[2] T.-A. Chu, *Synthesis of Self-Timed VLSI Circuits from Graph-theoretic Specifications*, PhD thesis, Massachusetts Institute of Technology, 1987.

[3] I. David, R. Ginosar, and M. Yoeli, "Self-Timed is Self-Diagnostic", Technical report, University of Utah, 1990.

[4] D. L. Dill and E. M. Clarke, "Automatic Verification of Asynchronous Control Circuits Using Temporal Logic", *1985 Proceedings at Chapel Hill Conference on VLSI*, pp. 127–143, 1985.

[5] A. J. Martin, "The Limitations to Delay-Insensitivity in Asynchronous Circuits", *Proceedings of the Sixth MIT Conference in Advanced Research in VLSI*, pp. 263–278, 1990.

[6] E. J. McCluskey, *Logic Design Principles with Emphasis on Testable Semicustom Circuits*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

[7] T. H. Y. Meng, R. W. Brodersen, and D. G. Messershmitt, "Asynchronous Design for Programmable Digital Signal Processors", to appear in *IEEE Transactions on ASSP,* April 1991.

[8] T. H. Y. Meng, R. W. Brodersen, and D. G. Messershmitt, "Automatic Synthesis of Asynchronous Circuits from High-Level Specifications", *IEEE Transactions on Computer-Aided Design*, 8(11):1185–1205, November 1989.

[9] R. E. Miller, *Switching Theory, Volume II: Sequential Circuits and Machines*, Wiley, New York, 1965.

[10] D. E. Muller and W. S. Bartky, "A Theory of Asynchronous Circuits", *Proceedings of an International Symposium of the Theory of Switching*, pp. 204–243, 1959.

[11] A. V. Yakovlev, *private communications*, June 1990, Alexandre Yakovlev is a professor at the Leningrad Electrical Engineering Institute.

Semi-Modularity Checker Results				
Circuit	No. of Reachable States	No. of State Transitions Checked	Semi-Modular	Run-Time
Pipeline Handshake	31	53	Yes	0.0 sec
Memory Interface 1	518	1476	Yes	0.3 sec
Memory Interface 2	N/A	980	No	0.2 sec
Demultiplexer	1287	4867	Yes	1.1 sec
Multiplexer	1008	3455	Yes	0.7 sec

Table 1: Semi-modularity checker gives number of reachables state and run-time statistics.

semi-modularity for non-deterministic circuits and proved that semi-modular non-deterministic circuits are fully testable for SAFs, but may not be self-diagnostic. We have also shown that in a SOP implementation, any two AND gates cannot simultaneously have output 1 and that semi-modular control circuits must be irredundant. This information aids in the development of hazard-free circuit synthesis at the gate-level, which is currently under investigation in our research group.

In addition we presented many example circuits and the results from an automated semi-modularity checker. While the only theoretical upper bound on the algorithm is exponential, the examples have indicated that the expected run-time is much less than the worst case and the program completes in under 2 seconds on all the circuits we have encountered so far.

Acknowledgment

This research was supported by an ONR Grant no. N00014-89-J-3036 and a Seed Grant from the Center for Integrated Systems, Stanford University. The authors would like to thank A. Hung and T. Castaneda for many valuable discussions on the properties of semi-modular circuits and all the people who were interested in this work and contributed their intelligence and experience to help clarify the issues of testability and hazard-free designs; to name a few: Prof. D. Dill and S. Nowick at Stanford University, Prof. A. Martin and S. Burns at the California Institute of Technology, and Prof. A. Yakovlev at the Leningrad Electrical Engineering Institute.

References

[1] S. Burns, *private communications*, June 1990. Steve Burns is a graduate student at CalTech University.

which each conditional is allowed to change, and the reset states for both values of each conditional. The output is a list of reachable states and a conclusion of whether the circuit is semi-modular or not.

5.1 Algorithm

We assess the semi-modularity of a circuit by breaking a circuit down into its reachable state transitions. A state transition is semi-modular if every signal in a state which was enabled and does not change in moving to the next state is also enabled in the new state. If all reachable state transitions are semi-modular then the circuit is semi-modular. The program recursively checks all reachable state transitions to determine if any transition breaks semi-modularity. The program terminates once it finds a non-semi-modular transition, or once all transitions are analyzed.

