Asynchronous Four-Phase and Two-Phase Circuits: Testing and Design for Testability

O. A. Petlin, S. B. Furber

Department of Computer Science, The University, Manchester, M13 9PL, UK

tel. +44 (161) 275-3547
fax +44 (161) 275-6202
{oleg, sfurber}@cs.man.ac.uk

1. Background and motivation

Asynchronous circuits, i.e., circuits which perform their calculations without being clocked globally, appear to offer benefits in low power applications, promise greater design modularity, exhibit average case performance rather than worst case performance and have no clock skew problem [Birt95]. The AMULET group at the University of Manchester has been developing asynchronous engineering design methodologies which have already demonstrated the practical feasibility of complex asynchronous VLSI systems. The first successful outcome of this work was a fully functional AMULET1 microprocessor chip, an asynchronous version of the ARM 32-bit RISC microprocessor [Furb94]. The AMULET1 microprocessor was built using the micropipeline design style and a two-phase bundle data convention as described by Sutherland in his Turing Award lecture [Suth89]. The experience gained from the AMULET1 design led to the AMULET2e design, an asynchronous embedded controller with an AMULET2 core. The AMULET2 microprocessor was based on Sutherland's micropipeline design style with four-phase transition signalling [Furb97].

According to the two-phase protocol both rising and falling transition events have the same meaning. When the data is ready to be sent to the receiver the sender produces a rising (or falling) request signal which is acknowledged by a rising (or falling) signal on the acknowledge control line. Four-phase signalling differs from the two-phase protocol in that both the control signals (request and acknowledge) must be returned to zero, i.e., new data can be transmitted only when both control signals are reset. Clearly, the four-phase protocol can be divided into two distinct phases: transmission (functional) part and return-to-zero (reset) part of the protocol.

However, efficient test techniques which provide for the detection of fabrication faults have to be developed before the commercial potential of the AMULET designs can be fully exploited. Fabrication faults in CMOS circuits are caused mainly by different sorts of physical phenomena which lead to deviations in silicon properties. Most common manufacturing defects can be grouped by their impact on the faulty VLSI circuit. This allows fault models to be used for the generation of test vectors. The single stuck-at fault model is used widely in the test generation [Russ89]. A stuck-at-0 or stuck-at-1 fault in node x (x-SA0 or x-SA1) is equivalent to a break between node x and its driving circuit while V_{ss} or V_{dd} is permanently connected to node x respectively.

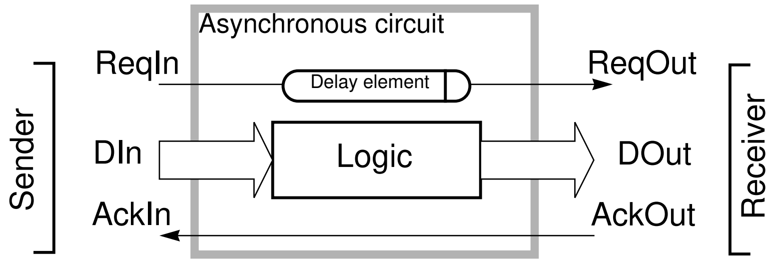

Figure 1 : *Example of the bundled data communication with data processing*

2. Stuck-at fault effects in asynchronous circuits

An example of the bundled data communication with data processing is illustrated in Figure 1. When the data is stable on the *DIn* inputs the sender generates a rising request signal on its *ReqIn* output. This request signal is delayed by the delay element for time matching the critical delay path of the data processing logic plus a safe margin. When the receiver receives a request on its *ReqOut* input it accepts the data from the *DOut* outputs of the logic circuit and produces an acknowledge rising event on its *AckOut* output. According to the two-phase protocol, the arrival of a rising acknowledge event on the *AckIn* input of the sender restarts the communication procedure described above using falling request and acknowledge events. According to the four-phase signalling protocol, the *ReqIn* and *AckOut* signals must be returned to zero before new data is sent to the data processing block. Stuck-at faults inside the processing logic exhibit themselves in the same ways as in synchronous designs. Clearly, any stuck-at fault on the control lines of the asynchronous circuit sets the faulty control line to a constant value which in turn causes the communication to halt. As a result, a stuck-at fault on a line is equivalent to an infinite signal propagation delay along this line. This so-called self-diagnostic property of asynchronous circuits in which each transition is confirmed by another has already been discussed elsewhere [Dav90, Haz92].

The delay element in the circuit shown in Figure 1 can be considered as an asynchronous control unit. In a two-phase implementation, where each transition is meaningful, the delay element can be implemented as a chain of inverters. As a result, a stuck-at fault on any input or output of any inverter in the chain causes the whole circuit to halt. A chain of inverters used as a delay element in the four-phase asynchronous circuit reduces the circuit performance since the functional (F-phase) and reset (R-phase) phases take the same time. In order to make the four-phase control faster the R-phase must take as little time as possible.

