Petri net models of metastable operations in latch circuits

F. Xia[*†], I.G. Clark[‡], A.V. Yakovlev[*†] and A.C. Davies[‡]

1. Abstract

Data communications between concurrent processes often employ shared latch circuitry of some kind, the most basic being a simple flip-flop which is written by one process and read by another, providing an inter-process assignment operation of a single bit binary variable. When the processes concerned do not operate in a synchronised fashion, metastable transients are possible inside shared latches. A method of deriving discrete Petri net models for such latches, covering possible metastable behaviour, is proposed. Both the local onset of metastability and the effects of metastable input signals are considered in the representation.

2. Introduction

The use of fully asynchronous processes is advantageous in many hard real-time distributed computer systems. For instance, the complete elimination of time interference in data communications between concurrent processes makes it possible to accurately predict the temporal progress of each process in the system because the timing of each one is completely independent. In certain safety critical systems it may also be required that a process cannot be temporally connected to any other processes and must progress at its own pace, even though this may make data communications with other processes a more complicated problem.

Various protocols and mechanisms have been proposed to make it possible for two asynchronous processes to communicate with each other. The so-called “slot” type mechanisms described in [1-4] are devised to accommodate the need for data communications between processes in the absence of any synchronisation. They employ multiple data storage slots, any one of which may be synchronised to either process at any time, but not simultaneously to both processes. The use of bit “control” variables makes the communication system globally asynchronous and locally synchronous [5]. In effect, the slot mechanisms realise “regular” and “atomic” registers in the data path between asynchronous concurrent processes with “safe” registers [6, 7] to convey the values of control variables. This reduces the overall adverse effect of possible metastable operations in the data communication to a minimum, as the smallest data item that can be transmitted from one concurrent process to another is a bit.

The transmission of a bit variable between two asynchronous processes may be implemented via a binary latch circuit. One example of this kind of system can be found in Figure 1. Here the complement of the value of variable *x* in Process 1 is passed to that of variable *y* in Process 2 when the clock/control signal *cl* occurs. Therefore this is an

[*] Department of Computing Science, University of Newcastle.
[†] Supported by EPSRC grant GR/K70715 (project HADES).
[‡] Department of Electronic Engineering, King’s College, University of London.

inter-process bit type variable assignment statement, *cl*: *y* := NOT *x*. Depending on the implementation of *cl*, this action may or may not imply any temporal co-ordination between *cl* and *x*.

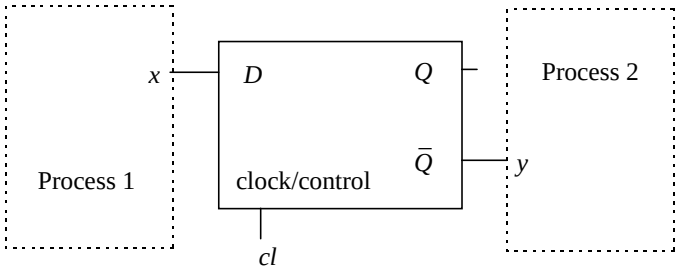

Figure 1 Concurrent processes communicating via a latch.

The latch device can be of the D-type, which is activated by either the rising or falling edge of the clock signal *cl* [8], whose traditional Petri net [9] model is shown in Figure 2, or the transparent type, which is activated by the level of the control signal *cl* [8, 10]

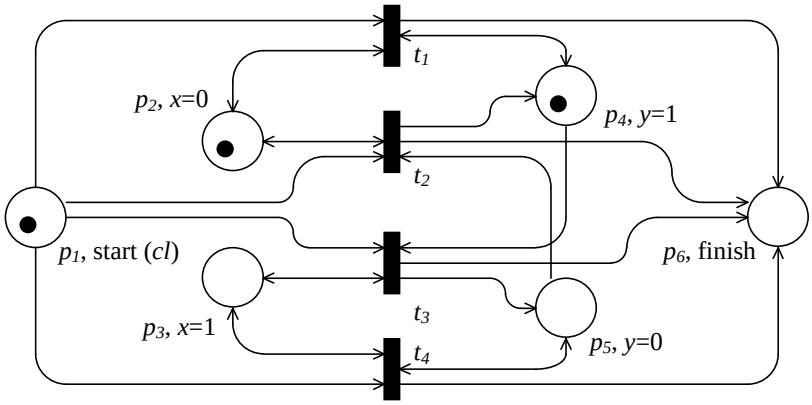

Figure 2 Model of D-type latch.