Conditionals are handled in two steps: 1) the circuit is checked with each conditional fixed at both values, and 2) conditional state transition bridges are checked to ensure that semi-modularity is preserved.

The circuit state is coded as a bit vector, at the moment realized by a 32 bit integer. Each bit represents a different node in the circuit. In addition to the state vector, there exists an *enable vector*, which codes the nodes that are enabled in the corresponding state. The visited states are stored in a binary tree to ensure that no state is examined twice. While 32 nodes represent a maximum number of nodes in the circuit, it is not difficult to expand to a multiple of 32 by concatenating multiple integers.

5.2 Circuit Examples

The semi-modular checker has been applied to some control circuits commonly used in digital design. Table 1 presents the performance figures of the run-time efficiency of the semi-modularity checker provided on a DECstation 3100.

On the third example circuit in Table 1, the memory interface 2, we purposely deleted a signal line and tested it for semi-modularity. The program complained that the circuit was not a semi-modular implementation and indicated that the output node of the signal line contained a hazard.

In many cases during the design process, we are unsure of the redundancy of some feedback connections. With the semi-modularity checker we can check altered designs very rapidly and use the results to determine what change in the design is needed to provide for a hazard-free circuit.

6 Conclusions

We have proved that semi-modularity is a necessary and sufficient condition for hazard-free asynchronous control circuits. Furthermore, we have proved that semi-modularity insures the testability of such circuits. We have also defined

if they are of the same value and either both are enabled or both are not enabled. A state bridge is semi-modular if all corresponding signals in the two bridge states, except the conditional signal, have the same status. Within a sub-graph the status of the input conditional signal is constant.

Theorem 4.1 *If the individual sub-graphs of a live asynchronous circuit are semi-modular and the state bridges are also semi-modular, then the circuit is hazard-free.*

Proof: A circuit is hazard-free if all state transitions in its expanded state diagram satisfy the conditions of semi-modularity. The circuit expanded state transitions can be decomposed into state transitions within a state sub-graph and conditional signal changes. Since the state sub-graphs are semi-modular, transitions within each sub-graph are semi-modular. The bridges represent two way connections between states where all signals have the same status; thus both transitions preserve semi-modularity. Therefore all transitions in the circuit are semi-modular and the circuit is hazard-free. ∎

Because of the non-deterministic input conditional, the circuit is no longer self-diagnostic: a SAF at the input conditional node will not cause the circuit to halt, but will force the circuit to stay in a particular mode. However, we can prove that the circuit is still fully testable for SAFs.

Theorem 4.2 *A live semi-modular asynchronous control circuit with distinguishable modes of operation is fully testable for any SAF.*

Proof: A SAF can either be on the signal representing the conditional or on some other signal. Since the sub-graphs are semi-modular and the circuit is live, then by the same argument used in Lemmas 3.2 and 3.3 all signal transitions must be in one of the state sub-graphs. Hence, if the SAF is on a non-conditional signal, there exists a mode of operation where the circuit will halt. Since the modes can be controllably entered by changing the conditional, the fault can be detected. If the SAF is on the conditional signal, then the circuit will be stuck in a particular mode. Since the modes are distinguishable, when the user attempts to move to a different mode, the input/output behavior will be different and the SAF will be detected.

Additional SAFs add no complexity to the testability issue, since one SAF make a state graph acyclic, multiple SAFs will also. Consequently, multiple SAFs will be detectable in conditional asynchronous control circuits as well. ∎

5 Tool to test for Semi-Modularity

A CAD tool written in C has been developed which tests for semi-modularity of a circuit implementation. The input is a gate-level description of the complete circuit, including the gate model of the environment and the reset state. Conditionals can be handled by specifying each conditional signal's name, the states in

no memory element. Its invalid ESG contains a cycle which does not contain all signal transitions. In fact, there exists a SAF which forces the circuit into an invalid state in which this cycle of states repeats. We conclude that to obtain a self-diagnostic circuit, it is necessary to rely on an additional assumption. We chose this assumption to be the structural property of the asynchronous control circuits; i.e., every loop contains a C-element.

We can now use the above two two lemmas to prove that semi-modular control circuits are self-diagnostic.

