ACMS IN MATLAB

Fei Hao, E. Graeme Chester
School of Electrical, Electronic and Computer Engineering, Univ. of Newcastle upon Tyne, UK

ABSTRACT

MATLAB® is a popular tool for the engineers in the electrical and electronic fields. ACMs can also be modelled in MATLAB® in order to investigate the applications in control system. An approach of converting a Petri net model into a Simulink model is studied. As an example, a Signal ACM model is created in Simulink with Stateflow. The simulation results show that the model maintains all the required asynchronous properties.

Key words: ACM, Signal, Petri net, MATLAB®

1. INTRODUCTION

1.1 ACM

An Asynchronous data Communication Mechanism (ACM) is a scheme, which manages the transfer of data between two processes not necessarily synchronised for the purpose of data transfer [1]. The provider of data is called the "writer", and the user of data is referred to as the "reader". The general scheme of these kinds of data communication mechanisms is shown as follows:

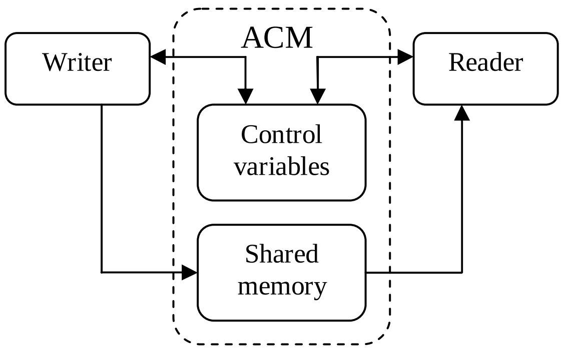


Figure 1 ACM Using Shared Memory and Possibly Control Variables

1.2 MATLAB, SIMULINK and Stateflow

MATLAB® is a commercial "Matrix Laboratory" package which operates as an interactive programming environment. It is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

Simulink is a graphical extension to MATLAB for the modelling and simulation of systems. In Simulink, systems are drawn on screen as block diagrams. Many elements of block diagrams are available (such as transfer functions, summing junctions, etc.), as well as virtual input devices (such as function generators) and output devices (such as oscilloscopes). Simulink is integrated with MATLAB and data can be easily transferred between the programs.

Stateflow is a graphical design and development tool for control and supervisory logic used in conjunction with Simulink. It provides clear, concise descriptions of complex system behaviour using finite state machine theory, flow diagram notations, and state-transition diagrams all in the same Stateflow diagram. Stateflow brings system specification and design closer together. [2]

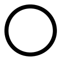
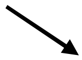
MATLAB® is a popular tool for the engineers in the electrical and electronic fields. Most of the control systems can be modelled in MATLAB and Simulink. To investigate the applications of ACMs in control system, it is one approach to model ACMs in MATLAB®, and apply the model into a control system model.

The algorithms of ACMs are always representing by Petri Net models. Is it possible to convert a Petri Net model into a stateflow model? To answer this question, a comparison should be made between the Petri Net and the stateflow.

2. PETRI NET AND STATEFLOW

Petri net is a mathematical representation of a system with interacting concurrent components. The basic components are places (also represent states) and transitions. Stateflow is a graphical design and development tool for control and supervisory logic. It plays games with states and transitions. Table 1 shows the comparison of the Petri Net and stateflow:

TABLE 1 -Petri Net and Stateflow

	Petri Net	Stateflow
Place		
Transition		

As the basic protocol of asynchronous communications, handshake is the key part of modelling ACMs in the stateflow. A four phase handshake follows this order: send a request, wait for the acknowledgement send from the other side, release the request, and release the acknowledgement from the other side.

In the stateflow, the handshake can be modelled as following: the requests can be generated in the state enter actions, which are executed when entering the states; then wait for the acknowledgements in the transition conditions, which lead to the executions of the transitions; if the conditions are satisfied, the requests are released in the state exit actions, which are executed when exiting the states; and finally release the acknowledgements. The diagram of the handshake in stateflow is shown in the figure 2.

