Modeling and Simulation of Parallelism by Colored Petri Nets

[1]Homayun Motameni, [1]Zohre Ramezani and [2]Zahra Usefi

[1]Department Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran
[2]Rouzbahan Institute of Higher Education, Sari, Iran

Abstract: Coloured Petri Nets (CPNs) is a language for the modelling and validation of systems in which concurrency, parallelism, communication and synchronisation play a major role. In this paper the prefix sums of a set of values is a example to show how we can simulate and model parallel and concurrent systems with colored petri nets.

Key words: Parallel · PRAM · Prefix sums · Petri net · CPN · Parallelism

INTRODUCTION

Coloured Petri Nets (CPNs) is a language for the modelling and validation of systems in which concurrency, communication and synchronisation play a major role. Coloured Petri Nets is a discrete-event modelling language combining Petri nets with the functional programming language Standard ML. Petri nets provide the foundation of the graphical notation and the basic primitives for modelling concurrency, communication, and synchronisation. Standard ML provides the primitives for the definition of data types, describing data manipulation and for creating compact and parameterisable models [1].

Furthermore, synchronous and asynchronous events present their prioritized relations and structural adaptive effects. The main difference between CPNs and Petri Nets (PN)is that in CPNs the elements are separable but in PNs they are not. Colored indicates the elements specific feature. The relation between CPNs and ordinary PNs is analogous to high level programming languages to an assembly code (Low level programming language). Theoretically, CPNs have precise computational power but practically since high level programming languages have better structural specifications, they have greater modeling power.

CPN's drawback is its non adaptivity [1, 2] therefore it is not possible to access the previous information available in CPNs. If there is more than one transition activated then each transition can be considered as thenext shot. This Colored Petri Net's characteristic indicates that since several events occur concurrently and event incidences are not similar, then when events occur they do not change by time and this phenomenon is in contrast with the real and dynamic world. Simulation would be similar to execution of the main program. Our Purpose is to use the simulated model for analyzing the performance of the systems, as a result here the system problems and the weak points would be identified.

However, classic CPN tools can do nothing to improve and solve problems and also it would not be possible to predict the next optimized situation [10].

Systems engineering is a comprehensive discipline involving a multitude of activities such as requirements engineering, design and specification, implementation, testing and deployment. The development of distributedsystems is particularly challenging. A major reason is that these systems possess concurrency and non-determinism which means that the execution of such systems may proceed in many different ways.

Parallel processing means using variety techniques in contemporary processing of data for increase meant of speed of calcuting computer systems [9].

A PRAM consists of a control unit, global memory and an unbounded set of processors, each with its own private memory. Although active processors execute identical instructions, every processor has a unique index and the value of a processor's index can be used to enable or disable the processor or influence which memory location it accesses.

A PRAM computation begins with the input stord in global memory and a signal active processing element. During each step of the computation an active, enabled processor may read a value from a signal private or global

---

Corresponding Author: Homayun Motameni, Department Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran

memory location, perform a signal RAM operation and write into one local or global memory location. alternatively, during a computation step a processor may active another processor. All active, enabled processors must execute the same instruction, albeit on different memory locations. The computation terminates when the last processor halts.

Various PRAM models differ in how they handle read or write conflicts; i.e., when two or more processors attempt to read from, or write to, the same global memory location. Most of the results in the research literature have been based upon one of the following models:

EREW: Read or write conflicts are not allowed, CREW: Concurrent reading allowed and CRCW: Concurrent reading and concurrent writing allowed.

In this Paper an example of a parallelism is showed for prefix sums. The rest of the paper is structured as follows. In section Colored Petri Nets, we present CPN as a basic strategy used in our proposed method. in section Prefix sums, the considered problem, which is the same prefix sums of a set of values, will be described. The proposed algorithm is introduced in last Section and in the end we have concluding remarks.

Colored Petri Nets: A Petri net (also known as a place/transition net or P/T net) is one of several mathematical modeling languages for the description of distributed systems. A Petri net is a directed bipartite graph, in which the nodes represent transitions (i.e. events that may occur, signified by bars) and places (i.e. conditions, signified by circles) [4]. The CPN model is an executive model system that shows position of a system and incidents which cause change in position of system.

The CPN modelling language is a general purpose modeling language, i.e., it is not focused on modelling a specific class of systems, but aimed towards a very broad class of systems that can be characterised as concurrent systems [9].