Figure 2 shows a design implementation of the four-phase delay element based on a two-input AND gate which has been designed using the following CSP (Communicating sequential processes) specification:

$$*[\ [In+];\ a+ \parallel b+;\ [+b];\ c+;\ [+a\ \&\ +c];\ Out+;\ [In-];\ a- \parallel b-;\ [-a];\ Out-;\ [-b];\ c-\ ] \qquad (1)$$

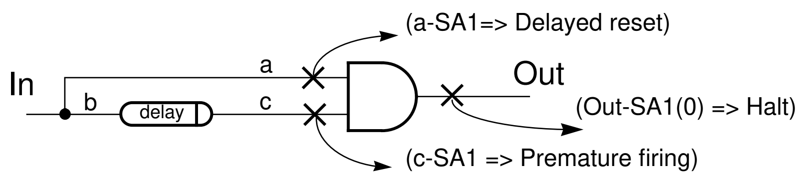

Figure 2 : *Four-phase delay element based on a two-input AND gate*

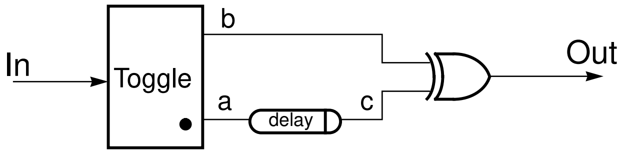

Figure 3 : *Four-phase delay element built on using a toggle element*

where a *handshake expansion* [+*exp*] or [-*exp*] denotes waiting for the Boolean expression (*exp*) or (*exp*) to become true respectively [Mart89].

The effects of some stuck-at faults in the four-phase delay element are shown in Figure 2. Obviously, any stuck-at fault on the *In* input and *Out* output causes the environment to wait forever which is easy to detect. Fault *c*-SA1 causes a premature rising event on the *Out* output. This fault violates the bundle data interface forcing the receiver to accept the output data from the processing logic which may not be stabilised yet. Bundle data violations in micropipelines can be tested only using delay testing techniques described elsewhere [Pet95]. Testing for premature firings in asynchronous four-phase circuits has already been addressed [Haz92, Hulg94]. Another stuck-at fault effect, which can be regarded as a 'harmless' one, is a delayed event. For instance, fault *a*-SA1 blocks the falling signal on input *a* to reset the *Out* output. Instead, the output of the AND gate is reset by a delayed falling event which arrives on its input *c*. As a result, fault *a*-SA1 does not violate the bundled data constraints but reduces the performance of the asynchronous circuit (see Figure 1).

A four-phase delay element can be designed using a toggle element and XOR gate as illustrated in Figure 3. The toggle element steers events alternately to one or other of its outputs when an event appears on its input. The first event is generated on the output marked with a dot [Suth89]. The delay element has two functional phases and two reset phases activated by a rising and falling input event respectively. In the presence of any stuck-at faults on a line of the delay element the sequence of the events is broken which in turn causes the circuit to halt. As a result, the delay element shown in Figure 3 is self-diagnostic.

Summarising the observations of stuck-at fault effects in the delay elements described above we can conclude that stuck-at fault effects in four-phase circuits can be divided into three groups faults which cause the faulty circuit to halt, cause premature firings or delayed events.

3. Circuit symmetry and testability

Definition 1. An asynchronous circuit is called combinational if it does not contain any feedback loops, otherwise it is sequential [Haz92].

Definition 2. An asynchronous circuit is symmetrical if and only if its specification for rising input events ('plus specification') is symmetrical to the specification for falling input events ('minus specification'), i.e., the internal signals of the circuit change their polarities from the plus to minus specification.

Lemma 1. Two-phase circuits are symmetrical circuits.

In four-phase circuits, the F-phase and R-phase of the specification are plus and minus specifications respectively. The F-phase is meaningful whereas the R-phase is specified to ensure the correct restarting of the F-phase. Hence, four-phase circuits are asymmetric circuits.

Lemma 2. Two-phase asynchronous combinational circuits (ACC) can always be designed using the symmetrical elements. Asynchronous sequential circuits (ASC) cannot always be implemented using only symmetrical elements.

Consequence of Lemma 2. ACCs can always be designed with the self-diagnostic property for stuck-at faults.

Since four-phase circuits with different F-phase and R-phases are asymmetrical circuits they cannot be designed using only symmetrical elements. As a result, four-phase circuits are not self-diagnostic for stuck-at faults.

4. Designing for testability of four-phase circuits

As was observed in the previous section two-phase circuits are symmetrical and can always be designed using symmetrical elements. As a result, they are self-diagnostic for stuck-at faults. Note that the building blocks of two-phase circuits [Suth89] are considered as atomic elements, i.e., stuck-at faults are considered only on inputs and outputs of these blocks.

The idea of designing testable four-phase circuits lies in changing the internal circuit interface and its behaviour preserving its external four-phase interface. The original four-phase specification of the circuit is extended making its both F-phase and R-phase symmetrical. The toggle element is used to convert the four-phase signalling into the two-phase signalling protocol and the XOR gate converts two-phase signals into four-phase ones.

