Newtonian Arbiters Cannot be Proven Correct*

Michael Mendler†
Institute for Computer-Aided Circuit Design
University of Erlangen, FRG.

Terry Stroup‡
Laboratory for Foundations of Computer Science
University of Edinburgh, UK

Abstract

Computing hardware is designed by refining an abstract specification through various lower levels of abstraction to arrive at a transistor layout implemented in a physical medium. Formalizing the refinements — one task of the mathematical semantics of computation — involves proving that the device described at each level of abstraction does indeed behave as prescribed by the description at the next higher level. One obstacle to this goal that has long been recognized is that certain classes of behaviors can be physically realized only approximately. The notorious problem of metastable operation precludes, for example, the realization on classical principles of flipflops that react in bounded time to arbitrary input signals.

The literature suggests that the difficulty lies ultimately in the specification's requiring that the realizing device react properly in bounded time. We show, however, that a simple time-*un*bounded synchronization problem, namely mutual exclusion by means of an arbiter, cannot be solved with perfect reliability using continuous, *i.e.* Newtonian, physical phenomena. In particular, for any physical device operating on Newtonian principles that satisfies specific assumptions concerning an arbiter's input-output behavior there always exist competing requests to which it reacts by granting them all.

1 Introduction

Hardware design proceeds — ignoring false starts and similar matters — by refining numerous levels of abstraction, beginning typically with block diagrams and ending by

*Abridged version to appear in *Proceedings of the* 2^{nd} *IFIP WG10.5/WG10.2 Workshop on Designing Correct Circuits* J. Staunstrup and R. Sharp editors.

†Partially supported by a scholarship from the Stevenson Foundation. Author's address: Wetterkreuz 13, D-8520 Erlangen, email: `mendler@lrs.e-technik.uni-erlangen.de`

‡Supported in part by SERC grant GR/F 38808, "Mathematically Proven Safety Systems" and by the Deutsche Forschungsgemeinschaft, Sonderforschungsbereich 182, "Multiprozessor- und Netzwerkkonfigurationen". Author's address: The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, email: `rts@lfcs.ed.ac.uk`

way of register-transfer and gate networks in a transistor layout that is implemented in a physical medium. This stepwise refinement of an abstract specification into more and more concrete designs has the practical advantage of postponing detailed design decisions until they are appropriate. The theoretical advantage is that a verification of the final implementation's correctness can be factored into a sequence of smaller steps: If at each level of abstraction the design is shown to behave as required by the next higher level, then the final implementation meets its original specification. This technique presupposes rigorous mathematical models for the descriptions at each level of abstraction so that behaviors described at different levels may be compared. The spectrum of models ranges from discrete computational structures appropriate to algorithmic descriptions down to differential equations modeling electrical behavior of transistors. Hence at some design step continuous behaviors are used to implement discrete specifications.

Unfortunately, a series of theoretical and laboratory investigations into the metastable operation of flipflops has shown that a significant class of simple, discrete behaviors possess no continuous realizations in bounded time [CM73], [May74], [CW75], [HE75], [VA75], [Mar77], [FD79], [Cha83]. It turns out that a continuous bistable system invariably possesses a range of inputs, called *marginal triggering*, that drive it into a *metastable* state in which it may linger indefinitely before finally settling into a stable state [Mar81], [KC87]. Though clocked systems routinely avoid marginal triggering by using clock pulses to control the sequencing and duration of other signals, the synchronization of independently clocked systems inevitably involves uncontrolled inputs to a bistable device that performs the synchronized task. The metastability results thus preclude the synchronization in *bounded time* of genuinely asynchronous systems using continuous electrical phenomena.

One of the synchronization tasks affected is the classical problem of excluding simultaneous access to shared resources, widely known as the *mutual exclusion* problem. Software engineers usually solve mutual exclusion problems by assuming that hardware engineers can provide them with a suitable bistable system called a binary semaphore. Now binary semaphores are not difficult to implement in a clocked system. But a hardware element, called an *arbiter*, that ensures mutual exclusion in a fully asynchronous environment by arbitrating between competing demands on shared resources requires careful design to function adequately [SC79], [CR79], [BJ83], [CAV86], [SD88].

The literature occasionally asserts that metastable operation does not prevent the use of bistable elements to solve synchronization problems, so long as the task, like asynchronous arbitration, need not be completed in bounded time [Pěc76], [SD88]. It is however an open question whether such *time-unbounded* solutions are susceptible to other kinds of intermittent failure, perhaps not involving metastable operation. In this paper we show that indeed any continuous device that satisfies certain minimal assumptions concerning the input-output behavior of a time-unbounded arbiter cannot guarantee the crucial mutual-exclusion property. In particular, there always exists under these assumptions a class of conflicting requests that drive the device into a state in which it grants simultaneous access to the shared resource.