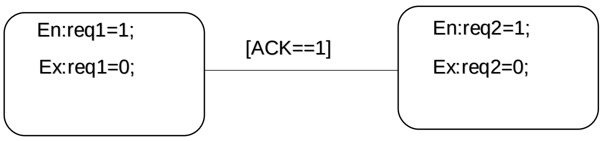


Figure 2 Handshake in Stateflow

3. MODELLING OF A 3-SLOT SIGNAL ACM

The algorithm of the 3-slot Signal ACM is shown in Figure 3, which can be found in [3]. For the Petri Nets of its read side and writ side, please refer to [4] for the details.

Write Side:
wr: write slot w;
w0: l:=w;
w1: w=differ (l, r);

Read Side:
r0: wait until (r!=l) r:=l;
rd: read slot r;

Figure 3 Algorithm of 3-Slot Signal

In the common cases, to convert a Petri Net into a stateflow, just replace the circles in Petri Net with the rectangles in the stateflow and the bars with the arrows. However, there is one case must be mentioned. If to fire one transition requires two places holding tokens simultaneously, the converting does not obey the rule. The reason is that in the stateflow only one state can be active at a time. In this case, one of these two places must be regarded as a transition condition in the stateflow.

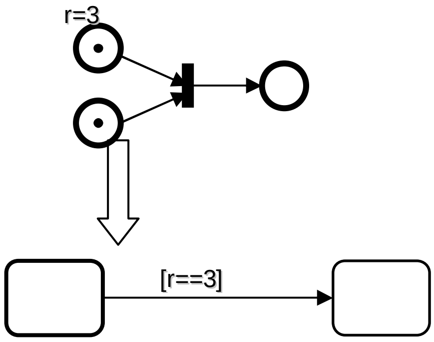


Figure 4 One Special Case

According to these two rules, a stateflow model for is established. Figure 5 shows the model for the write side.

4. RESULTS AND DISCUSSIONS

The simulation of the model was carried out in the Simulink environment. The resulting waveforms are shown in Figure 6.

In this sequence, after data items 7 and 13 were read by the reader, another read request arrived. The reader did not respond to the requests until new data items were available. During this period, the read request stayed high. On the other hand, when the writer delivered the data items quickly, such as 8 to 13, overwriting occurred (9, 10, 11 and 12 were overwritten). The time taken by the reader and writer outside the ACM was controlled by two independent random number generators written in MATLAB codes. An exponential distribution was assumed and the same mean value was set for both w0 to wr and rd to r0. This gave enough variation for reader waiting and writer overwriting to appear.

5. CONCLUSIONS AND FUTURE WORK

An approach of converting Petri Net model into a stateflow model is presented. A 3-slot Signal ACM model is created by this approach. The simulation shows the reasonable results. Pool, Channel and Message also have Petri Net models, so they can be modelled with the same method easily. The success of modelling ACMs in MATLAB provides the possibility of investigating the applications of ACMs in MATLAB.

More works need to be done on creating general models of each type ACMs. Furthermore, Applications of ACMs in control systems are to be investigated.

6. ACKNOWLEDGEMENT

This work is part of the Coherent project (http://async.org.uk/coherent) at the Newcastle University supported by the EPSRC grant (GR/R32666). The authors wish to thank Alex Yakovlev, Fei Xia, Ian Clark and other members of the Coherent project for their valuable ideas and comments.

7. REFERENCES

[1] Fei Xia, Alex V. Yakovlev, Ian G. Clark, Delong Shang. "Asynchronous communication mechanisms: classification and hardware implementations", MPCS'02, Fourth International Conference on Massively Parallel Computer Systems, sponsored by Euromicro, 10-12 April 2002, Ischia, Italy.

[2] MATLAB Help manual

[3] Fei Xia, Ian Clark. "Algorithms for Signal and Message Asynchronous Communication Mechanisms and their Analysis", Volume 50, Number 2 (2002), pp.205-222, Fundamenta Informaticae, IOS Press

[4] F. Hao, A. Yakovlev, E.G. Chester, F. Xia, I.G. Clark, D. Shang. "Implementation of a three-slot signal ACM", 13th UK Asynchronous Forum, 16-17th December 2002, University of Cambridge, Computer Laboratory.