Figure 4 shows a general structure of the testable four-phase circuit with *m* inputs and *n* outputs. The F-circuit and R-circuit are two-phase circuits which are activated during the F-phase and R-phase respectively. Sets of toggles and XOR gates are considered as conversion blocks from four-phase into two-phase and from two-phase into four-phase signalling respectively.

The design for testability algorithm of four-phase circuits can be described as follows:

1. Split the four-phase specification of the circuit into the F-phase and R-phase.
2. Make the F-phase and R-phase fully symmetrical by changing the polarity of their internal signals.
3. Add sets of toggles and XOR gates as shown in Figure 4.
4. Implement the F-circuit and R-circuit using symmetrical elements.
5. End.

Lemma 3. The F-circuit and R-circuit in the design shown in Figure 4 are two-phase ACCs.

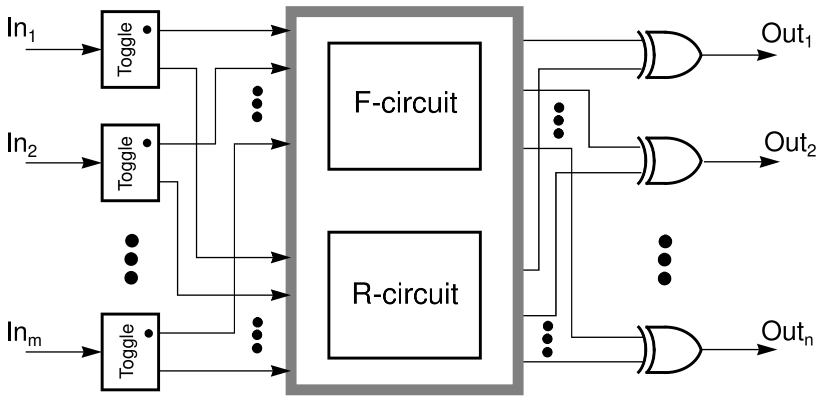

Figure 4 : *Design of a testable four-phase circuit*

Consequence of Lemma 3. The F-circuit and R-circuit are self-diagnostic for stuck-at faults.

The design for testability algorithm described in this section has advantages and disadvantages.

Advantages.

- The design for testability algorithm is straightforward. It does not require any testability analysis to be carried out in order to prove the testability of the circuit.

- The circuit designed using the algorithm proposed is 100% testable for stuck-at faults due to the self-diagnostic property.

Disadvantages.

- The design implementation of the testable circuit requires additional asynchronous blocks such as toggles and XOR gates to be incorporated into the original four-phase design. As a result, the actual implementation of the testable circuit requires more silicon area and reduces its performance.

5. Conclusions

It was shown that two-phase circuits are self-diagnostic circuits for stuck-at faults. Stuck-at fault effects in four-phase circuits have been extended and they can be classified as faults which cause the faulty circuit to halt, generate premature events (premature firings) or delayed events (delayed firings).

A new approach to design four-phase circuits with the self-diagnostic property has been described. The design for testability algorithm is simple and it is based on the extension of the original four-phase specification of the circuit making its F-phase and R-phase symmetrical. The major disadvantage of the proposed technique is a performance degradation of the testable circuit due to sets of toggle elements and XOR gates incorporated in the original design. Our design for testability approach can be used to design four-phase control circuits in asynchronous VLSI systems where their performance and testability can be traded off.

References

[Birt95] G. Birtwistle, A. Davis (Eds), “Asynchronous digital circuit design”, Springer, 1995.

[Dav90] I. David, R. Ginosar, M. Yoeli, “Self-timed is self-diagnostic”, TR-UT-84112, Department of Computer Science, University of Utah, Salt Lake City, UT, USA, 1990.

[Furb94] S. B. Furber, P. Day, J. D. Garside, N. C. Paver, J. V. Woods, “AMULET1: A micropipelined ARM”, CompCon’94, San Francisco, March 1994.

[Furb97] S. B. Furber, J. D. Garside, S. Temple, J. Liu, P. Day, N. C. Paver, “AMULET2e: An asynchronous embedded controller”, submitted to Int. Symposium on Advanced Research in Asynchronous Circuits and Systems (Async97), Netherlands, 1997.

[Haz92] P. Hazewindus, “Testing delay-insensitive circuits”, Ph.D. thesis, Caltech-CS-TR-92-14, California Institute of Technology, 1992.

[Hulg94] H. Hulgaard, S. M. Burns, G. Borriello, “Testing asynchronous circuits: A survey”, TR-FR-35, Department of Computer Science, University of Washington, Seattle, WA, USA, 1994.

[Pet95] O.A.Petlin, S.B.Furber, “Scan testing of micropipelines”, Proc. 13th IEEE VLSI Test Symposium, Princeton, New Jersey, USA, May 1995, pp. 296-301.

[Russ89] G. Russell, I. L. Sayers, “Advanced simulation and test methodologies for VLSI design”, Van Nostrand Reinhold (International), 1989.

[Suth89] I. E. Sutherland, “Micropipelines”, Communications of the ACM, Vol 32, No. 6, Jan., 1989, pp. 720-738.