Testing a Self-Timed Asynchronous Communication Mechanism (ACM) VLSI Chip

D. Shang, F. Xia, A. Yakovlev
Department of Computing Science, University of Newcastle upon Tyne, NE1 7RU, U.K.

Emails: {Delong.Shang, Fei.Xia, Alex.Yakovlev}@ncl.ac.uk

Abstract

A testing method and on-chip testing circuits for our VLSI chip of an asynchronous communication mechanism (ACM) using self-timed circuits are presented in this paper. The main testing objective is in the three important ACM properties, asynchrony, coherence and freshness. In addition, the general function of the chip and data loss are also of interest. Although this work is a specific solution for a specific problem, it is an important step in the development of a methodology of testing hardware solutions of fully asynchronous systems.

1 Introduction

In general, an ACM is accessed by two processes, the writer, which supplies the data, and the reader, which uses the data. An ACM may be implemented with shared memory in the scheme shown in Figure 1 (a). The two access processes may operate with various degrees of temporal independence. Depending on the degrees of synchronization between the access processes, ACMs may be classified into several types.

One type of ACMs is known as the Pool [1]. A Pool allows full temporal independence between the writer and reader processes, i.e. these two processes may operate fully asynchronously. A number of implementations for the Pool have been described before, such as the four-slot ACM [3] and our three-slot and four-slot ACMs [4, 5].

In the four-slot ACM of [3], no use is made of mutual exclusion, and timing interference is completely eliminated by shifting the problem of synchronization from data memory to control variables. However some timing assumptions are made in that fundamental mode (FM) is used.

In our three-slot and four-slot ACMs, the circuits are Speed Independent (SI), which results in the advantages such as safety, lower power consumption and so on. The synchronization problem is moved from the control variables to handshake interface because the SI circuits can be simply seen as handshake circuits. These slot mechanisms are not a pure form of fully asynchronous communication mechanisms because mini-interlock is introduced by employing mutual exclusion (MUTEX) in implementation. However the control variables in SI solution are safe compared to those in the fully asynchronous, FM solution.

Both of our slot mechanisms have been implemented using the Cadence VLSI design tool (0.6 micron technology). The four-slot ACM has been fabricated by EuroPractice.

2 Testing Aim

Testing is an essential part of the overall realization of a VLSI. The main purpose of testing is to verify the correct operation of the circuits and detect any possible presence of certain processing faults [8]. In our case, the three properties vital to ACMs need to be verified. These properties, asynchrony, coherence and freshness are described in [2, 3]. In addition, whether this VLSI chip works at all and whether it works as expected need to be tested.

3 Testing Methodology

Logic circuits can be divided into two types, one is the classical clocked circuits and the other is self-timed circuits. In general, clocked circuits can be tested by using testing patterns, but testing self-timed circuits is not so simple. This is because it is more difficult to correctly estimate their precise timing. It is often necessary to use on-chip testing circuits and make use of the handshake driven nature of self-timed circuits.

The ACMs are globally unsynchronsed systems and they are special applications implemented by using self-timed circuits. Their three main properties are also unique. So testing such systems is a difficult problem, even more difficult than testing self-timed circuits or just circuits without global clock in general. In this work we will show that such system can be tested by looking at sequences of data items, rather than individual items, whilst varying rates of data communication etc. These are all completely new aspects of testing asynchronous systems.

4 On-chip Testing Circuits

During the development of a hardware implementation of the four-slot ACM, the problem of testing for the important properties and general functions in hardware arose [4]. Although general functions and the properties such as data freshness and data coherence have been verified for the ACM both theoretically and via digital simulations (using the Cadence tool), testing for such properties in hardware is necessary to show that the fabricated chip does function as designed. The ACM hardware is implemented with relatively fast VLSI technology and the testing is desired to be at its maximum performance. In order to force the ACM to work at maximum speed, the testing circuits, the writer and reader hardware, had to be built into the same chip as the ACM and kept as simple as possible.