Theorem 3.6 *Under the SAF model, if any fault occurs in a live speed-independent semi-modular circuit, the circuit will halt; no cycle of signal transitions will be possible.*

Proof: A SAF by definition prohibits a signal transition. From the previous two lemmas, each cycle in the circuit contains all signal transitions. Hence, with the inclusion of the SAF the valid and invalid ESG become acyclic. Therefore, the circuit will stop.

Additional SAFs add no complexity to the testability issue, since one SAF make the state graph acyclic, multiple SAFs will also. Hence, deterministic asynchronous control circuits are self-diagnostic under single and multiple SAFs. ■

Thus, semi-modularity is a sufficient condition for a control circuit to be self-diagnostic. From the previous theorems, we can conclude that hazard-free speed-independent circuits are self-diagnostic. Designing for testability, therefore, requires no additional effort than designing for hazard-free circuits under arbitrary gate delay assumptions. The actual testing of these circuits for SAFs is minimal: test one cycle of input transitions according to the STG specification; if the outputs behave as predicted, the circuit is SAF-free.

4 Control Circuits with Conditionals

Consider the circuits with input conditionals modeled by free-choice Petri-nets, or STGs with free-choice places [2, 8]. There can be more than one conditional variable, for example in a multiplexer selecting one of many inputs to be transferred; however, we can use one conditional variable in our discussion without loss of generality. The value of the conditional determines the *mode* of a control circuit, such as a read mode or a write mode in a bus controller or memory interface. Each mode will have a different ring of partial orderings to which the modeled signals must abide.

Each mode of operation can be represented by a sub-graph in the expanded state graph with the conditional fixed at a particular value. Two modes are *distinguishable* if the partial ordering of output/input signals is different in the different modes. For those states where an input conditional is allowed to change (but is not required to change), the input conditional represents a set of *state bridges* between two states in the sub-graphs. Two signals have the same *status*

from starting in its reset state or if it made the circuit start in a different valid state, that the circuit would stop. However, because the SAF may prohibit the circuit from starting in an valid state and may force the circuit into an invalid state we first must prove the corresponding Lemma 3.2 for the invalid ESG.

Lemma 3.3 *In a live semi-modular control circuit, all cycles of invalid states contain all signal transitions.*

Proof: *(By Contradiction.)* Assume a cycle of states, C, which does not contain all possible signal transitions exists starting in an invalid state. Call the set of missing signal transitions in C, M.

By Lemma 3.2 every cycle in the valid ESG contains all signals. Consequently, every signal transition (or set of signal transitions) must be necessary for the enabling of some other signal. Hence, some signal transition (or set of signal transitions) in M must be necessary for some signal transition not in M, denoted as x. Since, M is not in the invalid cycle C and x is in C, x must be enabled by some signal y also in C which does not enable x in the valid ESG. We state y *misenables* x.

For y to misenable x the circuit implementation is such that y can enable x. In addition, because y does not enable x in the valid ESG, there must not exist a valid state where y has fired and x is not already enabled. Because of semi-modularity and speed-independence, we conclude that in the valid ESG x must be enabled before y can fire. Hence, for y to misenable x, y itself must have been misenabled.

This misenabling argument can be used iteratively, moving the cause of the first misenabling backwards. Because a cycle exists in the invalid ESG this backward traversal must converge upon itself, forming a loop. Because every feedback loop in the circuit contains a C-element, the misenabling argument must include the misenabling of a C-element. However, a C-element cannot be misenabled, it has only one enabling condition for both the rising and falling transitions of its output. Hence, the backward traversal of misenabling cannot form a loop. A contradiction. Therefore, all cycles of invalid states contains all signal transitions. ∎

The above lemma relies on the structure of asynchronous control circuits in that any loop in the circuit contains a memory element, in particular, a C-element. The proof is equally valid for different memory elements which have only one method of enabling both the upward and downward transition of its output. In fact as long as every loop in the circuit contains an element which contains only one method of enabling, the proof argument is valid.

An alternative proof, which relies on a restriction that the gates in the implementation be *simple gates*, that is, where each output transition is enabled by either a sum of signal levels or a product of signal levels but not a sum of products, is omitted due to space constraints.

As an aside, we state without proof that we have discovered an example semi-modular circuit which contains non-simple gates and a feedback loop with

it follows that if a node is not enabled in one cycle of signals, then it must not be enabled in every cycle of the same signals. We can now prove that every cycle in the valid ESG will contain all signal transitions.

