PRP Transition:

IOMS:

IOM_1:
Input Marking: ({tokentype.input place}...)
Output Marking: ({tokentype.output place}...)

IOM_2:
Input Marking: ({tokentype.input place}...)
Output Marking: ({tokentype.output place}...)

. .
. .
. .
. .

TDFS:

TDF_1:
TDF type:
TDF parameters:

TDF_2:
TDF type:
TDF parameters:

. .
. .

Figure 4.6

Definition : A **PRP token** is a composite entity defined as follows (Figure 4.7):

1. Token type: This corresponds to token color in colored Petri nets.

2. Token delay counter (DC): This is a down counter that is loaded from an appropriate transition delay function (as defined under PRP transition) when a token enters an enable slot. This counter is decremented by a clock until it becomes zero.

3. Token delay initial (DI): This is used to hold the initial value loaded into the DC.

4. Token wait counter (WC): This counter is zeroed when a token enters a place wait set; it is incremented by a clock until the token passes on into the place enable slot.

5. Token arrival in wait set time stamp (AWS): When a token enters a place wait set its AWS is stamped with the sum of the maximum AES (Definition. 6) and the DI (Definition. 3) of the tokens from the input marking (defined under PRP transition) of the transition that fired and created the new token.

   AWS = previous DI + maximum [previous AES]

6. Token arrival in enable slot time stamp (AES): This stamping is done when a token advances from the wait set into the enable slot of a place due to the enabling of a succeeding transition. The value stamped is the sum of the token arrival in wait set time stamp and the token wait counter.

   AES = AWS + WC

For reconfigurable machines tokens represent distinguishable switchable memories. The flow of the contents of the switchable memories is mimiced in this model.

PRP Token:

Type:

AWS : Arrival in wait set stamp

WC : Wait counter

AES : Arrival in enable slot stamp

DC : Delay counter

DI : Delay initial

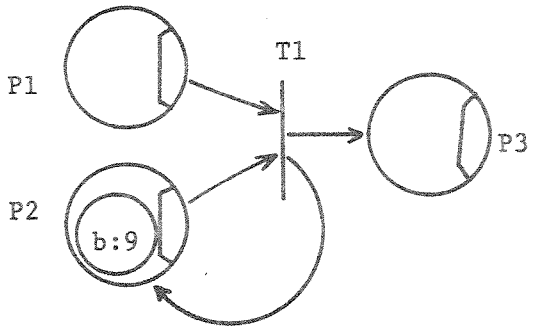
Figure 4.7

A transition is in a 'quiescent' state as long as none of its input markings are matched; (a match occurs when the place wait sets of the input places in any of the input markings of the IOMS contain the tokens defined for that input marking). The instant a match occurs and the place enable slots for the necessary input places (i.e. the places where a token will be advanced into place enable slots) are empty, the transition becomes 'enabled' and tokens are moved into appropriate place enable slots. These tokens are given AES time stamps, their DC and DI are loaded and all the DC begin down counting. When all the DC (simultaneously) count down to zero, the 'enabled' transition 'fires'. This 'firing' removes tokens from the place enable slots. New tokens, belonging to the output marking corresponding to the input marking that 'fired' the transition are created. These tokens are given AWS time stamps and they are deposited into the wait sets of appropriate output places. After a transition 'fires' it stays in a 'quiescent' state until it is 'enabled' again.

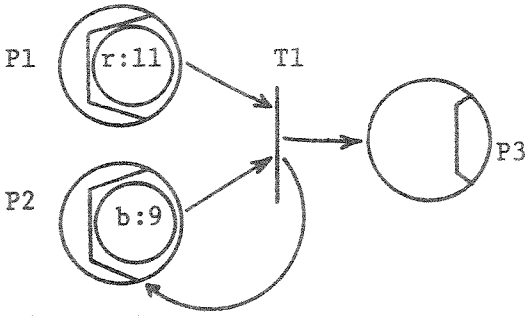
The primitives defined are now exercised to expose the working of the model. Consider the sequence of states shown in Figure 4.8; the PRP segment shown consists of two input places, two output places and a transition. The input output mapping set for the transition has two members and there is a time delay function (a constant in the example shown) associated with each mapping.

In figure 4.8.1 token 'b' enters the wait set at place p2 and is time stamped 9. In figure 4.8.2 token 'r' enters the wait set at place p1 and is time stamped 11; at this instant transition t1 becomes enabled because its input token requirement is fulfilled- the tokens immediately enter the enable slots in places p1 and p2 where they reside for 3 time units (as determined from the appropriate delay function for t1). In figure 4.8.3 transition t1 has fired: this results in the placement of tokens 'r' and 'b' in places p2 and p3 respectively where the tokens are time stamped with their arrival times.

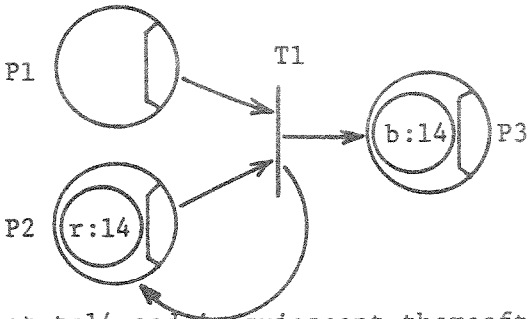
This model of behavior is similar to the E-net [Noe71] definition for the time delay function. The delay is specified at a transition and is applied to an appropriate input ensemble of tokens; the tokens however, reside within the input places for the duration of the delay. This is a direct representation of processing with simultaneously held resources. The delay function retains data path sensitivity in the sense that if t1 were to be enabled with a different distribution of input tokens (eg. 'y', 'g' as in figure 4.7) or if the tokens were destined to one of a number output mappings, a different delay function could be specified. This is in contrast to Berlin [Ber79], where the delay function is associated with tokens: delays associated with processing are invoked in a vacuum, insensitive to the path taken as a result of the firing. The wait for a



4.8.1:
T1 quiescent till t=9



4.8.2:
T1 enabled at t=11.