A testing method has been discussed in [7]. In particular, it was shown that the property of freshness could be verified by testing for sequence, whilst varying the relative speeds of the writer and reader processes. Based on this, a simple solution consisting of a counter serving as the writer and a FIFO buffer serving as the reader is proposed. An on-chip FIFO buffer is needed because external connections cannot keep up with the production of data from the reader part of the ACM and dumping data out directly implies slowing down the reader via its external handshakes. With an on-chip FIFO, data from the reader can be downloaded off-line after an experiment run. Because the ACM circuits are implemented with self-timed circuits, it is more difficult to correctly estimate their precise timing. Since the ACM interfaces externally with the writer and reader once per cycle through standard start/done handshake signals, the relative timing of individual writer and reader cycles may be manipulated in testing and it is better if we can adjust the relative lengths of the extra-ACM delays of the writer and reader to test the properties. So, in our ACM chip, delay parts for both the writer and reader are employed. In order for all of this to work properly, control circuits are needed in the testing circuits. The whole block diagram is shown in Figure 1 (b). In order to keep the testing circuits simple and testing process efficient, the testing circuits and the ACM can

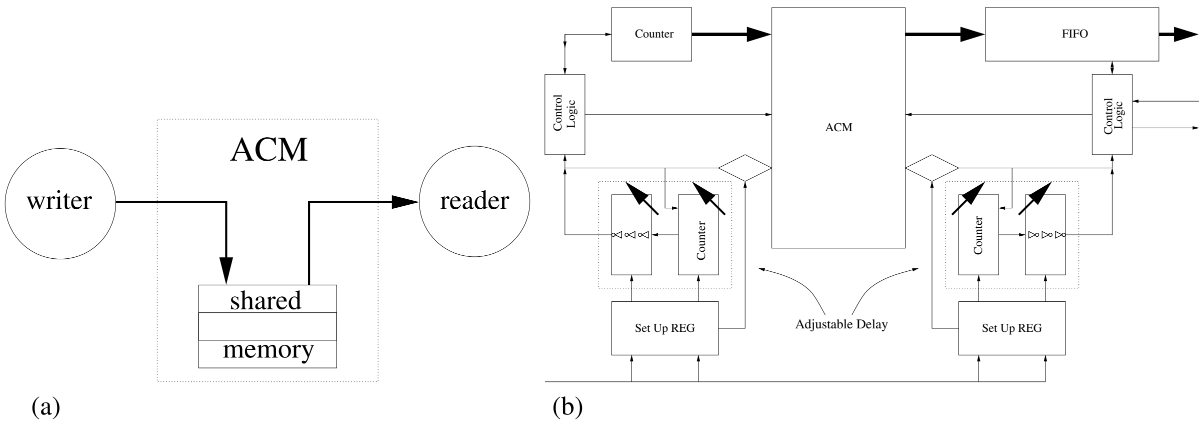


Figure 1: (a) A simple ACM and (b) The block diagram of the on-chip testing circuits and the ACM

be run in several modes which are set up initially.

The testing circuits work as the environment of the ACM. They can supply the data to the ACM, and collect the data from the ACM. An 8-bit input counter, which is a ripple carry counter, is used as the data generator (writer). Data with values from 0 to 255 can be sent to the ACM in increasing order. On the opposite side, the FIFO buffer can collect the data from the ACM as the reader. In order to analyze the data collected from the ACM, the FIFO should be big enough to allow as much data as possible be stored. On this chip, the ACM and its on-chip testing circuits share the chip area with circuits for other projects. As a result, there is not enough space to build a really big FIFO. In the end a 64 stage FIFO, which occupies more than half of the total chip area used by the ACM and its testing circuits, is included on-chip.

In order to test the properties of the ACM, some relative timing assumption is needed. So in the on-chip testing circuits, two delay parts are designed for this purpose for both the writer and the reader. For each delay part, there is an 8-bit counter designed by using the same technology as the input counter. The delay length of these counters can be set up by writing the initial values which are chosen from 1 to 255 based on the requirement to the control circuits. When the counter counts from 0 to the initial value, it then stops followed by other circuits. However, from the time point of view, the counter is discrete with too large a granularity. In order to adjust the delay time as finely as possible, another type of delay is also included. This part consists of inverters connected in series. In our case, there are 64 inverters in total and they are divided into 8 groups. How many group inverters are needed can be decided by setting up the initial value through the input data bus. These provide the fine-tuning required.

