Petri net models of metastable operations in latch circuits

F. Xia*, I.G. Clark†, A.V. Yakovlev*, A.C. Davies†

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 models for such latches, covering possible metastable behaviour, using Petri nets, is proposed. Both the local onset of metastability and the effects of metastable input signals are considered in the representation.

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] and [2] 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 [3]. In effect, the slot mechanisms realise "regular" and "atomic" registers in the data path between asynchronous concurrent processes [4, 5] with basic "safe" registers to convey the values of control variables. This reduces the overall adverse effect of possible metastable operations in the data communication to a minimum.

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 been analysis 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 [6, 7, 8]. Furthermore, existing analysis of metastable behaviour usually consider a stand-alone device, such as a flip-flop, a synchroniser or an arbiter. Thus the complexity issue is unlikely to be critical. However, examples found in [1] and [2] suggest that devices which are prone to metastability can be connected and influence one another. Before going into detailed system-wide analysis in the continuous 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 paper 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 P/T nets and coloured Petri nets are used to provide some generality.

Discrete state model of metastability for a digital signal

Traditionally, 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

* Department of Computing Science, University of Newcastle.
† Department of Electronic Engineering, King's College, University of London.

 Contact authors: Fei.Xia@ncl.ac.uk and Ian.Clark@kcl.ac.uk

and systems where every signal has at most two possible stable equilibria, 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. In digital circuits it 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.

Clearly, a discrete model of a digital system including only the values of 0 and 1 in its state space for every signal does not provide for the representation of the metastable state. On the other hand, an analogue model, though 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 1 is therefore proposed.

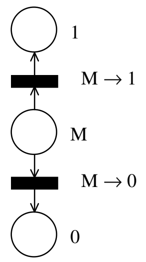

Figure 1 Discrete model of metastability.

In this Petri net model, the M place recognises the possibility of variables to become metastable. 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.

The D-type latch

The smallest data item that can be transmitted from one concurrent process to another is a binary bit. This kind of transmission between two asynchronous processes may be implemented via a binary latch circuit. One example of this kind of system can be found in Figure 2. 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 signal cl is produced. Therefore this is an inter-process bit type variable assignment statement, cl: y := NOT x. Depending on where the clock signal cl comes from and how it is implemented, this action may not require synchronisation between the two processes.

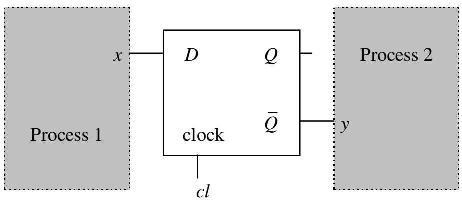

Figure 2 Concurrent processes communicating via a latch.

The latch device can be of the D-type, which responds to either the rising or falling edge of the clock signal [9]. The traditional Petri net model of this circuit is shown in Figure 3.

There are two crucial assumptions behind this model. The first is that the variables involved are strictly binary, without the possibility of staying at a middle value. The other is that the assignment is instantaneous, or atomic, as it is associated with the firing of a single Petri net transition. Neither of these assumptions holds all the time in an asynchronous environment. The required relative timing of the clock signal *cl* and the input signal *x* cannot be ensured if *cl* is not controlled by Process 1. This means that metastability may occur in the output signal *y*. On the other hand, for a model of the latch to be complete, no assumption may be made about the relative timing or execution speed of the processes to which it is connected. Therefore the input signal *x* may also become metastable. Furthermore, in an asynchronous concurrent system, it is important to recognise the possibility of any action, such as the assignment operation considered here, being interleaved in time by other processes elsewhere. As a result an accurate model should treat the assignment process as not atomic but requiring finite time.

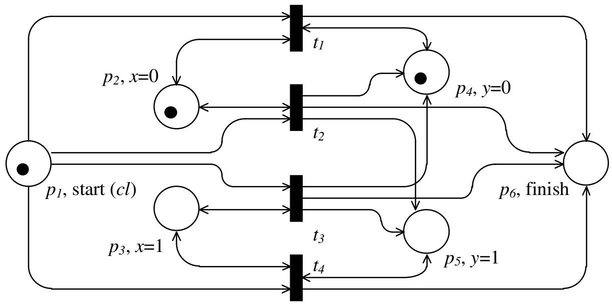

Figure 3 Model of latch in Figure 2.

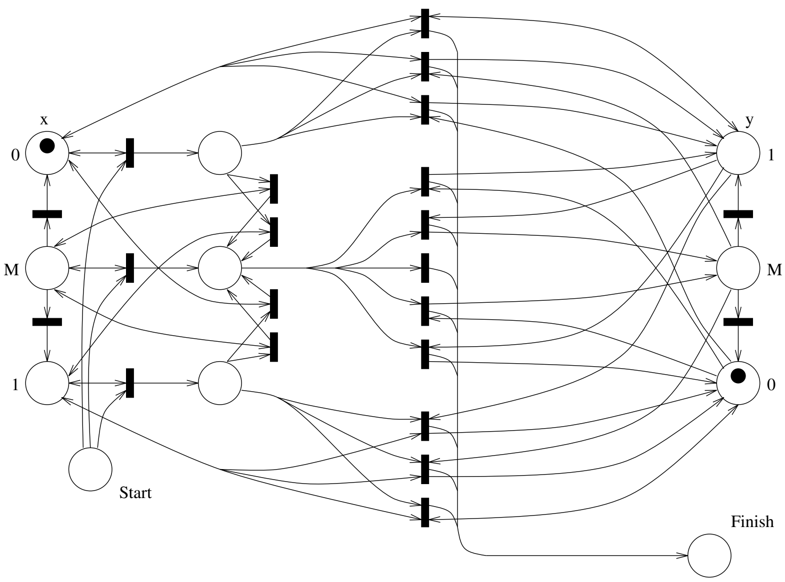

Figure 4 Non-atomic model with metastable states for the D-type latch.