4.8.3:
T1 fires at t=14 and is quiescent thereafter

Figure 4.8: Quiescence, Enabled and Fire
PRP Transition States.

IOM_1

	1	2	3
U_1	r	b	ϕ
V_1	ϕ	ϕ	ϕ
R_1	ϕ	r	b

$TDFS_1$

TDF type	Deterministic
TDF param.	3

Figure 4.9: PRP Transition T1 From Figure 4.8 Using SOS Notation

parallel algorithms executing on reconfigurable computers. A modelling structure is necessary to project performance behavior on reconfigurable machines with different resource and speed parameters and to validate the behavior observed on available machines.

An abstract model of the time resolved behavior of such parallel programs was developed. This model incorporates features of queuing networks and Petri nets where multiple resource holding and process blocking is modelled. The parallel program is specified in an input language that creates data structures that are exercised to obtain run times and run time components. The input language supports hierarchical composition in the same form as the Computation Specification Language for parallel programs on TRAC: the degree of parallelism and interconnection geometries of configurations corresponding to phases are represented.

5.2 DIRECTIONS FOR FURTHER WORK

The parallel programming of reconfigurable architectures is a sufficiently young endeavor such that no one methodology has been completely validated, nor is it clear which mechanisms of the architecture will be prominently used. The underlying mechanisms and the methodologies of programming that are based on them are areas open for further study.

Within the context of the methodology and the mechanisms considered in this dissertation, algorithms in which block data flow requirements can be isolated are good prospects for study. And as additional insight is gained into the use of the architecture, the approach needed in the modelling of program execution will require modification to more precisely reflect heavily used features.

A bias in favor of the use of switched memories permeates this entire report. A similar study using packet switching, beginning with a quantification of packet flow characteristics, is a fertile area. A comparative study of packet use versus switched memory use will be useful.

The problem of executing programs in the situation where insufficient resources are present to schedule a configuration has not been addressed This is an important open area that must be studied in depth, both from the viewpoint of machine and program reconfiguration.

Block algorithms are used frequently in numerical algorithms. In linear system solution, block iterative methods deserve further study. Blocking is seen as a form of program structuring similar to, but less restrictive than vectorization of codes for pipelined computers.

Thus, almost all aspects of reconfigurable machine design and use are fertile areas for further work.

1.3 EXAMPLE

Figures A.3 show the pictorial and PRP specification language description of a net fragment for the 2-3 (Figure 3.2a, 3.3) switch in OEE and OER. This switch is the second step of the elimination (reduction) stages. Sample template definitions are given in figure A.3b and figure A.3c shows the subnet definition.

