Studying the data loss and data re-reading behaviour of a four-slot asynchronous communication mechanism using stochastic Petri net techniques

A. Madalinski[*], F. Xia and A. Yakovlev
Department of Computing Science, University of Newcastle

30th November 1999

Abstract

The data loss and data re-reading behaviour of a four-slot asynchronous communication mechanism (ACM) depends among other things on the rate of the writer and reader process. Data loss occurs if the reader does not obtain an item of data written by the writer, and data re-reading occurs when the reader obtains data which it has obtained during previous read cycle. Using stochastic Petri net (SPN) techniques the occurrence of data loss and data re-reading of the four-slot ACM has been investigated and their relationship to the relative speeds of the reader and writer processes analysed quantitatively.

1 Introduction

In distributed and real-time systems shared memory is often used for communication between concurrent processes. An access conflict may occur if the data transfer is not synchronized. An asynchronous communication mechanism (ACM) was proposed in [1] which is able to communicate with no mutual timing. The data is passed between two processes, a writer and a reader, through a shared memory location (slot) where the access to these slots is coordinated by small shared control variables. Schematically, data communication between processes is shown in Figure 1.

The four-slot mechanism in [1] is asynchronous in that operation by one process has no timing effects on operation by the other. The algorithm is shown in Table 1. The writer and reader processes are single loops with three statement each. A four slot array $d[0,0]$ to $d[1,1]$ represents the data storage. The control variables n, l, r, $s[0..1]$, $v[0..1]$, which are either single bits or vectors of two single bits, steer the access to the slots. The variables n, l and r, respectively, indicate which is being used to assemble new data, which slot holds the latest data and which has been selected for reading. The control variables s and v indicate, respectively, the slot which holds the latest data and which slot has been selected for reading. The four-slot ACM is shown schematically in Figure 2.

[*]Contact: A.A.Madalinski@newcastle.ac.uk

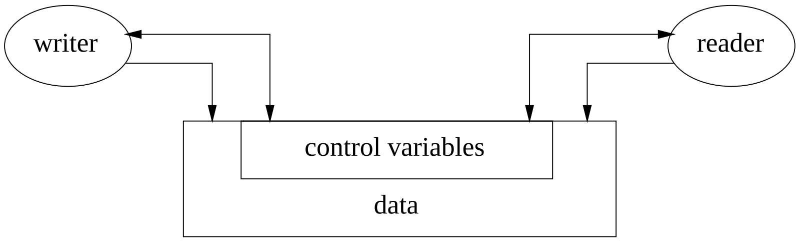


Figure 1: Asynchronous data communication mechanism

writer	reader
$wr : d[n, \overline{s[n]}] := input$	$r0 : r := l$
$w0 : s[n] := \overline{s[n]}$	$r1 : v := s$
$w1 : l := n \| n = \overline{r}$	$rd : output := d[r, v[r]]$

Table 1: Algorithm of the four-slot mechanism

The four-slot ACM guarantees data freshness and coherence. In the former case data obtained by a read operation are the latest supplied by a write operation, and in the latter case data obtained by a read operation has been completely written by only one write operation, so that the data is passed always as a coherent set. This ACM is said to be fully asynchronous because the writer and reader processes are entirely independent to each other temporally.

This mechanism employs control variables to avoid conflicts on data memory thus the problem of the synchronization between the processes is shifted from data memory to the control logic.

The data loss and data re-reading properties of this mechanism are more complex. Data loss occurs when some items written into the ACM by the writer are not eventually obtained by the reader. Data loss can be described quantitatively by the rate of its occurrence. Data re-reading occurs when the reader obtains an item which it has obtained during an earlier cycle. Data re-reading can also be described quantitatively by the rate of its occurrence.

No realistic ACM of finite size can avoid data loss and data re-reading when operating in fully asynchronous mode. Data loss will occur if the writer is faster than the reader and data re-reading would be inevitable if the reader is faster than the writer.