A modified Petri net model of the assignment operation of Figure 2 is shown in Figure 4. In this model the assignment statement is not assumed to be atomic. 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* has been received by the latch but the Q and $\overline{Q}$ signals have not been set. This recognises the fact that the device may have non-zero “set up time” and “propagation delay” and during this time period other things may happen elsewhere in the system. 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. Both the input and output signals are recognised as being capable of metastable operations and therefore are represented as having three states each. 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 only happens when the input signal changes when the clock signal arrives.

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. Instead of operating on the edges of a clock signal, 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 usually acting on levels rather than on edges. For instance, the latch could be transparent when the control signal is high and latched when it is low, or vice versa [9, 10].

From Figure 4 it is obvious that complete latch models tend to be rather complicated to present in P/T net form. The model of the transparent latch, therefore, is here presented as a coloured Petri net [11] in Figure 5.

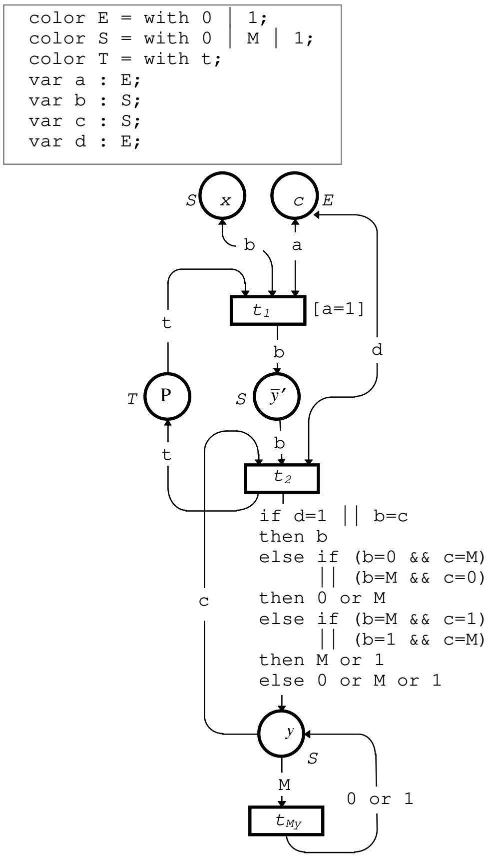

Figure 5 Model of a transparent latch.

Similarly to the model of the D-type latch in Figure 4, the model for the transparent latch also 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 groups of transitions represent the situation of "statement in progress". In the coloured Petri net model, the groups of transitions are simplified into single ones. The coloured Petri net is presented in the fashion of [11].

In this model the colours and presence of tokens in the places are used to represent the states of the latch. Place *c* 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 $c = 1$ (The token in place *c* has the colour of 1) and disabled otherwise. The variable *c* 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.

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, this model is intended to be as locally stand-alone as possible. Therefore no statement starting and ending places are included in the model as no assumption is made on the sequential nature of the overall process that includes the assignment statement.

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 are to be reported in [12] and [13].

Acknowledgement

The work at Newcastle has been supported by EPSRC grant GR/K70715 (project HADES).

References

1 Simpson, H.R., "Real time networks in configurable distributed systems", Int. Workshop on Configurable Distributed Systems, Imperial College, London, UK, March 1992.

2 Simpson, H.R., Methodological and notational conventions in DORIS real time networks, Dynamics Division, BAe, 11 February 1994.

3 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.

4 Lamport L., "On interprocess communication, Part I: Basic formalism", Distributed Computing, Vol. 1986, No.1, pp.77-85, Springer-Verlag, 1986.

5 Lamport L., "On interprocess communication, Part II: Algorithms", Distributed Computing, Vol. 1986, No.1, pp.86-101, Springer-Verlag, 1986.

6 Kleeman, L. and Cantoni, A., "On the unavoidability of metastable behaviour in digital systems", *IEEE Transactions on Computers*, C-36(1):109-112, January 1987.

7 Marino L.R. "General theory of metastable operation", IEEE Trans. Comput., Vol. 30, No. 2, pp.107-115, 1981.

8 Mendler, M. and Stroup, T., Newtonian arbiters cannot be proven correct. *Formal Methods in System Design*, 1993(3):233-257, December 1993.

9 Horowitz, Paul and Hill, Winfield, The art of electronics, 2nd ed., Cambridge University Press, Cambridge and New York, 1989.

10 Cortadella, J., Lavagno, L., Vanbekbergen 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 Jensen, K., Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts. Monographs in Theoretical Computer Science, Springer-Verlag, 1992.

12 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.

13 Clark, I.G., PhD thesis, King's College, University of London, in preparation.