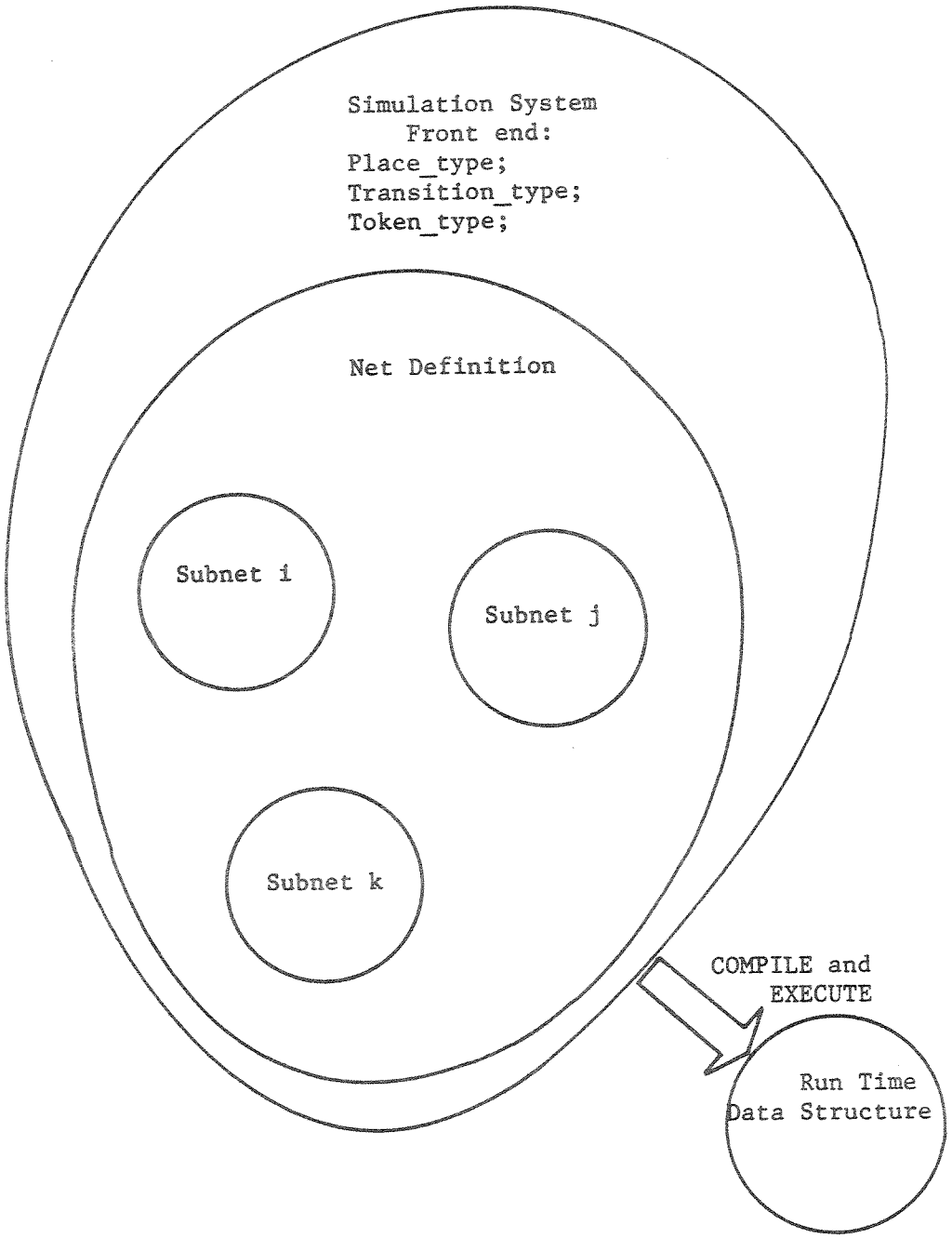


Figure A.1: PRP Simulator Front End Operation

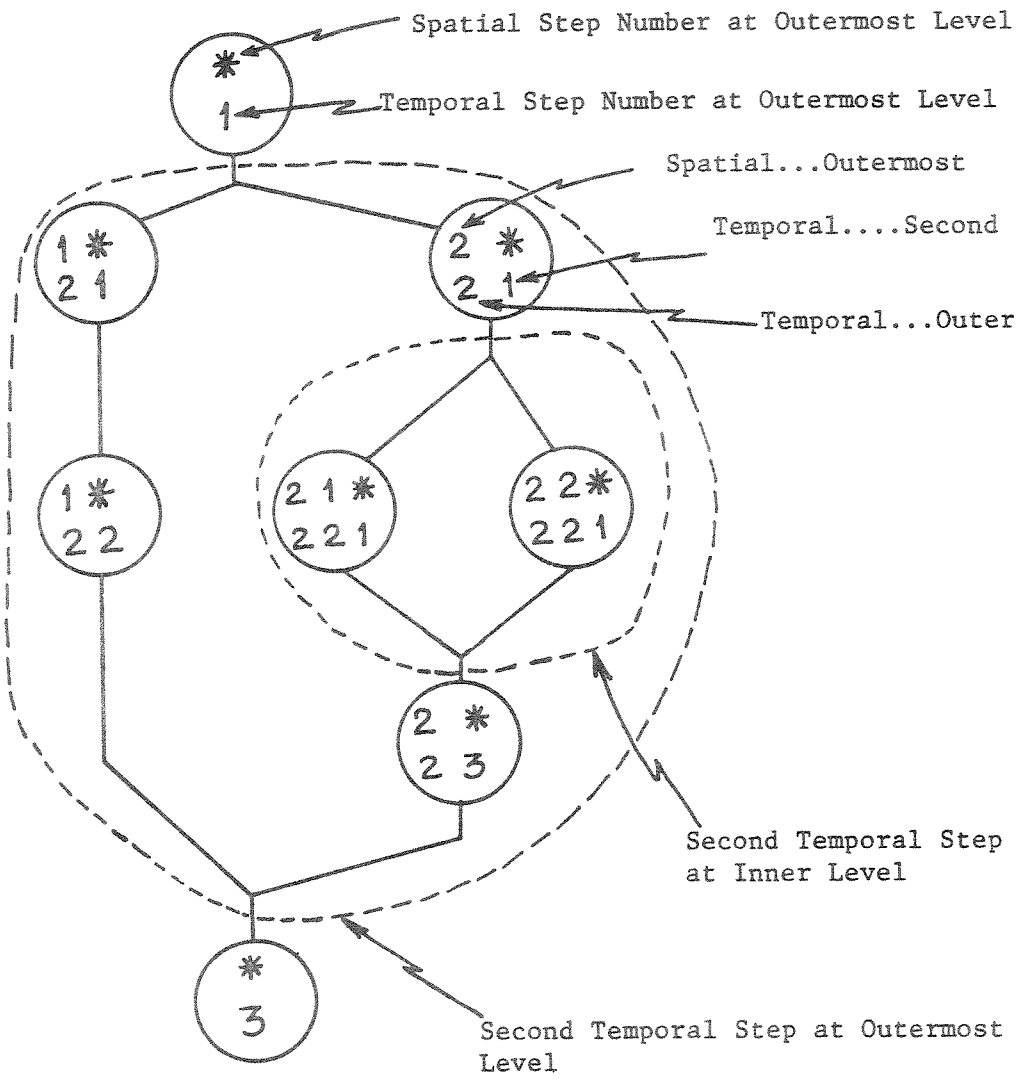


Figure A.2a: Subnet Tags

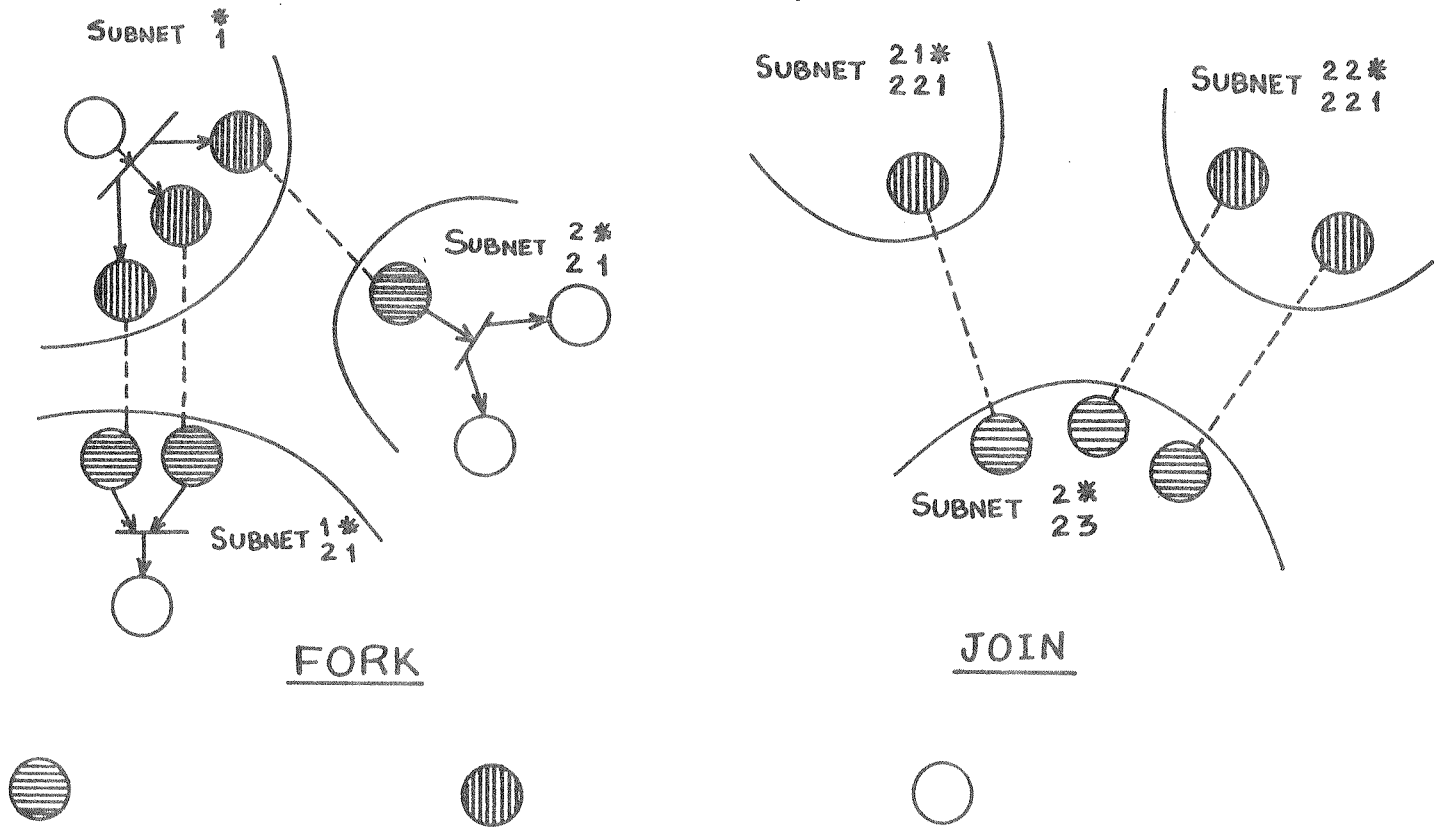


Figure A.2 b: Internet Binding for Fork and Join

and Vt[p]

= -1 if (p,t) belongs to I
and (t,p) does not belong to O

= +1 if (t,p) belongs to O
and (p,t) does not belong to I

= 0 if (t,p) belongs to O
and (p,t) belongs to I.

Moreover, the vector q0 of a VRS corresponds to the initial state and q - t -> q' in a VRS corresponds to the action of a transition t.

The idea of a Petri net can be generalized to allow parallel arcs between places and transitions. Keller [Kel74] shows this generalized Petri net equivalent to VRS.

2.2 SOS AND VRS

The relationship between colored Petri nets and Petri is similar to the relationship between SOS and VRS. A constructive method used by Peterson [Pet80] in the case of Petri nets is adapted to show the correspondence between SOS and VRS.

Let the cardinality of the set M=[m1,m2,m3,....mn] be n (n finite), i.e. there are n distinguishable tokens in the sense

of colored Petri nets in a SOS. This SOS can be transformed into another system that is a VRS (with normal transition rules) as follows.

Each vector element in the SOS is mapped into a set of vector elements and each member of the indexing set is mapped into a set of members of the indexing set of the new system. Hence there is a homomorphic mapping from the new system, its state space and transition sequences into the old net, its state space and transition sequences, and an inverse homomorphic mapping in the other direction.