Fundamentally, however, no asynchronous communication scheme can avoid the possibility of metastability in the basic bit register or latch. Much study has been carried out and reported in the literature of metastability in such devices as arbiters and latches. Most of the studies have involved analyses in the analogue domain, i.e. treating the system as continuous both in time and signal levels. The general consensus among these studies is that in a fully asynchronous environment metastable operations in such circuits are not completely avoidable [11-13]. Furthermore, existing analyses of metastable behaviour usually consider a stand-alone device, such as a flip-flop, a synchroniser or an arbiter. This makes the analysis practical using analogue techniques without the model being too complicated [14]. However, examples found in [1-4] suggest that devices which are prone to metastability can be connected and influence one another. Before going into detailed system-wide analysis in the analogue domain it may be advantageous to obtain a discrete-level model which reflects some of the properties and effects of metastability. We believe that the Petri net based model proposed here could act as such an intermediate facility.

In this report we propose a discrete description and analysis of the problem of metastability. A method of establishing discrete, Petri net based models of latches

operating in an asynchronous environment is presented. Considerations will be given to such aspects of the problem as the atomicity of models and the representation of D-type and transparent latches. Both basic place/transition (P/T) nets [9] and coloured Petri nets [15] are used to provide some generality.

3. Discrete state model of metastability for a digital signal

Conventionally, the state space of a binary digital signal is considered to be {0, 1}, i.e. the signal may take the discrete values of 0 and 1. This is consistent with the construction of correct binary digital circuits and systems where every signal has at most two possible stable levels, corresponding to these two numerical values.

Generally speaking, metastability is a state wherein a signal in a bistable system stays at an intermediate level between logic 1 and logic 0 for an indeterminate period of time and appears to be stabilised at this level. This phenomenon is schematically shown in Figure 3.

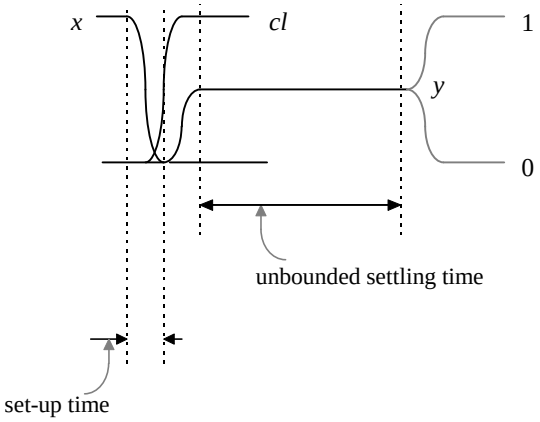

Figure 3 Metastable transient at signal *y*.

In digital circuits metastability is usually the result of unstable equilibria introduced by conflicting demands of input signals and current states. Assuming that the system in question is Newtonian, i.e. disregarding such issues as the arguments of quantum mechanics and the counting of electrons and holes in semiconductors, such a state may persist for an unlimited period of time [14].

Clearly, a discrete model of a digital system including only the values of 0 and 1 in its state space for every signal, such as the Petri net shown in Figure 2, cannot represent the metastable state. On the other hand, an analogue model, though it certainly can be made comprehensive enough to include all possible states, may present problems at analysis if the system is complicated. Since in any dynamic system, the only sustainable states (i.e. those capable of persisting autonomously for indeterminate periods of time) are those corresponding to the equilibria in the system, it would be useful to establish discrete models in which only these discrete states are fully represented. In the case of digital circuits, this requires the proper representation of the logic values 0 and 1 and the metastable state for any signal where metastable operations are possible.

The basic modelling technique shown in Figure 4 is therefore proposed.