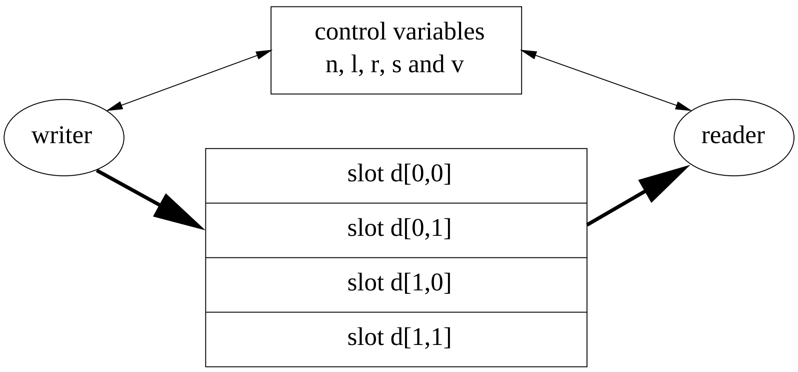


Figure 2: Scheme of the four-slot mechanism

2 Petri Net model of the four-slot mechanism

In this section a Petri net model is described, which was presented in [3] to study the characteristics of the four-slot mechanism. In order to maintain readability the PN model was divided into sub-models for the writer and reader with three sub-nets each for a statement of the algorithm in Table 1.

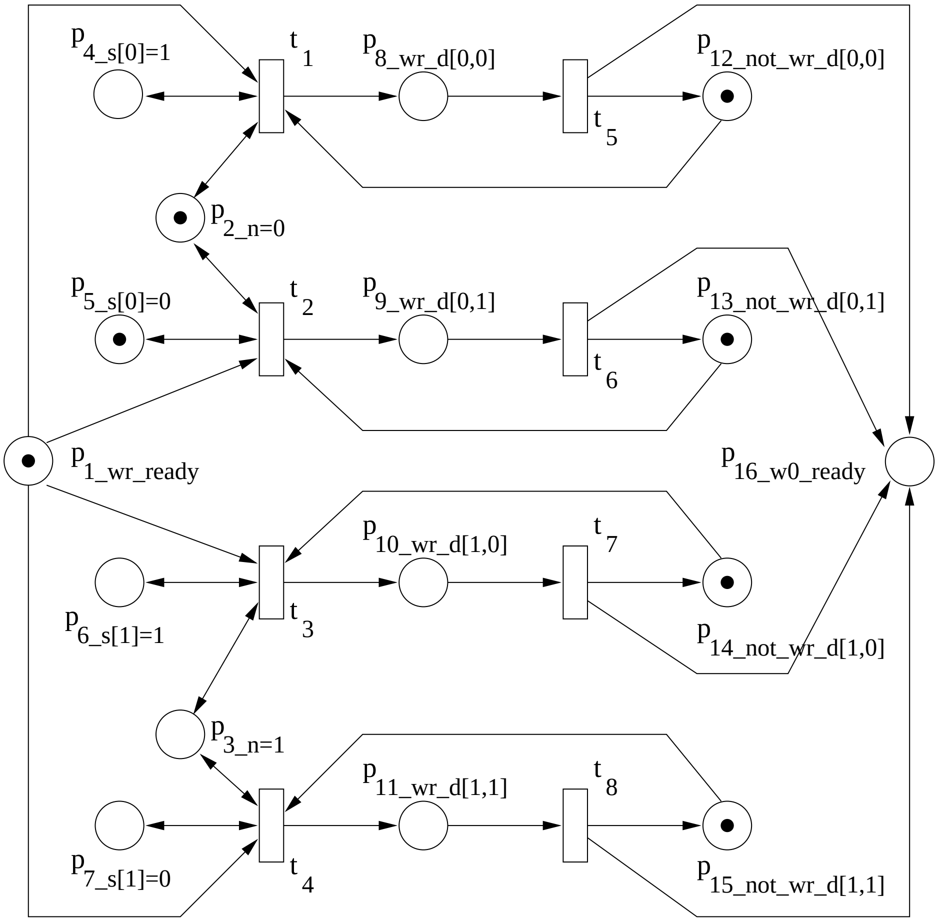


Figure 3: PN of the first writer statement wr

The writer process is divided into three statements wr, $w0$ and $w1$. The first writer statement wr is the writer slot access whose model is shown in Figure 3. In this sub-net each slot has its own start writing, writing, end writing and not writing cycle. The value of the control variable $s[0..1]$ determines which slot the writer would go to.