The new system is created by replicating the vectors in the old system n times. Thus for each vector element qi we define a set of elements qi.1,qi.2,.... .,qi.n. We then redefine the triple (Uj,Vj,Rj) as the pair (Uj.new,Vj.new) (with the dimensions of the new vectors equal to d*n) such that:

each element of the new vector, Uj.new[i.c] =
 -cardinality of the subset of Uj[i] of type c,

and each element of the new vector Vj.new[i.c] =
 -cardinality of the subset of Vt[i] of type c
 +cardinality of the subset of Rt[i] of type c.

Figure B.1 is the VRS equivalent of the SOS shown in figure 4.4.

2.3 PRP AND COLORED PETRI NETS

The basic definition of the PRP model is structurally equivalent to the definition of colored Petri nets. Colored Petri net tokens are uninterpreted except for tokens, which are 'typed' (colored) in a manner analogous to the 'typing' of PRP tokens.

Consider the following construction: In a PRP model strip away all primitive interpretations except token type and place enable slots to create a reduced PRP. The enable state and firing of a transition collapses into an atomic firing and the residency time of tokens in place enable slots is reduced to zero. A transition firing removes at most one token from a place in a reduced PRP. To each reduced PRP there corresponds a colored Petri net which contains no multiple input arcs to any transition from a place. Conversely, for each such colored Petri net, there corresponds a reduced PRP.