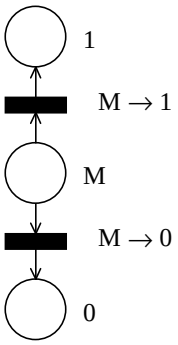

Figure 4 Discrete model of metastability (technique A).

In this Petri net model, the M place allows the representation of a metastable signal. It also represents the fact that while at a metastable state, a variable may be referenced in value by other parts of the system. The included transitions are the ones representing the natural, non-externally influenced state changes where a signal may settle out of the metastable state due to the fundamental instability of the metastable equilibria.

4. Atomicity

Conventional Petri net models of an assignment statement in the form of Figure 1, such as the one given in Figure 2, associate the action with a single transition, and so represent the statement as atomic, or instantaneous. In reality, latch circuits have non-zero set up time and propagation delay [8]. The effect of these phenomena is schematically shown in Figure 5.

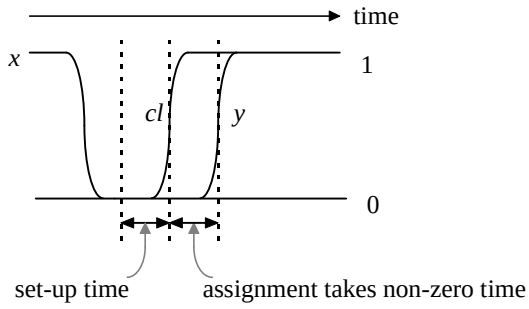

Figure 5 The non-atomicity of latch operations.

Considering the actual time progression of the inter-process assignment statement and similar latch operations, atomic models of the type shown in Figure 2 may not represent all the possible states of a system [16]. One example is shown in Figure 6.

Assuming that both statements 1 and 2 in Figure 6 are implemented with latch circuitry of the type shown in Figure 1 and the statements are truly concurrent without synchronisation, the conventional atomic model in Figure 2 would be unable to represent the possibility of *cross-assignment*, i.e. *x* ending up with the initial value of *y* and vice versa. This is because in conventional Petri nets, which follow interleaving semantics, transitions are fired one at a time and never simultaneously. Other types of semantics, such as step sequences [17], do allow the simultaneous firing of transitions,

but do not address the fundamental issue of atomicity. They also require additional modifications to the analytical techniques of Petri nets.

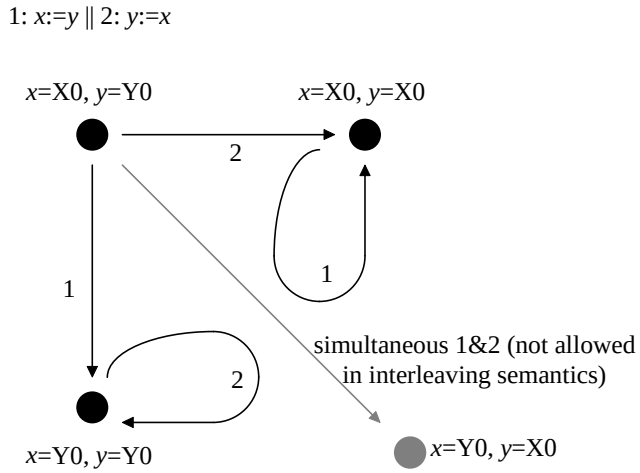

Figure 6 Value exchange cannot be represented by atomic model.

In addition, when analysing concurrent systems, it is often important to represent such statements as individual states which can be interleaved by other processes in the system [18]. This is not represented adequately by the conventional model in Figure 2 either as a single transition firing cannot be interleaved by other transitions firing.

As a result, we propose the modelling technique shown in Figure 7 which represents the beginning and end of a statement with transitions and the process itself with a place, giving an additional model state indicating “statement in progress”. This provides for the representation of the possible interleaving of processes across concurrent systems. It also addresses the issue of atomicity while staying with interleaving semantics.

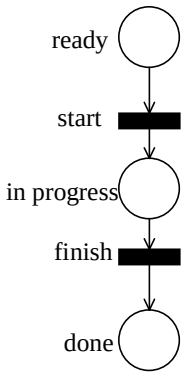

Figure 7 Model of the time-progression of a process (technique B).