In the on-chip testing circuits, because the delay counters are introduced, a control circuit is needed to decide either to continue counting or to stop (done). The function of the circuit is described by the Signal Transition Graph (STG), shown in Figure 2 (a).When delay request (delay_start) arrives, following this STG, the counter will add one, and then compare with the value set up initially. If these are equal, the done signal will be produced and then the delay_start signals will be withdrawn; if not, the go_on signal will be given to stimulate the counter again. The circuit implementing this function is shown in Figure 2 (b). This circuit is derived by using the Petrify tool and optimizing by hand.

In order to test the chip, in the writer part, the input counter with two working modes is developed. One mode is stoppable, that means when it counts to 255, the writer will stop. The other mode is unstoppable, the writer will send data from 0 to 255 and repeat this cycle.

The operations in the writer part are: (1) preparing the data, (2) writing into the ACM, (3) making sure whether to finish and (4) adjusting the delay length.

The control circuits are built by using David cells (based on SR flip-flops) and are translated from the high level Petri Net specifications directly. An automatic synthesis tool employing this direct translation technique has been developed by our group and more than 30 types of modified David cells have so far been developed.

In the reader part, the main control circuits are implemented by using another method, different from David cells. Four operations, reading ACM, writing to FIFO, reading from FIFO and setting delay, are connected to each other by handshake circuits. This FIFO works as a micro-pipeline [6]. After the writing to the FIFO is done (just write to the first stage and on completion, produce the done signal), the delay action, which is used to adjust the delay length, will happen. The FIFO reading operation is only used by external connections for collecting the data from the chip.

5 Testing Methods and Results

As stated above, certain basic testing circuits which provide some testing functions, such as the input counter, the FIFO buffer, the delay parts and the control circuits, have been built on the same chip with the ACM in order to obtain results when the ACM is running at full speed.

In fact, what we do in testing is to analyze the data collected using the following testing programs from the observation points, including the output of the FIFO, and the timing relationship for the logic signals. They are: (1) The writer and reader are both operated as fast as possible, (2) The writer is operated faster than the reader, (3) The writer is operated slower than the reader, (4) With the writer stopped, the reader is operated as fast as possible, and (5) With the reader stopped, the writer is operated as fast as possible.

5.1 Testing General Functioning

The general functioning of the ACM and on-chip testing circuits need to be tested. If they are not correct, any subsequent testing for the properties of the ACM can not be considered.

We analyzed the timing of the start/done handshake signals, initial values and write the data to the ACM and then read it from the FIFO. The results demonstrated that the general functioning is as expected.

5.2 Testing Asynchrony

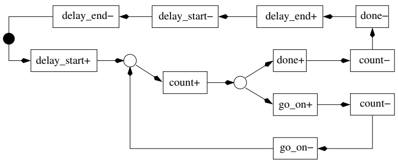
In asynchrony testing, the assumption is that if the writer and reader do not run independently, that means the writer (or the reader) will wait for something coming from the reader (or the writer), when the writer (or the reader) runs slowly or stops, the reader (or the writer) should be affected. In other words, if the two sides are temporally dependent, the faster side will be slowed down by the slower one. In fact, in testing, no matter how we adjust the delay time for the reader and writer, no waiting occurs according to the waveforms observed on both a digital and an analog oscilloscopes. This indicates that the ACM on this chip does support asynchrony.

5.3 Testing Data Freshness, Coherence and data loss

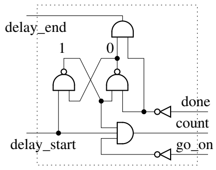
The same testing methods are used for these properties as in Section 5.2. The difference is in the observed signals. Here we will focus on the data results collected from the ACM when we adjust the delay time for the writer and reader. Since the order of the data going into the ACM from the input counter is known, if we adjust the relative timing for the writer and reader, the data collected from the FIFO buffer should be regular and have the same order with the input data. Otherwise data freshness is violated. In our case, in order to save the chip space and money, the width of the data transfered to the ACM and the FIFO buffer is 8 bits. With such a data packet size, it is very difficult to verify possible violations of data coherence. In the circuits, the logic of this mechanism guarantees that the writer and reader cannot access the same slot (packet) at the same time. If the writer and reader can access the same slot at the same time, the data will then be combined and the frequency of the data appearing will be confused. This implies that loss of data coherence should show up in the observation as loss of data freshness. So by testing for data freshness we are also testing for data coherence.