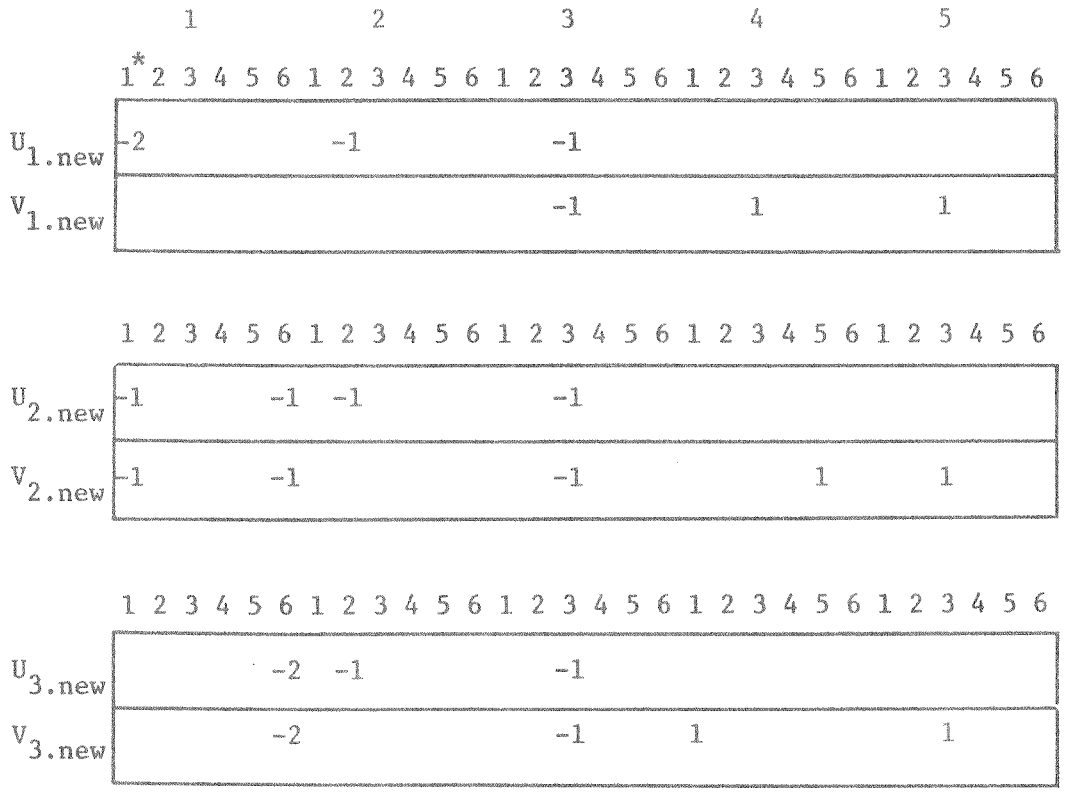
2.4 SUMMARY

Figure B.2 shows the relationships between the models discussed so far. Petri Nets can be transformed into a restricted class of Vector Replacement Systems; 'Generalized' Petri nets that allow parallel arcs between places and transitions are equivalent to Vector Replacement Systems

[Kel74].

SOS and Colored Petri Nets are notational extensions to Generalized Petri Nets and VRS respectively that aid in model compaction. PRP is a time extension to SOS (or colored Petri nets).

The essential difference between VRS and Petri nets is notation- VRS uses an algebraic set of primitives as opposed to the symbolic primitives in Petri nets. VRS is directly representable on a computer while Petri nets must be simulated. A similar distinction exists for SOS and colored Petri nets.



* 1 - Red
2 - Black
3 - Blue
4 - Yellow
5 - Orange
6 - Green

Figure B.1: VRS Equivalent Of SOS Of Figure 4.4

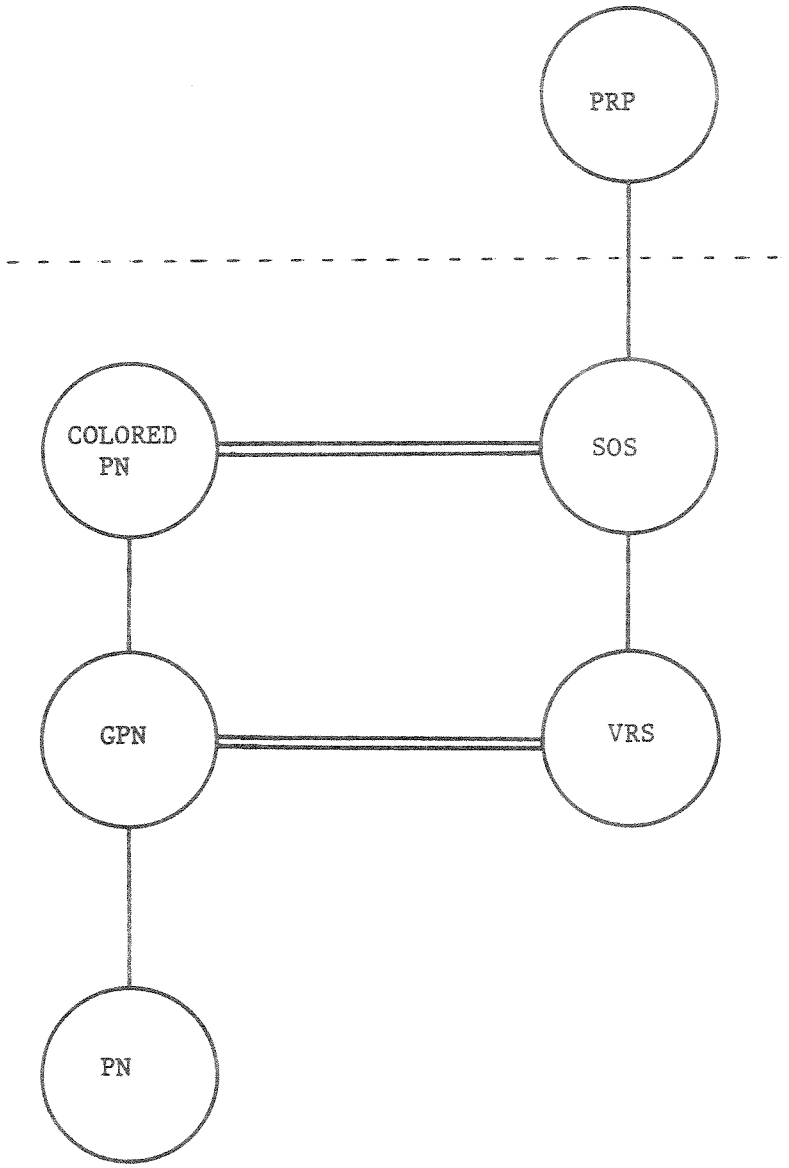


Figure B.2: Model Relationships

REFERENCES

[Abd76] Abd-Alla, A.M. and Moffet, L.H., 'On-line Architectural Tuning Using Microcapture', Proc. 3rd. Ann. Symp. on Comput. Archi., pp. 165-171, 1976