Figure 8 shows that if statement 1 in Figure 6 is modelled employing the technique given in Figure 7, all possible states resulting from the concurrent assignment statements would be represented. This effectively allows statement 1 to be interleaved by statement 2 in all possible asynchronous fashions. Therefore, by applying the technique of Figure 7 wherever appropriate it is possible to represent all interleaving patterns among the processes of a concurrent and asynchronous system to an arbitrary resolution.

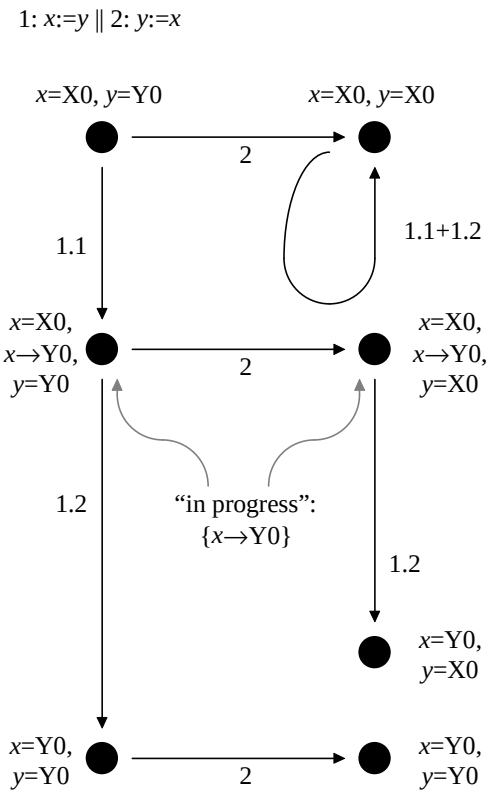

Figure 8 Modelling technique of Figure 7 helps to cover all states.

5. The D-type latch

A Petri net model of the assignment operation of Figure 1 using the D-type latch is shown in Figure 9. The output of a D-type latch follows its input when either a rising or falling edge occurs at signal *cl*. This model incorporates both techniques A and B introduced above in Figure 4 and Figure 7.

The state of "statement in progress", i.e. having started but not properly finished, is represented by the three places in the middle of the graph. Any one of them being marked indicates that signal *cl* is about to send an active edge to the latch. This recognises the non-zero set up time of the device. In one round of action only one of the transitions in the left-most column and one of the transitions in the right-most column would fire. The former signifies the start of the statement and the latter its conclusion.

Input and output signals are both recognised as being capable of metastable levels and therefore each is represented by three states. The "statement in progress" states also include a place representing "a metastable state being copied". Both the local onset and possible propagation of metastable states are represented in this model. The local onset of metastability can happen when the input signal changes during the set up time before the clock signal arrives.