Lemma 3.2 *Given a live semi-modular asynchronous control circuit, all cycles of states in the valid ESG must contain all signal transitions.*

Proof: *(By Contradiction.)* Assume there exists a cycle of transitions, Y, in which not all signal transitions occurred. Let the set of signals which cycle in Y be called $\mathcal{C}$, and the set of signals which do not cycle in Y be $\mathcal{S}$.

The proof will be in 3 steps:
1) Showing that some signal in S must be enabled during some state in Y.
2) Showing that while Y cycles, all signal transitions of signals in S can occur.
3) Proving the original theorem by contradiction using 2) and the fact that the circuit is fully connected.

We first prove 1) by contradiction.

Assume no signal in $\mathcal{S}$ is enabled during the cycle Y. Then, by Lemma 3.1 no possible orderings of the signals in $\mathcal{C}$ will cause a signal in $\mathcal{S}$ to be enabled. Consequently, once cycle Y was entered, the nodes in $\mathcal{S}$ could never again be enabled, violating the liveness property of the circuit. Hence, at least one signal, $v \in \mathcal{S}$ must be enabled at some point in Y. Since v does not change in any state in Y, to preserve semi-modularity, v must remain enabled throughout Y.

We prove statement 2) through repeated use of a thought experiment in which the enabled node in S found in 1) fires.

Let v fire.

At this point either the signals in $\mathcal{C}$ must stop cycling or continue cycling. If they stop, then there must be a signal transition t in $\mathcal{C}$, which is prevented by v. However, because the circuit is speed-independent, there exists a set of delays such that v fires after t is enabled but before t has fired, consequently disabling t and violating semi-modularity. Hence, the signals in $\mathcal{C}$ must continue cycling; v can not prevent a transition of a signal in $\mathcal{C}$.

At this point, we have the same situation as we did before v fired; the signals in S are not cycling and the signals in C are cycling. Therefore, by 1) a new signal in S must now be enabled.

Because the circuit is live, if we repeatedly fired the newly enabled signal, indefinitely, every possible transition of the signals in $\mathcal{S}$ must fire. For the same reasons v did not cause the cycle Y to stop, neither will any of the other signal transitions in $\mathcal{S}$. Hence, no signal transition of a signal in S can prevent a signal transition in Y.

We now complete the proof by reaching a contradiction.

Because the circuit is fully connected, there exists a path of gates which connects a signal in $\mathcal{S}$ to some signal in $\mathcal{C}$. Hence, some transition of a signal in $\mathcal{S}$ must prevent some signal transition in Y. This contradicts 2). ∎

The above theorem proves that all cycles in the valid ESG contain all signal transitions. At this point we could prove that if a SAF did not prevent the circuit

from which the gate-level implementation has been derived. Because the Boolean equations satisfy the STG, we conclude that the circuit behavior will be identical to the behavior specified by the STG. This represents a speed-independent hazard-free circuit. ∎

These two theorems imply that a hazard-free circuit must be semi-modular, and, if the derivation of the gate-level implementation is error free, *vise versa*. Hazards were traditionally classified as races, static hazards, dynamic hazards, etc., but all of these anomalies have the same cause: the existence of two sets of gate delays, one in which a node would fire and one in which it would not fire. In the case where the node did not fire, the signal transition was first enabled but was then disabled before it fired. The state graph of this circuit thus violates semi-modularity. When viewed in this light, the condition for hazard-free circuits is clearly semi-modularity.

3.5 Semi-Modularity and Self-Diagnostic Circuits

In this subsection, we first present three lemmas will be used to relate semi-modularity to self-diagnosis and then use these lemmas to prove that a semi-modular circuit is self-diagnostic. The first lemma proves that in a live semi-modular circuit changing the ordering of signal transitions cannot alter which other signals will be enabled. The second lemma proves that the valid ESG contains only cycles of states which contain all signal transitions; i.e, every cycle contains a transition of every signal in the circuit, in both rising and falling forms. The third lemma proves that if there exists a cycle in the invalid ESG, then, it too must contain all signal transitions.