The second statement $w0$ is the first part of the write post sequence. In this sequence the control variables are updated by the writer. The value of the variable s changes to its complement. The second part of the post sequence, the $w1$ statement, is a parallel assignment of the control variables n and $\overline{r}$. Figure 4 shows the model of the writer post-sequence $w0$.

The reader model is similarly established as the writer model. The first and second statements $r0$ and $r1$, respectively, denote the read pre-sequence, where the control variables are updated. The third reader statement rd represents the reader slot access, where each slot has its own start reading, reading, end reading and not

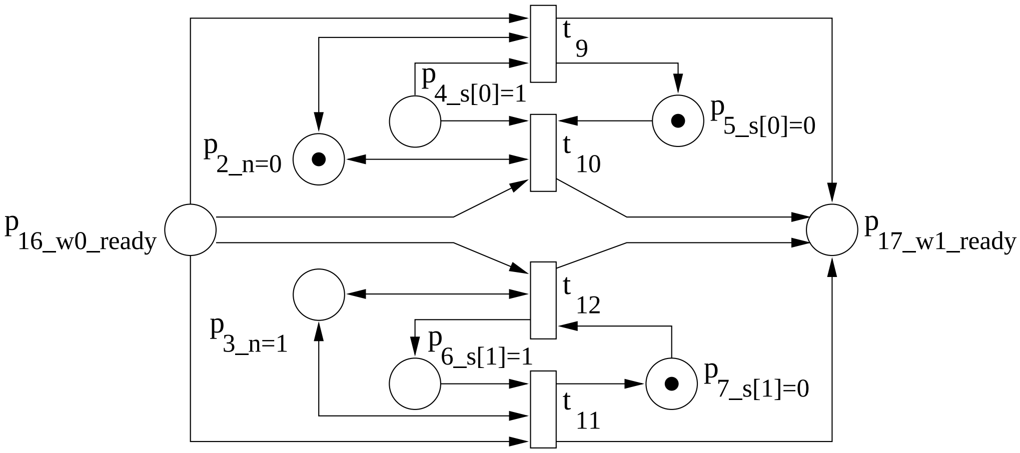


Figure 4: PN of the second writer statement $w0$

reading cycle.

3 Modelling and analysis of data loss and data re-reading

Using the GreatSPN2.0.2 tool [4] an SPN model of the four-slot mechanism was constructed and analysed. The Petri net model described in the previous section was modified in order to study the average rate of data loss and data re-reading of the four-slot mechanism using SPN techniques. The GreatSPN2.0.2 tool is a software package for modeling, validation, and performance evaluation of distributed systems using Generalized Stochastic Petri Nets, which provides among other performance analysis methods Markovian solution.

The periods of time outside the ACM access taken by the writer and reader in each cycle are known as writer extra-ACM delay and reader extra-ACM delay. They are shown schematically in Figure 5. These periods of time are varied when investigating the behaviour of a faster writer and slower reader and vice versa.

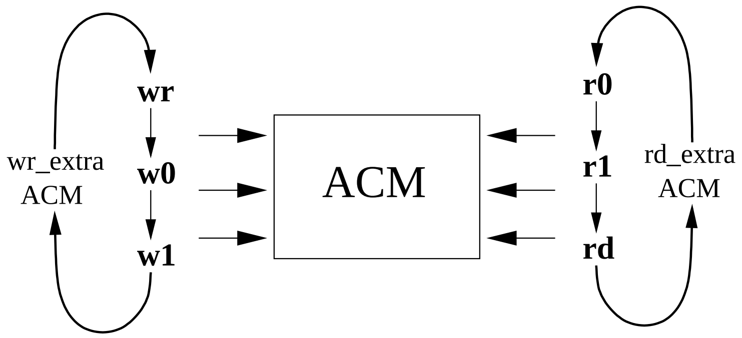


Figure 5: Scheme of an ACM with extra delay

The modeling of the writer extra-ACM delay is depicted in Figure 6. The shared place between the statements $w1$ and wr, $p_{1\_wr-ready}$, was divided and a transition, which represents the delay, was introduced. The modeling of the reader extra-ACM delay was realized in the same way. Between the statements $r0$ and rd a transition was added and the shared place of this statements was divided.