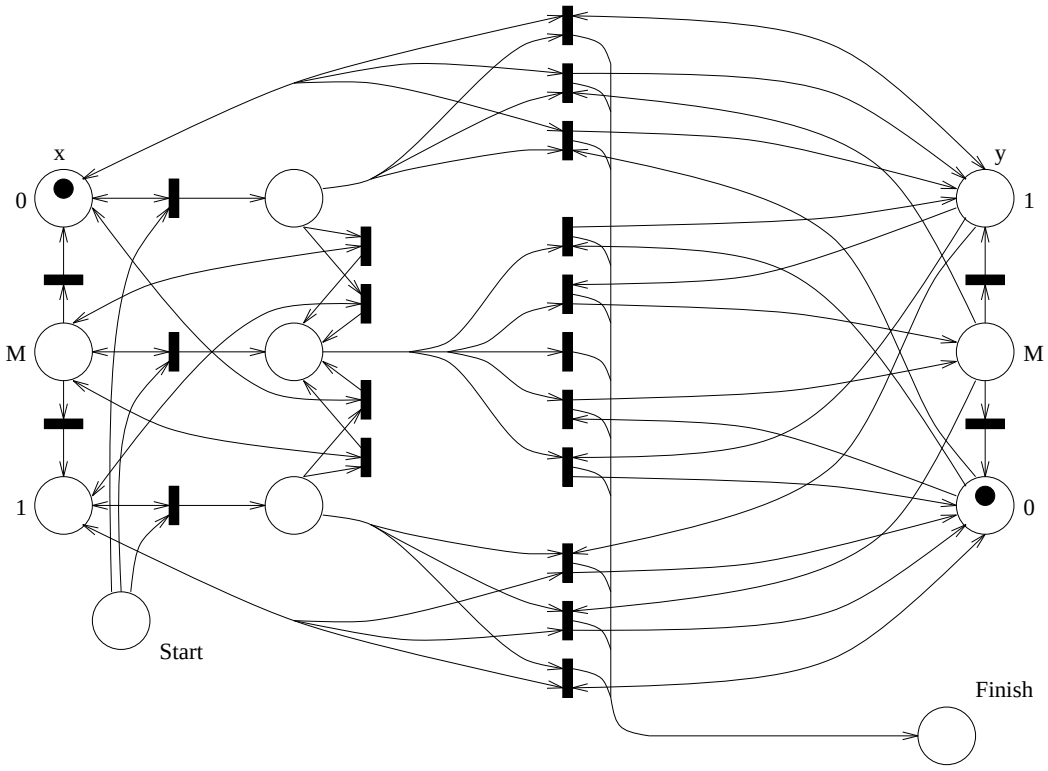

Figure 9 Non-atomic model with metastable states for the D-type latch.

The statement starting and ending places demonstrate the appropriate modelling technique when the overall process that includes the assignment statement is assumed to be sequential. Because of this assumption, the statements start off each other in an overall sequence and there is no need to maintain an internal pipeline control within the model of each statement.

The new model is more complex than the conventional one in Figure 2 and has the potential of causing significant increases in system model size. Therefore such models must be used with care [16]. On the other hand, there is usually no need to apply this type of model throughout the system. For instance, as shown in Figure 6 and Figure 8, in order to avoid missing the cross assignment state, technique B needs to be applied to the model of only one of the statements, not both.

6. The transparent latch

Similar to the D-type latch, transparent latches may also be used at the basic level to pass data between concurrent processes. A transparent latch has two operating modes. One is the enabled (transparent) mode in which the output follows the input. The other is the disabled (latched) mode in which the output is held at the last value before disabling happened. For transparent latches, the equivalent to the clock signal in D-type devices is a control signal acting on levels rather than on edges. The latch could be transparent when the control signal is high and latched when it is low, or vice versa [8, 10].

From Figure 9 it is obvious that complete latch models tend to be rather complicated to present in place/transition net [9] form. The model of the transparent latch, therefore, is here presented as a coloured Petri net [15] in Figure 10.

Similarly to the model of the D-type latch in Figure 9, the model for the transparent latch also employs technique B and includes two layers of main transitions in one action of the statement. The first (t_1) represents the start of the statement and the second (t_2) its conclusion. The middle state between the firings of these two layers of transitions represents the situation of "statement in progress". In the coloured Petri net model, the layers of transitions are simplified into single ones.

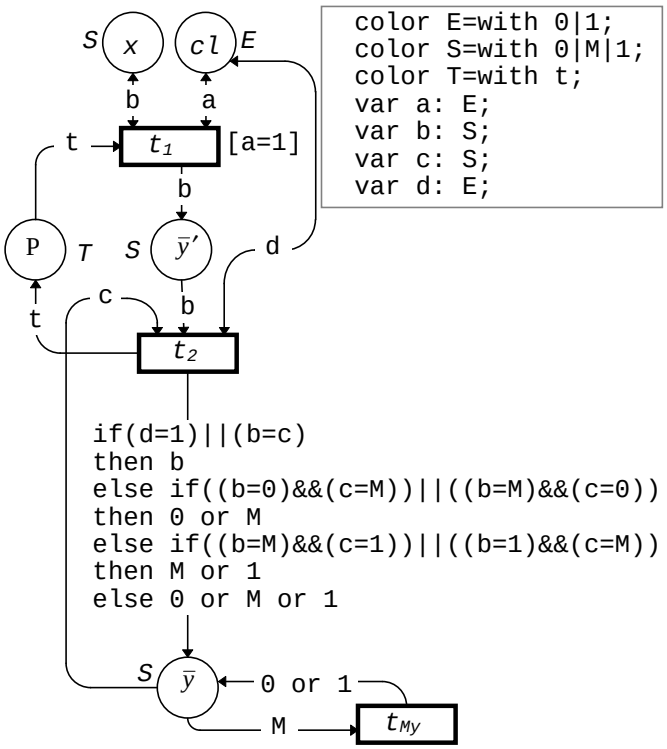

Figure 10 Model of a transparent latch.