Lemma 3.1 *Given a speed-independent live semi-modular asynchronous control circuit and a state s, if a signal, e, is enabled but does not fire, in the course of one sequence of signal transitions which starts in state s, then e must be enabled at the completion of all valid permutations of the signal transitions which start in state s.*

Proof: *(By Contradiction)* Assume there exists a valid permutation of signals such that e is not enabled at its completion; call this permutation P. For semi-modularity to hold, e must have never been enabled during the sequence of signal transitions. Call the state in which the circuit is in at the completion of the signal transitions q. In state q, e must not be enabled. However, we are given that in the course of the original ordering e is enabled but does not fire. Hence to preserve semi-modularity of this signal transition sequence, in the state in which the sequence leaves the circuit, e must be enabled. However, this state must also be q. Because e cannot be enabled and not enabled in the same state we have a contradiction. ∎

The above theorem is general in that it refers to any sequence of signal transitions. By examining the sequence of signal transitions that form a cycle of states, we can conclude that if a node is enabled in one cycle of signals (but does not fire), then it must be enabled in every cycle of the same signals. Furthermore,

Proof: Assume a hazard exists only at internal node but does not affect the modeled signals. Then there exists a set of gate delays such that an enabled node is disabled (the enabling inputs changed before the node had a chance to fire.) Because the circuit is live and irredundant this node must later become observable. By the given assumption and the definition of observability, we know that when the node becomes observable, the node's value must be as expected and unique. Therefore, under this set of gate delays, either the node does not fire and is not expected to fire, or it fires and was expected to fire, or the node exhibits a voltage spike. By the definition of observability, the time until the node becomes observable is independent of the node's value. Therefore, we can alter particular instances of the node's rise/fall time to either prevent the node from firing or force it to fire, without changing the time until the node becomes observable. We can thus arbitrarily fix the node's value at the time the node becomes observable. Hence, by altering the gate delays to fix the node to the unexpected value at the time it becomes observable, we have determined a set of gate delays which presents a hazard at the modeled signals. ∎

For example, suppose an internal node has a glitch. Given some specific delays in gates, this glitch is unobservable for some time. However, we can arbitrarily make either the node's rise time or fall time longer such that the node ends up in the "wrong" state (a 1 when a 0 is expected or *vice versa*) when the node becomes observable. Since the node is then observable, only one state can lead to the expected behavior of the modeled signals. Thus, there always exists a set of gate delays where the modeled signals will behave differently than expected: a hazard at the modeled signals.

3.4 Semi-Modularity and Hazard-Free Designs

We now present two theorems which relate semi-modularity and hazard-freedom.

Theorem 3.4 *An irredundant circuit which is hazard-free must be semi-modular.*

Proof: *(By contradiction)* Assume the circuit is not semi-modular. Then there exists a node i and a pair of states a and b such that $a\mathcal{R}b$ and node i is enabled in state a but not in state b: node i may or may not fire when the circuit enters state b, and thus may exhibit a glitch on signal i. This represents an internal hazard and, according to Theorem 3.3, there exists a hazard at the modeled signals. A contradiction. ∎

Theorem 3.5 *A semi-modular circuit whose Boolean equations satisfy its STG specification and whose gate-level implementation satisfies its Boolean equations is a speed-independent hazard-free circuit.*

Proof: According to the definition of semi-modularity, once an output of a gate is enabled, the node will fire independently of the delay in the gate. Since this is true for all gates in the circuit implementation, the circuit will behave independently of gate delays. This behavior must satisfy the Boolean equations

can be made and the circuit is in deadlock, violating the liveness assumption. A contradiction. ∎

As a result of this theorem, either additional inputs must be added to the AND gates to prevent two AND gates from being one at the same time or additional wire forks must be used to provide a mechanism to observe the output of all the AND gates. Interestingly, this theorem implies that in a live SOP an AND gate may require more inputs than its Boolean equation does. A larger number of inputs will slow the SOP down and require more circuit area. Hence there exists a tradeoff between speed/area and testability/hazard-free designs.

3.2 Semi-Modularity and Irredundant Circuits

A redundant combinational logic block contains a node which if removed does not change the Boolean equations which the combinational logic block satisfies [6]. It is easy to extend Theorem 3.1 to show that in a live semi-modular circuit all combinational logic blocks are irredundant:

Theorem 3.2 *A live semi-modular asynchronous control circuit must be irredundant.*

Proof:*(By contradiction)* By definition, the firing of a redundant node cannot be verified at the feedback signals, and hence, cannot be enabled more than once because of semi-modularity. Therefore, the redundant combinational logic block is not live. ∎

Thus, all combinational logic blocks in live semi-modular circuit must not contain any redundant nodes.