In actual testing, we obtained a wide range of results. Two examples are shown in Figure 2 (c) and (d).

From the testing results, we can see that the data from the FIFO is very regular without any obvious violations to data freshness and coherence, and the patterns of data overwriting and re-reading (characteristic of the four slot ACM) can be clearly seen.



(a)



(b)

With Twrite =80ns, Tread = 110ns, Tsample = 40ns.											
1.	41H	7.	44H	13.	46H	19.	49H	25.	4CH	31.	4FH
2.	41H	8.	44H	14.	46H	20.	49H	26.	4CH	32.	4FH
3.	41H	9.	44H	15.	46H	21.	49H	27.	4CH	33.	4FH
4.	42H	10.	45H	16.	48H	22.	4AH	28.	4DH	34.	50H
5.	42H	11.	45H	17.	48H	23.	4AH	29.	4DH	35.	50H
6.	42H	12.	45H	18.	48H	24.	4AH	30.	4DH	36.	50H

(c)

With Twrite =80ns, Tread = 80ns, Tsample = 40ns.											
1.	73H	7.	76H	13.	79H	19.	7CH	25.	7FH	31.	82H
2.	73H	8.	76H	14.	79H	20.	7CH	26.	7FH	32.	82H
3.	74H	9.	77H	15.	7AH	21.	7DH	27.	80H	33.	83H
4.	74H	10.	77H	16.	7AH	22.	7DH	28.	80H	34.	83H
5.	75H	11.	78H	17.	7BH	23.	7EH	29.	81H	35.	84H
6.	75H	12.	78H	18.	7BH	24.	7EH	30.	81H	36.	84H

(d)

Figure 2: (a) The STG, (b) The logic circuit of choice function for the delay counter, (c) and (d) Testing results

In addition, when the speed of the writer and the reader does not match, data loss may happen. The loss rate depends on the difference between the speeds of the writer and reader. The results are similar to the simulation results obtained by using the Cadence tool.

6 Conclusion

The testing results show the slot ACM implemented by using self-timed circuits performs as expected. The mini-interlock introduced by using the MUTEX does not affect the operation of the ACM appreciably. No obvious violations of the three important properties has been observed. In addition, the testing results correspond with the simulation results obtained by using the Cadence tool, which indicates that the implementation did not introduce errors.

In addition, the testing methodology is shown to be appropriate.

However, owing to the limit on the area and the pin number of the chip, the scope of the testing was necessarily limited.

Though the testing is successful and on-chip testing circuits performed as expected, there are still potentials for improvement. The testing circuits should be made more nimble and provide more observation points. We plan to continue research on developing "more asynchronous" techniques for testing inherently asynchronous properties of circuits that are free from global clock.

7 Acknowledgement

The authors would like to thank Prof. D.J. Kinniment and Dr. G. Russell for kindly offering help with testing and providing necessary equipment. This work is part of the Comfort project at the University of Newcastle supported by the EPSRC grant GR/L 93775.

References

[1] Yakovlev, A., Xia, F., Shang, D., Synthesis and implementation of a signal-type asynchronous data communication mechanism, ASYNC2001, Salt Lake City, Utah, March 2001, accepted for publication.

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

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

[4] Xia, F., Yakovlev, A., Shang, D., Bystrov, A., Koelmans, A. and Kinniment, D.J., Asynchronous communication mechanisms using self-timed circuits, Proc. Async 2000, 6th International Symposium on Advanced Research in Asynchronous Circuits and Systems, Eilat, Isreal, April 4-6, 2000.

[5] Shang, D., Xia, F. and Yakovlev, A., An implementation of a three-slot asynchonous communication mechanism using self-timed circuits, Proc. AINT' 2000, pp. 37-44, TU Delft, The Netherlands, July 19-20, 2000.

[6] Sutherland, I.E., Micropipling, Communications of ACM, 32(6):720-738,June 1989.

[7] Xia, F., Yakovlev, A. and Clark, I.G., Testing the data freshness properties of asynchronous communication mechanisms, Proc. of the 7th UK asynchronous forum, Newcastle upon Tyne, U.K., Dec. 20-21 1999.

[8] Shepherd, P. R., Integrated circuits design, fabrication and test, First published 1996 by Macmillan press Ltd.,