CPN Tools is also a suitable tool for editing, modeling, analysis of space of position and analysis of operation of CPN models. Graph of Petri net is a method for showing of structure of Petri nets that two shapes are in them. These places and transitions connect to each other by arcs. When an arc connects from a transition to a place, it means, the place will be the exit of the mentioned transition and if an arc be drawn from a place to a transition, it means, that place will be entrance of the mentioned transition. For description of Petri net action, tokens add to this graph, too. it causes, concept of position be defined in this graph. Number of these tokens

in the whole graph and manner of their distribution among places, determines the position of Petri net, which called it a petri net marking [10].

A formal definition of CPN is as follows: [1, 2, 4] A Colored PN (CPN) is a 6-tuple CPN $(P,T,C, I^-, I+, M)$ where:

- $P = \{p1, p2,..., pn\}$ denotes a finite and non-empty set of places,
- $T = \{t1, t2,..., tm\}$ denotes a finite and non-empty set of transitions, $P \bullet T = \emptyset$,
- C is a colour function that assigns a finite and non-empty set of colors to each place and a finite and non-empty set of modes to each transition.
- I^- and I denote the backward and forward incidence functions defined by $P \times T$, such that $I^-(p,t)$, $I(p,t) \bullet [C(t) \bullet C(p)MS], v(p,t) \bullet P \times T^2$.
- Mo denotes a function defined on P, describing the initial marking such that $Mo(p) \bullet C(p)MS$.

Prefix Sums: Modeling parallel computations is more complicated than modeling sequential computations because in practice parallel computers tend to vary more in organization than do sequential computers.

A multiprocessor model is a generalization of the sequential RAM model in which there is more than one processor. Multiprocessor models can be classified into three basic types: local memory machine models, modular memory machine models and parallel random-access machine (PRAM) models.

In a PRAM model, a processor can access any word of memory in a single step. Furthermore, these accesses can occur in parallel, i.e., in a single step, every processor can access the shared memory.

The topology of a network has a large influence on the performance and also on the cost and difficulty of constructing the network.

The simplest network topology is a bus. This network can be used in both local memory machine models and modular memory machine models. In either case, all processors and memory modules are typically connected to a single bus. In each step, at most one piece of data can be written onto the bus. This data might be a request from a processor to read or write a memory value, or it might be the response from the processor or memory module that holds the value. In practice, the advantages of using a bus is that it is simple to build and, because all processors and memory modules can observe the traffic on the bus, it is relatively easy to develop protocols that allow processors to cache memory values locally [11].
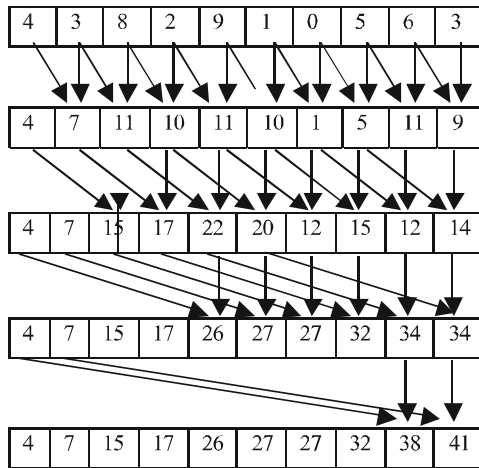
4	3	8	2	9	1	0	5	6	3

4	7	11	10	11	10	1	5	11	9

4	7	15	17	22	20	12	15	12	14

4	7	15	17	26	27	27	32	34	34

4	7	15	17	26	27	27	32	38	41

Fig. 1: Prefix sums of a list of n values.

Fig. 2: Definition of variables in CPN Tools.

```
▼ Declarations
  ▼ Standard declarations
    ▼ colset UNIT = unit;
    ▼ colset INT = int;
    ▼ colset BOOL = bool;
    ▼ colset STRING = string;
    ▼ var a,b:INT;
    ▼ var c:BOOL;
```

In Prefix Sums of a list of numbers in PRAM, Given a set of n values a1, a2,..., an and an associative operation •, the prefix sums problem is to compute the n question:

a1
a1 • a2
a1 • a2 • a3
...
a1 • a2 • a3 •... • an

for example, given the operation + and the array of integers [3, 1, 0, 4, 2], the prefix sums of the array are [3, 4, 4, 8, 10].

Prefix sum are also called parallel prefixes and scans. Prefix sums have many users. For instance, suppose we are given an array A of n letters.we want to pack the uppercase letters in the initial portion of A while maintaining their order. All prefix sums of a list of n values can be computed in [log n] addition steps on an EREW PRAM (Fig. 1).