In this model the colours and presence of tokens in the places are used to represent the states of the latch. Place *cl* signifies the value of the control variable which determines whether the latch is enabled or disabled. Here it is assumed that the latch is enabled if the latch control signal *cl* = 1 (The token in place *c* has the colour of 1) and disabled otherwise. The variable *cl* can take only the values of 0 and 1 as it is assumed that the externally supplied latch control variable always takes digital values and cannot be metastable. This is similar to assuming correct clock signals with D-type flip-flops. Any misbehaviour of the latch due to control variable errors should not be attributed to the latch itself.

Employing technique A given above, the colour set for places representing signals that can go metastable includes the colour M. The enabling colours and output logic of transition t_1 are derived from the truth table given below.

cl	*x*	P	action
1	0	t	$\bar{y}'$:= 0
1	M	t	$\bar{y}'$:= M
1	1	t	$\bar{y}'$:= 1
0	any value	any value	not enabled
any value	any value	no token	not enabled

Table 1 Truth table for transition t_1.

The actions in Table 1 can be described as follows:

- not enabled: $\bar{y}'$ keeps its current value (transition does not *occur* or fire);
- $\bar{y}'$:= 0: new $\bar{y}'$ = 0;
- $\bar{y}'$:= 1: new $\bar{y}'$ = 1;
- $\bar{y}'$:= M: new $\bar{y}'$ = M;
- no modification to the input variables *cl* and *x* if t_1 fires;
- a token is taken away from P if t_1 fires.

The enabling colours and output logic of transition t_2 are derived from the truth table given in Table 2.

cl	$\bar{y}'$	*x*	action
1	any value	any value	$\bar{y}$:= $\bar{y}'$
0	any value	same as $\bar{y}'$	$\bar{y}$:= $\bar{y}'$ (no change)
0	0	M	$\bar{y}$:= 0 or M
0	0	1	$\bar{y}$:= 0, M or 1
0	M	0	$\bar{y}$:= 0 or M
0	M	1	$\bar{y}$:= M or 1
0	1	0	$\bar{y}$:= 0, M or 1
0	1	M	$\bar{y}$:= M or 1

Table 2 Truth table for transition t_2.

The actions in Table 2 can be described as follows:

- $\bar{y}$:= $\bar{y}'$: new $\bar{y}$ has the value of current $\bar{y}'$, regardless of current value of $\bar{y}$.
- $\bar{y}$:= 0 or M: new $\bar{y}$ may have a value of either 0 or M, with unspecified probability for each possibility.
- $\bar{y}$:= M or 1: new $\bar{y}$ may have a value of either M or 1, with unspecified probability for each possibility.
- $\bar{y}$:= 0, M or 1: new $\bar{y}$ may have a value of 0, M or 1, with unspecified probability for each possibility.
- a firing of t_2 drains the token out of place $\bar{y}'$, but does not disturb the token in place *cl*.

In the transparent latch model a local pipeline control element (place P) is necessary as during the transparent phase, the output of the latch follows the input. This makes it difficult and probably unsafe to rely on an overall sequence control of the processes to manage the internal local pipeline, such as in the case of the D-type latch model above. The local pipeline assumes that the model cannot "process" more than one change of input while latch switching is in progress. In addition, in this model no statement starting and ending places are included. This demonstrates the correct technique when no assumption is made on the sequential nature of the overall process that includes the assignment statement.