[Ack82] Ackermann, W.B., 'Data Flow Languages', IEEE Computer, vol. 15/2, pp. 15-25, Feb. 1982

[Age81] Agerwala, T.K.M. and Lint, B.J. 'Communication Issues in the Design and Analysis of Parallel Algorithms', IEEE Trans. on Software Engg., vol. SE-7/2, Mar. 1981

[And75] Anderson, G.A. and Jensen, E.D., 'Computer Interconnection Structures: Taxonomy, Characteristics and Example', ACM Computing Surveys, vol. 7/4, pp.197-213, Dec. 1975

[And62] Anderson, J.P. etal., 'D825- A Multiple Computer System for Command and Control', Proc. AFIPS FJCC, vol. 22, pp. 86-96, 1962

[App81] Applewhite, Hugh, personal communication

[Arn76] Arnold, R. and Page, E.W. 'A Hierarchical, Restructurable Multiprocessor Architecture', Proc. 3rd. Ann. Sym. on Comput. Archi., pp. 40-45, Jan. 1976

[Bac78] Backus, John, ACM Turing Award Lecture, 1978

[Bai81] Bain, W.L. and Ahuja, S.R., 'Performance Analysis of High Speed Digital Busses for Multiprocessing Systems', Proc. 8th. Ann. Symp. on Comput. Archi., pp.107-134, May 1981

[Bar68] Barnes G. H., etal, 'The Illiac IV Computer', IEEE Trans on Comput, vol. c/17, pp. 746-757, 1968

[Bat76] Batcher, K.E., 'The Flip Network in Staran', Proc. I'ntl. Conf. on Parallel Proc., pp. 65-71, 1976

[Ber79] Berlin, F.B., 'Time-Extended Petri Nets', MA Thesis, Dept. of Comput. Sci., U.T., Austin, Tx., Aug. 1979

[Bro70] Brown, B.S. etal., 'Sorting in a Paging Environment', Comm. of the ACM vol. 13/8, pp.438-494, Aug. 1970

[Bro79] Browne,J.C. and Tripathi etal. 'A Preliminary Definition of a Job Control Language for TRAC', TR-15, Dept. of Elect. Engg. and Dept. of Comput. Sci. U.T. Austin, Jan. 1979

[Bro81] Browne,J.C. etal., 'A Computation Specification Language', in preparation

[Bur77] Burroughs Corp., 'BSP: Overview, Perspective, Architecture', 1977

[Bus62] Bussel, B., 'Properties of a variable Structure Computer

System in the Solution of Parabolic Partial Differential Equations', PhD Dissertation, Elect. Engg. Dept. UCLA, 1962

[Com67] Comeau, L.W., 'A Study of the Effect of User Program Optimization in a Paging System', Proc. ACM Symp. on Operating Syst. Principles, Gatlinburg, Tenn. 1967

[Den75] Dennis, J.B., 'Packet Communication Architecture', Proc. of the 1975 Sagamore Comput. Conf. on Parallel Proc., pp. 224-229, Aug. 1975

[Den77] Dennis, J.B. etal. 'A Highly Parallel Processor Using a Dataflow Machine Language', Computation Structures Group Memo 134, Lab. for Comput. Sci., M.I.T., Cambridge, Mass., Jan. 1977

[Den80] Dennis, J.B., 'Data Flow Supercomputers', IEEE Computer, vol. 13/11, pp. 48-56, Nov. 1980

[DeG80] DeGroot, R.D. and Malek, M.M., 'Resource Allocation for Macro-pipelines', Proc. of Dist. Data Acquisition, Comput. and Control Symp., pp. 23-27, Dec. 1980.

[DeG81] DeGroot, R.D., 'Mapping Computation Structures to Regular SW Banyans', PhD dissertation, Dept. of Comp. Sci., U.T., Austin, Dec. 1981.

[Ens77] Enslow, P.H. Jr., 'Multiprocessor Organization - A Survey', ACM Computing Surveys, vol. 9/1, pp. 103-129, Mar,

1977

[Est60] Estrin, G., 'Organization of Computer Systems- The Fixed plus Variable Structure Computer', Proc. AFIPS WJCC, vol. 17, pp.33-40, 1960.

[Fed80] Fedak, S., 'On the Implementaion of a Comptation Specification Language', TR, Dept. of Comput. Sci., U.T. Austin, Sept. 1980

[Fer76] Ferrari, D., 'The Improvement of Program Behavior', IEEE Computer, vol. 9/11 pp. 39-47, Nov. 1976

[Gan81] Gannon, D., 'On Mapping Non Uniform P.D.E. Strutures and Algorithms onto Uniform Array Architectures', Proc. 10th. Intl. Conf. on Parallel Proc., pp. 100-105, Aug. 1981

[Gen78] Gentleman, W.M., 'Some Complexity Results for Matrix Computations on Parallel Processors', Journal of the ACM, vol. 25, pp. 112-115, 1978

[Gok73] Goke R. L. and Lipovski G. J., 'A Banyan Network for Partitioning Multiprocessor Systems', 1st Symp on Comput Arch, pp. 21-28, 1973