The Algorithm Simulation: In this section we propose an algorithm Simulation for Prefix Sums with using of Colored Petri Net and CPN Tools that has explaned before.
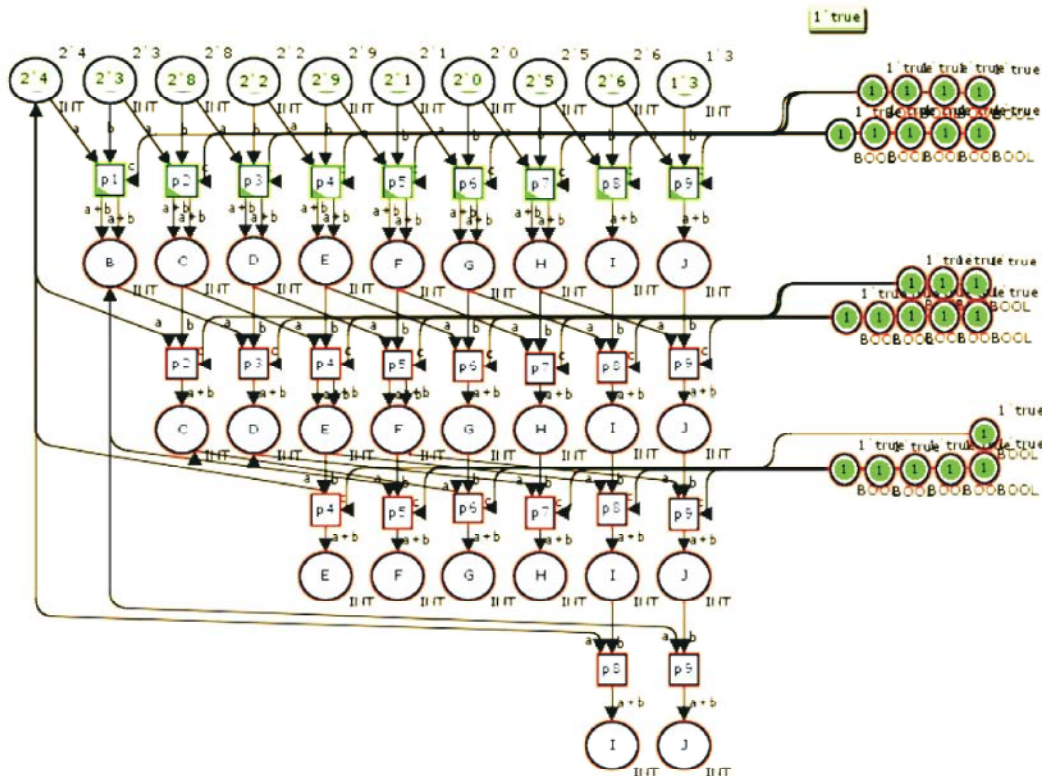

Fig. 3: General view of Proposed Model Simulation in CPN Tools

CONCLUSION

Parallel processing, in the literal sense of the term, is used in virtually every modern computer.

With using CPN Tools which is a powerful tool for processing modeling and provides more description power regard to staff nets, we can models parallelism.

REFERENCES

1. Jensen, K., L. Michael Kristensen and L. Wells, 2007. Colored Petri Nets and CPN Tools for modeling and validation of concurrent systems. International validation of concurrent systems. International, pp: 213-254.
2. Jensen, K., 1997. Colored Petri Nets-Basic Concepts, Analysis Methods and Practical Use. Springer-Verlag, pp: 1-3.
3. Beaudouin-Lafon, M., W.E. Mackay and M. Jensen, 2001. CPN Tools: A Tool for Editing and Simulating 2001. CPN Tools: A Tool for Editing and Simulating for the Construction and Analysis of Systems, pp: 574-580.
4. Peterson, J., 1981. Petri Net Theory and the Modelling of Systems, Prentice Hall.
5. Harper, R., 2000. Programming in M.L. Standard, Carnegie Mellon University.
6. Turon, A., 2006. SML/NJ Language Processing Tools: User Guide.
7. Hierons, R.M., T.H. Kim and H. Ural, 2002. "Expanding an Extended Finite State Machine to Aid Testability", In Proceedings of the 26th Annual International Computer Software and Applications Conference(COMPSACp02), pp: 1-6.
8. Elkoutbi, M. and R.K. Keller, 1998. Modeling Interactive Systems with Hierarchical Colored Petri Nets. Advanced Simulation Technologies Conf., Boston, MA, pp: 432-437.
9. Kurt Jensen, Lars Michael Kristensen and Lisa Wells, 0000. "Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems", Department of Computer Science University of Aarhus IT-Parken, Aabogade 34, DK- 8200 Aarhus N, DENMARK.
10. Behnam Barzegar, Homayun Motameni, Zohre Ramezani and Zahra Yousefi, 0000. "Modeling and Simulation of Pipeline by Colored Petri Nets".
11. Guy E. Blelloch and Bruce M. Maggs, 0000. " Parallel Algorithms", Carnegie Mellon University.