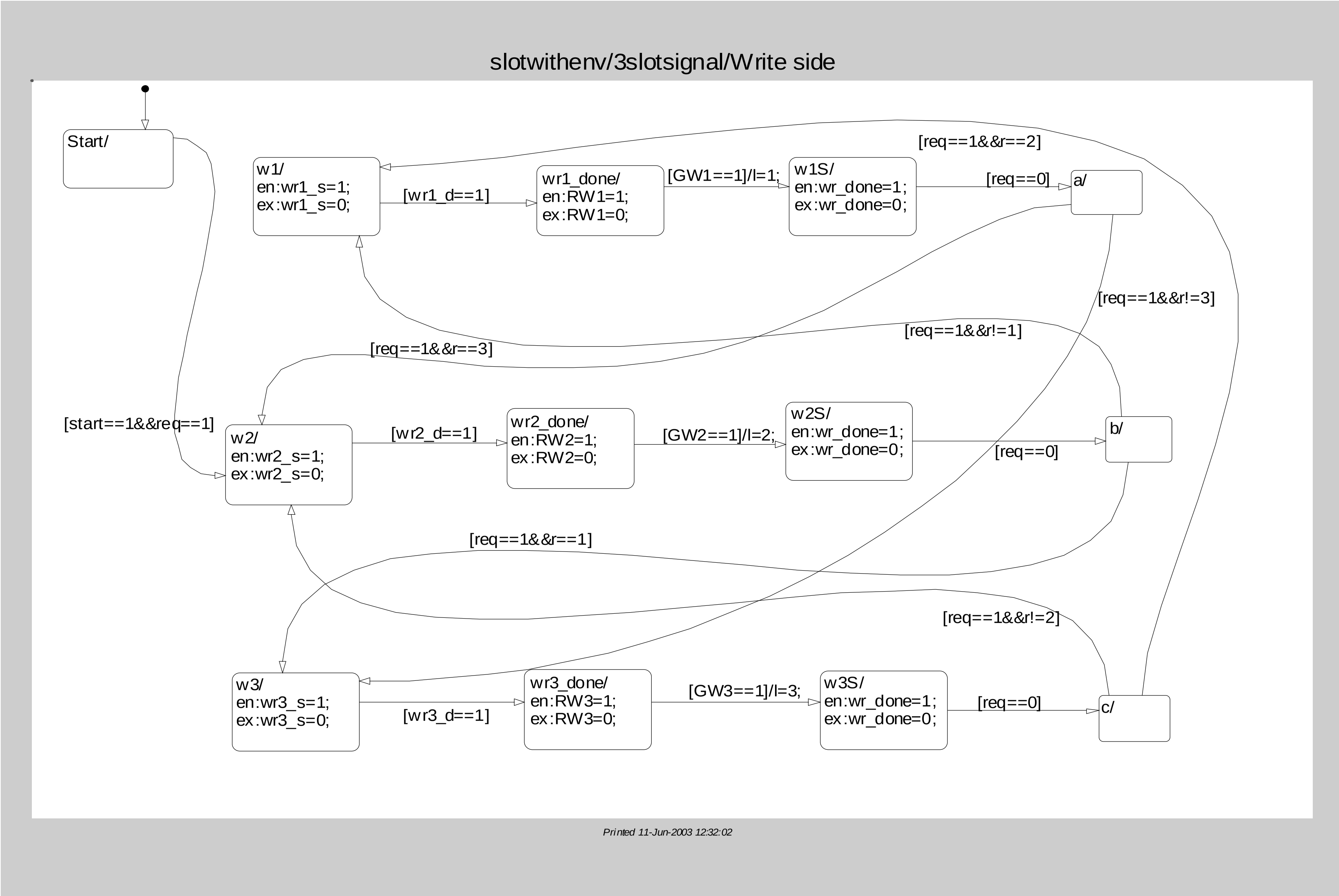


Figure 5 Stateflow Model for Write Side