3.3 Hazards and Semi-Modular Circuits

We define a *hazard-free* circuit as one in which all modeled signals behave as predicted by the corresponding STG or reduced state graph independent of the relative delays on the individual gates. A hazard can occur in the form of an unexpected voltage spike, sometimes referred to as a *glitch*, or in a situation where a signal transition may or may not fire depending on the relative delays among gates. Given the assumptions of speed-independence, in which the delays on gates are unknown and unfixed, a hazard represents the existence of a set of gate delays which will cause an unpredicted behavior.

We now prove a theorem which states that a hazard at any internal node implies a hazard at a modeled node.

Theorem 3.3 *In a live speed-independent asynchronous circuit if an internal node has a hazard, then there exists a set of delays such that the hazard will show up in the modeled signal; i.e., the circuit has a hazard at the modeled signals.*

We now define semi-modularity [10]:

Definition 3.1 *(Semi-modularity) A state transition $a \rightarrow b$ is semi-modular if for every node i which is enabled in state a which does not change to its enabled value a_i', when the circuit goes to state b, is enabled in state b and to the same value as in state a. A circuit is semi-modular with respect to state r if every pair of states a, b satisfying $r \mathcal{F} a$ and $a \mathcal{R} b$ has $b_i' = a_i'$ for each node i satisfying $b_i = a_i \neq a_i'$.*

In this discussion, r is considered the reset state of the machine. By this definition, if the circuit is initialized to state r, and every reachable transitions satisfies the above definition, the circuit is semi-modular. For convenience, we will assume the existence of the reset state when we refer to a circuit as semi-modular. Hence, we need only examine the state transitions in the valid ESG to determine if the circuit is semi-modular.

Semi-modularity ensures that an enabled node is ensured to fire before it can be disabled by other transitions. In other words, before the inputs to an enabled gate can change, the firing of that node must be verified at the feedback signals. If this were not the case, then we can always extend the gates rise/fall time such that the inputs changed disabling the node, and, as a result, violate semi-modularity.

We caution the reader that the state transitions in the invalid ESG may not be semi-modular because these states are not reachable from r. One situation which must be analyzed in determining if a circuit is self-diagnostic is if the SAF prevents the circuit from being initialized to state r.

The first theorem we prove concerns the synthesis of hazard-free asynchronous circuits using sum-of-product forms. A similar result can be derived from work in delay-insensitive circuits (circuits which work correctly without the existence of isochronic forks) [5]. However, the class of delay-insensitive circuits has been shown to be very small [5] and hence extending this theory to speed-independent circuits is valuable.

Theorem 3.1 *In a sum-of-product (SOP) combinational logic block within a live semi-modular circuit, the outputs of two AND gates can never be 1 at the same time.*

Proof: *(By contradiction)* Assume two ANDs were on (i.e., the outputs were 1) at the same time. Then, initially there must have been one AND on while the second AND is enabled high. Because one AND gate is on and thus the output of the SOP network is already 1, the firing of the second AND gate cannot be verified. Because of semi-modularity, the second AND gate cannot be enabled low until its firing high has been verified. For this firing high to be verified the first AND gate must first be verified low. To this end, assume the first gate is enabled low. This downward transition, however, cannot be verified until the second AND gate is verified to be low. This represents a stalemate. No progress

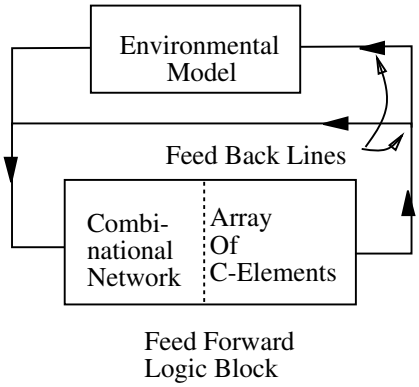

Figure 1: Asynchronous control circuit model.