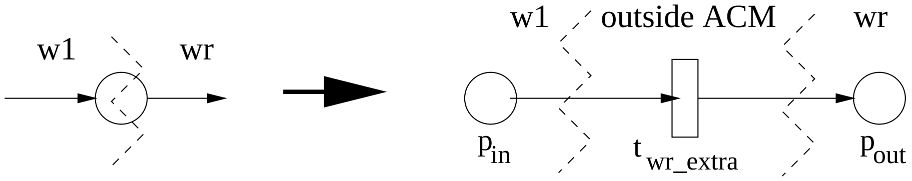
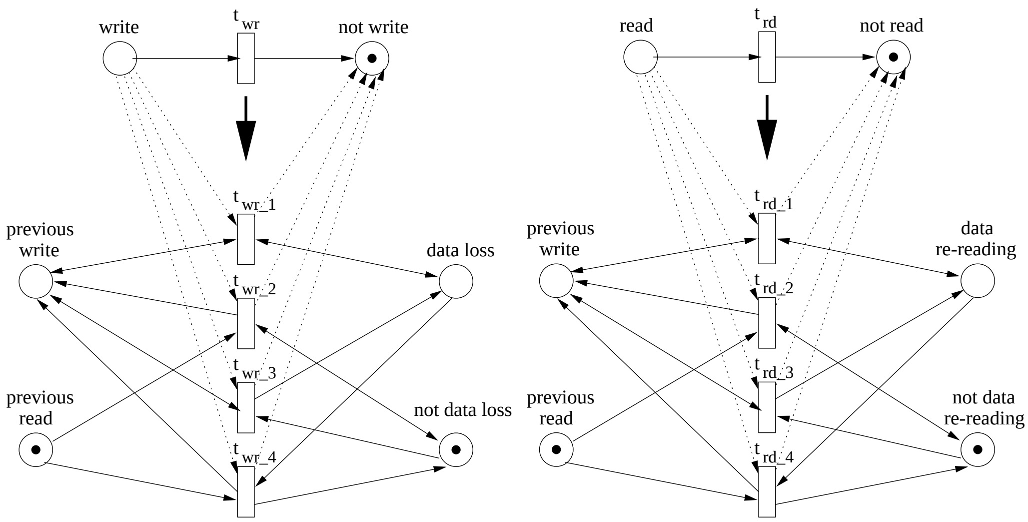


Figure 6: Implementation of the writer extra-ACM delay

Data loss can be denoted as a write operation accessing a particular slot under the condition that the previous access to the same slot was also by a write operation, so that the reader does not obtain the previous item of data written by the writer. No data loss occurs if the previous access to the slot was by a read operation. This consideration leads to the extension of the sub-net representing wr.



(a) wr statement

(b) rd statement

Figure 7: Part of the modified wr and rd statement models

This extension is depicted in Figure 7(a). The wr sub-net was modified by adding the places "previous write", "previous read", "data loss" and "not data loss" between the writing and not writing places for each slot. The transition t_{wr} (finish writing) was divided into four transitions which are denoted by an index 1...4. The initial markings are set to places previous read and not data loss. Now data loss can be quantitatively described as the token distribution in the data loss places.

Similar consideration must be made to the read statement to obtain the rate of data re-reading, which can be denoted as a read operation whereby the previous operation was also a read operation. For instance, the rd statement model is modified in the way shown in Figure 7(b).

For the model in Figure 7, if a writer access to the slot follows a reader access, there is no data loss. This is represented by the firing of either t_{wr-2} or t_{wr-4} which marks the previous write and the not data loss places and unmarks the previous read and no data loss places. If a writer access follows a previous writer access without a reader access in between, there is data loss. This is represented by the firing of either t_{wr_1} or t_{wr_3}, resulting in the place data loss being marked. The "data re-reading", "not data re-reading" and previous read places are updated by the reader transitions t_{rd_1} to t_{rd_4} in a similar way.

