations in parallelism	Variance in parallelism	Average parallelism model	Variance model (integrated analytical model)	%difference in processing power
5±1	0.6005	2.92	2.894	0.891
5±2	1.8067	2.92	2.830	3.082
5±3	3.9205	2.92	2.675	8.391
5±4	7.3557	2.92	2.445	16.268

Number of functional units	10±1	10±2	10±4	10±5	10±6	10±7
1	0.999	0.999	0.999	0.999	0.999	0.999
3	2.999	2.999	2.995	2.989	2.975	2.940
5	4.922	4.905	4.828	4.749	4.611	4.400
7	6.252	6.187	5.949	5.755	5.505	5.176
9	6.724	6.617	6.314	6.097	5.820	5.458

Table 5: Processing power of the Manchester machine.

Average Parallelism = 10;
Number

Variation in parallelism	Processing unit mean queue length	Matching unit mean queue length	Nodestore unit mean queue length
0	3.25	0.514	0.335
1	3.09	0.661	0.364
2	3.04	0.711	0.378
3	3.05	0.723	0.380
4	3.08	0.715	0.373
5	3.12	0.694	0.360
6	3.16	0.662	0.342
7	3.17	0.619	0.318

Table 6: Mean queue lengths of the PU,MU,and NSU of the Manchester machine.

Variation in parallelism	Average parallelism (Exact GSPN Model)	Average parallelism (Integrated analytical Model)	Variance in parallelism (Exact GSPN Model)	Variance in parallelism (Integrate! analytical Model)
10±1	9.999	9.999	0.600	0.600
10±2	9.999	10.000	1.779	1.778
10±3	10.000	**9.999**	**3.627**	**3.624**
10±4	10.000	9.998	6.179	6.174
10±5	10.000	9.996	9.513	9.506
10±6	9.999	9.992	13.825	13.815

Table 7: A comparison of the values of the average parallelism and the variance in parallelism computed by the exact GSPN models and the integrated analytical models.