We define an internal signal to be *observable* if and only if the values of one or more of the modeled signal depend on its value. We define a signal transition to be *enabled* in a state when the signal is expected to change values. For example, if all the inputs to an AND gate are 1 and the output signal is 0, then we say that the output is enabled low. A signal transition *fires* when the signal changes values. The firing of the transition is *verified* when the new signal value is observable on at least one of the feedback signals. In other words, if a transition y on one of the feedback signals cannot fire until transition x has fired, then when y fires, transition x has been verified. A circuit is *live* if every transition can be enabled infinitely often. We define a *cycle* of transitions as a sequence of signal transitions in which the start and the end of the sequence is the same state. An *irredundant* asynchronous control circuit is a circuit in which all combinational logic blocks within that circuit have no nodes which can be removed without violating the specification of the circuit. We will show that a hazard-free implementation requires that the asynchronous control circuit be irredundant.

3 Semi-Modularity and Deterministic Control Circuits

In this section we address semi-modularity and the properties of semi-modular deterministic control circuits with no input conditionals.

3.1 Semi-Modularity and Automated Synthesis

To facilitate the discussion of the relationship of semi-modularity and automated synthesis we define two relationships between states [10]: the $\mathcal{R}$ function and the $\mathcal{F}$ function. The $\mathcal{R}$ function represents a relation between two states: $a\mathcal{R}b$ expresses that state b may follow state a with no intervening states. The $\mathcal{F}$ function represents a relation between two states like the $\mathcal{R}$ relation but where intervening states are allowed: $a\mathcal{F}b$ expresses that state b may follow state a with any number of intervening states.

a VME-Bus controller.

In this paper, we use the notation of signal transitions for specifying circuit behavior, which is described by a set of partial orderings of signal transitions in a signal transition graph (STG). The signal transitions always cycle continuously according to the partial orderings in the STG, which is assumed to be connected in our discussion.

The expected circuit behavior of an asynchronous control circuit can also be described in a *reduced state graph* derived from a STG [2, 8]. In a reduced state graph, each node represents an input/output state of the circuit, coded as a bit vector of the input and output signals (the modeled signals). Each arc represents a potential state transition and is labeled with a corresponding signal transition representing the modeled signal which changes.

From the state graph the Karnough map can be derived for each output signal and thus its Boolean expression. From the Boolean equations, a circuit implementation can be developed with the available gates in the design library. After a gate-level implementation is obtained (which has to satisfy the Boolean equations of course), the circuit need be tested for hazards. If it has a hazard, the circuit must then be altered until a hazard-free implementation is found.

The implementation of a circuit usually requires many more internal nodes than just the input and output nodes modeled by the corresponding STG, depending on the individual building blocks available to the designer. Muller and Bartky [10] analyze the characteristics of an implementation by representing the state of an asynchronous circuit by a bit vector. Each gate output (a node) of the circuit is represented by a bit and the bit vector collectively describes the *expanded state* of the circuit. In this paper, subscripts are used to denote the node of a state and primes are used to denote the enabled state (a transition to the next state) of a node. For example a_i represents the value of the i^{th} node in the circuit, and a_i' represents the enabled state of that node. If node i is not enabled in state a then $a_i = a_i'$. We refer to this type of detailed state graph as an *expanded state graph (ESG)* [9]. Each circuit implementation defines two ESGs. The first ESG contains all states reachable from a given reset state r. It is referred to as the *valid ESG*. The other ESG contains all *invalid* states, those states not reachable from r. We refer to this second ESG as the *invalid ESG*.

The structure of a deterministic asynchronous control circuit can be broken up into two components, the feedforward combinational logic block and an array of C-elements which control the feedback signals, as shown in Figure 1. Modeled input/output signals by definition are feedback signals but there may be other internal signals which are also feedback signals.

C-elements are memory elements which have two inputs A and B and one output C. The next state equation for a C-element is $C' = AB + AC + BC$, where C is the current output and C' is the next output. The operation of the C-element can be described as follows: if the output of the C-element is a 1, it will remain 1 until both inputs move to 0, at which time it will be enabled 0; if the output of the C-element is a 0 it will remain 0 until both inputs move to 1, at which time it will be enabled 1.

circuits operating on control signals not synchronized to a global clock. In asynchronous control circuits the inputs usually have restricted transitions. The set of constraints which are put on the inputs are referred to as the circuit's *environmental constraints.* The circuit and the circuit's environment are modeled together in what is termed a *complete circuit* [9]. The inputs to the circuit are modeled as functions of the outputs. Speed-independent asynchronous control circuits are intended to operate correctly assuming arbitrary gate delays but negligible wire delays *within* a circuit module. We model the delay in a gate using the *transparent delay model*, in which a logic gate functions as an instantaneous Boolean evaluator followed by an arbitrarily long delay element. This model implies that wire forks, termed *isochronic forks*, provide simultaneous transmission of values to each terminal [5].