7. Conclusion and future work

Non-atomic, discrete models of D-type and transparent latches that include explicitly representations of the metastable state in input and output signals have been developed. These models are part of on-going studies on the behaviour of asynchronous communication protocols and mechanisms being carried out at both the King's College London and the University of Newcastle. More detailed descriptions and the reasoning behind their derivations can be found in [18] and are to be reported in [16].

The techniques A and B presented in this report provide finer resolution than conventional binary discrete models of digital systems but the resulting models remain discrete. The representation of the metastable state and the less stringent assumption of atomicity provide the capability of representing vital characteristics without needing the complexity of full analogue modelling. The modelling techniques are compared in the table below.

Modelling	Time resolution	Level resolution
Analogue	infinite*	infinite*
Conventional digital	0	2
Step sequences	0 but allow simultaneous firing	2
M level (A)	0	3
Process non-atomic (B)	1	2
Combined A+B	1	3

Table 3 Resolutions of modelling techniques.

8. References

1 Simpson, H.R., "Fully-asynchronous communication", IEE Colloquium on MASCOT in real-time systems, May 1987.

2 Simpson, H.R., "Four-slot fully asynchronous communication mechanism", IEE Proceedings, Vol. 137, Pt. E, No. 1, pp.17-30, January 1990.

3 Simpson, H.R., "Correctness analysis of class of asynchronous communication mechanisms", IEE Proceedings, Vol. 139, Pt. E, No. 1, pp.35-49, January 1992.

* Limited only by computational resources such as digital mechanism and simulation time.

4 Simpson, H.R., “New algorithms for asynchronous communication” and “Role model analysis of an asynchronous communication mechanism”, IEE Proceedings on Computing and Digital Techniques, Vol. 144, No. 4, pp.227-240, July 1997.
5 Xia, F, Clark, I.G. and Davies, A.C., “Petri-net based investigation of synchronisation-free interprocess communication in shared-memory real-time systems”, The Second UK Asynchronous Forum, University of Newcastle upon Tyne, Newcastle upon Tyne, UK, July 1-2, 1997.
6 Lamport L., “On interprocess communication, Part I: Basic formalism”, Distributed Computing, Vol. 1986, No.1, pp.77-85, Springer-Verlag, 1986.
7 Lamport L., “On interprocess communication, Part II: Algorithms”, Distributed Computing, Vol. 1986, No.1, pp.86-101, Springer-Verlag, 1986.
8 Horowitz, Paul and Hill, Winfield, The art of electronics, 2nd ed., Cambridge University Press, Cambridge and New York, 1989.
9 Peterson, J.L., Petri net theory and the modeling of systems, Prentice-Hall, 1981.
10 Cortadella, J., Lavagno, L., Vanbekbergen, P. and Yakovlev, A., Designing asynchronous circuits from behavioural specifications with internal conflicts, Proceedings, International symposium on advanced research in asynchonous circuits and systems, pp. 106-115, Salt Lake City, Utah, USA, November 3-5, 1994.
11 Kleeman, L. and Cantoni, A., “On the unavoidability of metastable behaviour in digital systems”, IEEE Trans. Comput, Vol. 36, No. 1, pp.109-112, January 1987.
12 Marino L.R. “General theory of metastable operation”, IEEE Trans. Comput., Vol. 30, No. 2, pp.107-115, 1981.
13 Mendler, M. and Stroup, T., “Newtonian arbiters cannot be proven correct”, Formal Methods in System Design, 1993(3):233-257, December 1993.
14 Xia, F. and Yakovlev. A, “Overview of modelling and analysis techniques for arbiters and related circuits”, Technical Report, Computing Science, University of Newcastle upon Tyne, January 1998.
15 Jensen, K., Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts. Monographs in Theoretical Computer Science, Springer-Verlag, 1992.
16 Clark, I.G., PhD thesis, King’s College, University of London, in preparation.
17 Janicki, R. and Koutny, M., “On equivalent execution semantics of concurrent systems”, In LNCS, Vol. 266, Springer-Verlag, 1987.
18 Xia, F., Combining MASCOT with Petri Nets — A New Aspect of the Modelling, Analysis and Design of Real-Time Systems, PhD degree thesis, King’s College, University of London, submitted Nov. 1997.