[Gro80] Grosch, C., 'The Effect of Data Transfer Pattern of an Array Computer on the Efficiency of some Algorithms for the Tridiagonal and Poisson Problem', Array Architectures for Computing in the 80's and the 90's, ICASE Workshop, April 1980,

Hampton, Virginia

[Hab79] Habermann, A.N., 'A Software Development Control System', ACM, GI, ERO, IEEE, 1979

[Han78] Han, Y.W., 'Performance Evaluation of a Digital System Using A Petri Net like Approach', Proc. National Electronics Conf., pp. 166-172, 1978

[Hat71] Hatfield, D.J. and Gerard, J., 'Program Restructuring for Virtual Memory', IBM System Journal, vol. 10-3, pp. 168-192, 1971

[Hel77] Heller D., 'Direct and Iterative methods for Block Tri Diagonal Linear Systems', PhD dissertation, CS Dept, CMU, Pittsburgh, 1977

[Hin72] Hintz R. G. and Tate D. P., 'Control Data STAR-100 Design', 6th Ann IEEE Comput Soc Conf, COMPCON, pp. 1-4, 1972

[Iku81] Ikumi, K., 'Resource Allocation for a Configurable Multi-Resource Varistructure Network Architectured Computer System', TR, Dept. of Comput. Sci., U.T. Austin, May 1981

[Inf81] Information Research Associates, 'CADS: Computer Analysis and Design System', I.R.A. Report, Austin, Tx., May 1981

[Jon79a] Jones, A.K. and Schwans, K., 'TASK forces:

Distributed Software for Solving Problems of Substantial Size', Proc. 4th Soft. Engg. Conf., pp.315-330, Sept. 1979

[Jon79b] Jones, A.K. and Schwans, K., 'The TASK specification Language', Dept. of Comput. Sci., Carnegie Mellon University, 1979

[Kap81] Kapur, R.N. and Browne, J.C., 'Block Tridiagonal System Solution on Reconfigurable Array Computers', Proc. 10th. Intl. Conf. on Parallel Proc., pp. 92-99, Aug. 1981

[Kar78] Kartashev, S.P. and Kartashev, S.I., 'LSI modular Computers, Systems and Networks', Computer, vol. 11/7, pp.7-15, July 1978.

[Kat78] Katsuki, D. etal., 'Pluribus: An Operational Fault-Tolerant Multiprocessor', Proc. IEEE, vol. 66/10, pp. 1146-1159, Oct. 1978

[Law75] Lawrie, D.H., 'Access and Alignment of Data in Array Processors', IEEE Trans. on Comput., vol. C-24, pp. 1145-1155, Dec. 1975

[Lin79] Lint, B.J. 'A Study of Communication Issues in Parallel Algorithms', Phd Dissertation, Dept. of Elect. Engg., U.T. Austin, 1979

[Lip77] Lipovski G. J. and Tripathi A. R., 'A Reconfigurable Varistructured Array Processor', Proc Intl Parallel Proc

Conf,pp. 165-174, 1977

[Lis74] Liskov, B., 'A note on CLU', Computation Structures Group, Memo:112, MIT Project MAC, Nov. 1974

[Lon61] Lonergan, W. and King, P., 'Design of the B 5000 System', Datamation, vol. 7/5, May 1961.

[McK69] McKellar, A.K. and Coffman, E.G. Jr., 'Organizing Matrices and Matrix Operations for Paged Memory Systems', Comm. of the ACM vol. 12/2, pp. 153-164, Mar. 1969

[Mad75] Madsen, N. and Rodrigue, G., 'A Comparison of Direct Methods for Tridiagonal System Solution on the STAR-100', Lawrence Livermore Laboratory, 1975

[Met76] Metcalfe, R.M. and Boggs, D.R., 'Ethernet: Distributed Packet Switching for Local Computer Networks', Comm. of the ACM, vol. 19/7, pp.395-404, Jul., 1976

[Mye80] Myers, W.M and Malek, M.M., 'Figures of Merit for Interconnection Networks', Proc. of the Workshop on Interconnection Networks, pp. 74-83, April 1980.

[Noe71] Noe, Jerre, 'A Petri Net Description of the CDC6400', Proc. ACM Workshop on System Performance Evaluation, harvard University, pp. 362-378, 1971.

[Noe73] Noe, Jerre and Nutt, G.J., 'Macro E-Nets for

Representation of Parallel Systems', IEEE Trans. on Comput., vol. c-22/8, pp. 718-727, Aug. 1973

[Nut77] Nutt, G.J., 'Microprocessor Implementation of a Parallel Computer', Proc. 4th. Symp. on Comput. Archi., pp.147-152, Mar. 1977

[Oka76] Okada, Y. etal., 'A Novel Multiprocessor Array', Proc. 2nd. Symp. on Microarchitecture, Euromicaro, North Holland Publishing Co. pp. 83-90, 1976.

[Ole78] Oleinick, P., 'The Implementation and Evaluation of Parallel Algorithms on the C.mmp', PhD Dissertation, Dept. of Comput. Sci., CMU, Nov. 1978

[Org78] Organick E. and Hinds, J.A., 'Architecture and Programming of the B1700/B1800 Series', North Holland Co., Amsterdam, 1978

[Par72] Parnas, D.L., 'On the Criteria to be used in Decomposing Systems into Modules', Comm. of the ACM 1vol. 15/12, 1972

[Pea77] Pease, M.C., 'The Indirect Binary N-Cube Microprocessor Array', IEEE Trans. on Comput., vol. C/26, pp. 451-473, May 1977

[Por60] Porter, R.E., 'The RW-400: A New Polymarphic Data System', Datamation, vol. 6/1, pp.8-14, Jan./Feb. 1960

[Pre79] Premkumar Uppaluru V., etal, 'Interprocessor Communication in TRAC', 1st Intl Conf on Disti Comput and Systems, pp. 51-62, Aug. 1979