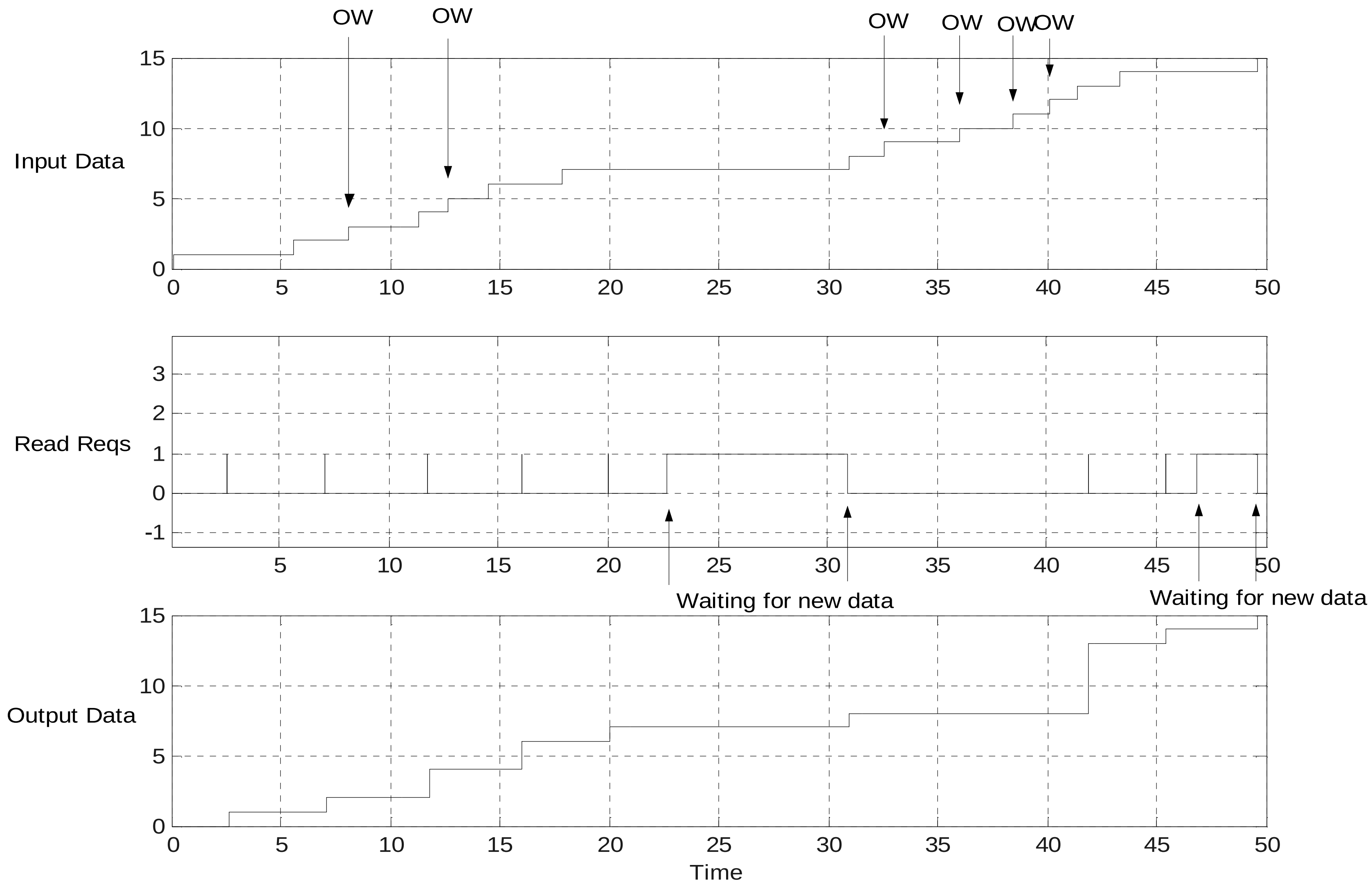


Figure 6 Resulting Waveforms