Semi-modular circuits (defined later in section 3) are a subclass of speed-independent asynchronous circuits which bear an association between the states of the circuit and a semi-modular lattice [10, 9]. We show that semi-modularity is the necessary and sufficient condition for an asynchronous circuit to be hazard-free, and that this same condition also ensures that the circuit is self-diagnostic regardless of the circuit style. While this idea may not be new [9, 11, 1], to the best of our knowledge it is the first time that such claims were proven in the literature.

An introduction to asynchronous control circuits and a description of our design algorithm will be given in section 2. In section 3, we define semi-modularity and circuit hazards and describe their relationship in asynchronous circuits. We prove that under the transparent delay model, semi-modularity is a necessary condition for a speed-independent circuit to be hazard-free. In addition, we show that if the design algorithm is error free, semi-modularity is a sufficient condition for the circuit to be hazard-free. Furthermore, we show that any SAF (single or multiple) would halt a semi-modular control circuit and thus all hazard-free control circuits are self-diagnostic. Then, in section 4, we define semi-modularity for non-deterministic asynchronous control circuits and prove that hazard-free non-deterministic circuits are also semi-modular and consequently fully testable for SAFs, but may not be self-diagnostic. We describe a tool designed to test for semi-modularity and present circuit examples in section 5. We give our conclusions in section 6.

2 Asynchronous Control Circuits

Asynchronous control circuits manage communication of data using handshaking signals between concurrent communicating processes; they are sometimes referred to as *interconnection circuits*. Asynchronous control circuits can be divided into two types: circuits with no input conditionals (a completely deterministic behavior), and circuits with input conditionals (a non-deterministic behavior based on the environment) [8, 2]. Input conditionals represent an environmental choice; for example, the user can select between a read/write cycle in

Semi-Modularity and Self-Diagnostic Asynchronous Control Circuits

Peter A. Beerel
Teresa H.-Y. Meng
Computer Systems Laboratory
Stanford University
Stanford, CA 94305

Abstract

Semi-modularity is a property of asynchronous circuits which bear an association between the states of the circuit and a semi-modular lattice. A circuit that does not require testing for stuck-at-faults (SAFs) is called self-diagnostic: any SAF in a self-diagnostic circuit halts the operation of the circuit. In this paper, testability of asynchronous circuits is related to the property of semi-modularity, and speed-independent circuits are proven to be fully testable for SAFs if their implementation do not contain any circuit hazards. We show that deterministic hazard-free asynchronous circuits are semi-modular and self-diagnostic, and that non-deterministic hazard-free asynchronous circuits are fully-testable for SAFs but not self-diagnostic.

1 Introduction

As VLSI circuits become smaller and smaller, problems with synchronous designs have made people reevaluate the decision to work totally in the discrete time domain. Asynchronous circuits avoid synchronous design problems such as clock distribution and skew. With new automated design techniques, asynchronous circuits have the promise of easier design and respectable speeds [8].

The issue of testability of asynchronous circuits has only been addressed recently. One method of synthesis for testability is the use of *self-diagnostic* circuits. A *self-diagnostic* circuit does not require testing for stuck-at-faults (SAFs): any SAF in a self-diagnostic circuit halts operation of the circuit. It was demonstrated that the use of dual-rail code of a ternary network, in which each input and output is an element of $\{0, 1, U\}$, leads to a self-diagnostic circuit [3]. This technique needs extra circuitry to maintain dual-rail inputs. Scan-path techniques have also been proposed for testing asynchronous circuits. However, these techniques incur long testing time because the test input and output have to be sequentially scanned into and out of each memory element. Another disadvantage to scan-path techniques is that the added circuitry must be designed carefully to maintain the hazard-free property in an asynchronous design.

This paper deals with *asynchronous control circuits*, which are sequential