All transitions were assumed to have exponential distribution with the mean time given in Table 2, which were obtained from analogue simulations [2]. The mean times taken outside the writer and reader cycle, respectively, should vary between 10 and 100ns in order to completely investigate the behaviour of slower writer and faster reader and vice versa.

time $[ns]$	
WR: 4.42	R0: 2.18
W0: 3.67	R1: 2.59
W1: 5.97	RD: 3.56

Table 2: Data from analogue simulation

4 Results of analysis

The diagram in Figure 8 shows the occurrence of data loss and data re-reading depending on the ratio of the time taken outside the writer and reader process denoted by rd_extra_ACM and wr_extra_ACM.

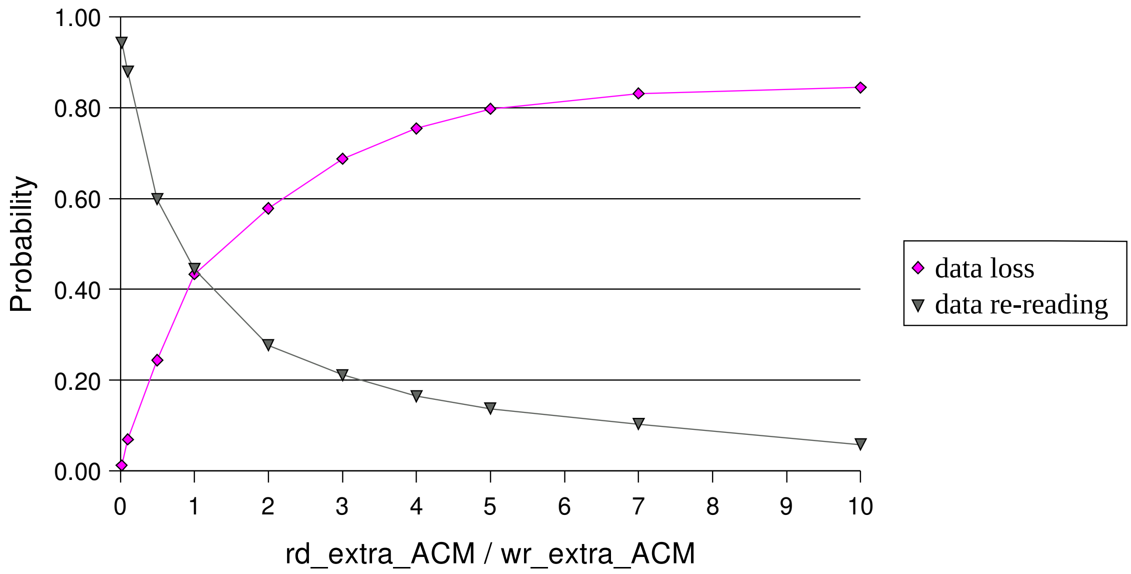


Figure 8: Occurrence of data loss and data re-reading

5 Conclusion and acknowledgement

The analysis of data loss and data re-read of the four-slot ACM was performed using the GreatSPN2.0.2 tool. The existing PN model for the four-slot ACM was modified and some extensions were introduced in order to perform this analysis using SPN techniques.

The result of the analysis shows that data loss increases when the reader becomes slower relative to the writer and data re-reading increases when the reader becomes faster relative to the writer. The curves appear to be exponential in shape.

This work is partly supported by the EPSRC projects Comfort (grant GR/L93775) and Timbre (grant GR/L28098) at University of Newcastle.

References

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

[2] F. Xia, A. Yakovlev, D. Shang, A.Bystrov, A. Koelmans, D. J. Kinniment. An asynchronous communication mechanism using self-timed circuits, Technical report, University of Newcastle, Department of Computing Science, October 1999.

[3] F. Xia. MASCOT, Petri nets, and asynchronous data communications, PhD Thesis, King's College, University of London, 1999.

[4] University of Turin, http://www.di.unito.it/~greatspn/