[Ram73] Ramchandani, C., 'Analysis of Asynchronous Concurrent Systems by Timed Petri Nets', Phd Thesis, M.I.T., Sept. 1973

[Ram80] Ramamoorthy, C.V. and Ho, G.S., 'Performance Evaluation of Aysnchronous Concurrent Systems using Petri Nets', IEEE Trans on Software Engg., vol SE-6/5, pp. 440-449, Sept. 1980

[Rau78] Raucher, T.M. and Aggarwala, A.K., 'Dynamic Problem Oriented Redefinition of Computer Architecture via Microprogramming', IEEE Trans. on Comput., vol. C/29, 1006-1014 (1978)

[Rob79] Robinson, J.T., 'Some Analysis Techniques for Asynchronous Multiprocessor Algorithms', IEEE Trans. on Software Engg., vol. SE-5/1, pp. 24-31, Jan. 1979

[Rum77] Rumbaugh, J.E., 'A Data Flow Multiprocessor', IEEE Trans. on Comput.,vol. C-22/2, pp. 138-146, Feb. 1977

[Rus78] Russel, R. B., 'The Cray-1 Computer System', Comm. of the ACM, vol. 21/1, pp. 63-72, Jan. 1978

[Sch78] Schwetman, H.D., 'Hybrid Simulation Models of Computer Systems', Comm. of the ACM, vol. 21/9, Sept. 1978, pp. 718-723

[Sea75] Searle, B.C. and Freberg, D.E., 'Tutorial: Microprocessor Applications in Multiple Processor Systems', IEEE Computer, vol. 8/10, pp. 22-30, Oct. 1975

[Sej80] Sejnowski M. C., etal, 'An Overview of the Texas Reconfigurable Array Computer', AFIPS NCC pp. 631-642, May 1980

[Sej81] Sejnowski, M.C., 'Packet Architecture of TRAC', MA Report, Comput. Sci. Dept., UT, Austin, May 1981.

[Sie79] Siegel H. J., etal, 'PASM' TR.EE-79-40, School of Electrical Engineering, Purdue University, W. Lafayette, IN. 47907, Aug. 1979

[Sif77] Sifakis, J., 'Use of Petri Nets', Measuring, Modelling and Evaluating Comput. Systems, Beilner, H. and Gelenbe, E. Editors, North-Holland Publishing Co., 1977.

[Skl81] Sklansker, M., personal communication.

[Sto71] Stone H. S., 'Parallel Processing with the Perfect Shuffle', IEEE Trans. on Comput., Vol C/20, pp. 153-161, 1971

[Swa77] Swan, R.J., Fuller, S.H. and Siewiorek, D.P., 'Cm*: A Modular Multi-microprocessor Network', Proc. of the AFIPS NCC, pp. 637-644, 1977.

[Tho63] Thompson, R.N. and Wilkinson, J.A., 'The D825 Automatic Operating and Scheduling Program', Proc. AFIPS SJCC, vol. 23,

pp. 41-49, 1963

[Tra76] Traub J. F., etal. 'Accelerated Iterative Methods for the Solution of Tridiagonal Systems on Parallel Computers', Journal of the ACM, vol. 23, 1976, pp. 636-654

[Tri80] Tripathi, A.R. etal. 'AN Overview of Research Directions in Distributed Computing', Proc. of the Compcon, 1980

[TRW60] TRW, Ramo Woolridge Division, 'RW-400, Polymorphic Data Processing System', Canoga Park, Calif., Jan. 1960

[Upp81a] Uppaluru, V.Premkumar, 'A Theoretical Basis for the Analysis and Partitioning of Regular SW Banyans', PhD Dissertation, Dept of Elect. Engg, U.T. Austin, Aug. 1981

[Upp81b] Uppaluru, V.Premkumar, personal commmunication

[Vra73] Vranesic, Z.G. etal., 'Design of a Fully variable Length Structured Minicomputer', Proc. 1st. Ann. Symp. on Comput. Archi., pp. 251-255, 1973

[Wat72] Watson W. J., 'The TI ASC- A Highly Modular and Flexible Supercomputer Architecture!', AFIPS FJCC, Vol 41, pp. 221-228, 1972

[Wu80] Wu, C.-L. and Feng, T.Y., 'On A Class of Multistage Inteconnection Networks', IEEE Trans. on Comput., vol. c/29,

pp. 694-702, Aug. 1980

[Wul72] Wulf W. and Bell C. G., 'C.mmp- A Multi mini processor', AFIPS FJCC, pp. 765-777, Dec. 1972

[Wul76] Wulf, W. etal., 'Abstraction and Verification in Alphard, Introduction to Language and Methodology', Comput. Sci. Dept., CMU, Jun. 1976

[You72] Young, D.M. and Gregory, R.T., 'A Survey of Numerical Mathematics', Vol. I and II, Addison-Wesley Publ. Co., 1972

VITA

Rajan Netralal Kapur was born in Mandi, Himanchal Pradesh, India on January 1, 1953, the son of Shrimati Bhawani Kapur and Shri Netralal Kapur. After completing his work at Loyola High School, Pune, India in 1969 and Fergusson College, Pune, India in 1970 he entered The Indian Institute of Technology in Kanpur, India, where he received the degree of Bachelor of Technology in Electrical Engineering in May 1975. Thereafter he was awarded the degree of Master of Electrical Engineering from Rice University, Houston, Texas in May 1978. He joined the Department of Electrical Engineering at The University of Texas in Austin in May 1977 where he worked at different times as a teaching assistant, a research assistant and as a research engineer.

Permanent Address: c/o Shri Netralal Kapur,
D-2/33, N.D.A, Khadakvasla,
Pune, 411023, INDIA.

This dissertation was prepared with the aid of TECO and RUNOFF on a DEC computer by Rajan Kapur.