Our assumptions are as follows. Requests to use the shared resource can occur at arbitrary times and in particular simultaneously. Any persisting request not in conflict with other requests will eventually be honored; and of conflicting requests that persist, one will, given time to react, indeed be honored. Any request that has been honored remains so for the duration of the request. Finally, permission to use the resource is withdrawn once the request to use it is withdrawn. These assumptions concern the arbiter's logical input-output behavior; they are formulated explicitly as axioms whose satisfiability by models of physical processes may then be examined.

Further assumptions concern the continuity of the implementing physical phenomena, one characterisitic of devices built on Newtonian principles. They are made by restricting our attention to a particular class of models. The essence of these assumptions is the following. Real signals are continuous changes in some physical quantity. And if input signals to real hardware be continuously deformed the hardware reacts by continuously deforming the resulting output signals.

Our investigation combines techniques familiar from the theory of specification and verification in computer science with techniques familiar from control theory in electrical engineering. The use of axioms to specify a system's abstract input-output behavior is widespread in the former, while the use of so-called dynamic systems to model a circuit's concrete internal behavior is common, often in the guise of linear systems, in the latter. We combine these techniques by asking whether there exists a continuous automaton that satisfies the axioms we propose for an ideal arbiter. Since this combination of standard logical techniques is unusual — and to our knowledge unique — in the literature on synchronizer failures, we assume no acquaintance with dynamic systems or specification methods.

The paper is organized as follows. In the next section we formalize our assumptions and discuss their relation to those usually made concerning arbiters. In the subsequent section we then prove that the assumptions we have made about an arbiter's input-output behavior are not simultaneously satisfiable by a certain mathematical model of Newtonian physical processes. In particular we show that any continuous automaton that satisfies the explicit assumptions proposed above concerning its interaction with an asynchronous environment will react to certain conflicting requests by granting simultaneous access to the shared resource. Following discussion of certain technical aspects of the result, we conclude with some remarks on the result's possible practical and theoretic significance.

2 On Ideal Arbiters

We begin by determining what we shall mean by a solution to the mutual exclusion problem, how an arbiter solves the problem, and what properties it must have to perform its function properly.

The synchronization task is most succinctly depicted in a simple concrete situation. Suppose a multiprocessing system consists of two independently clocked processors connected via a common bus to passive components such as memory and input-output units. Since the processors are driven by independent clocks, their attempts to use the bus can occur at arbitrarily interleaved moments and, in particular, even simultaneously. Simultaneous transmission, however, corrupts all of the signals on the bus. So numerous techniques, such as synchronous buses, CSMA/CD or token rings, among others, are used in such systems to ensure that if both processors try to transmit at once, one of them defers to the other.

An especially simple technique involves the use of a hardware element called an *arbiter* to mediate between competing demands on the bus. The processors request the use of the bus from the arbiter, which grants a request once the bus is free. Upon completing its transmission, a processor signals to the arbiter that it has relinquished the bus. The essence of the arbiter's task is to arbitrate between any conflicting requests by ensuring that only one request at a time is ever granted. By linking each of the processors to a common arbiter we arrive at the configuration seen in Fig. 1.

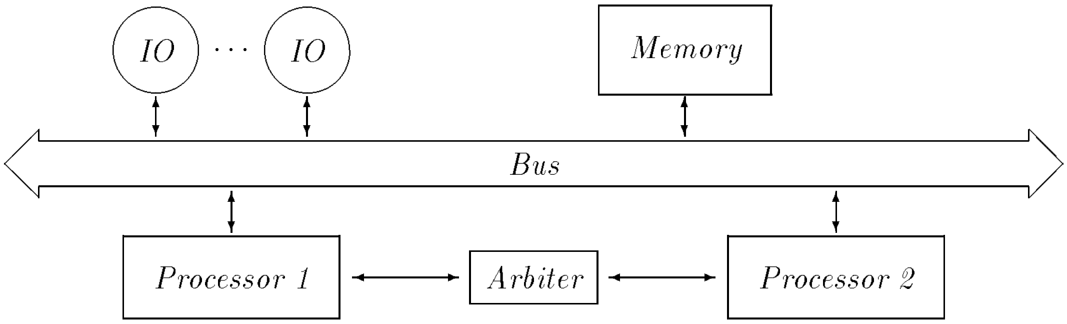

Figure 1: A Simple Multiprocessor System with Arbiter

Now if the processors abide by the rules of a standard four-phase handshake protocol, they will avoid colliding on the bus. The procedure uses two lines apiece from each processor to the arbiter and consists of the four phases depicted in Fig. 2.

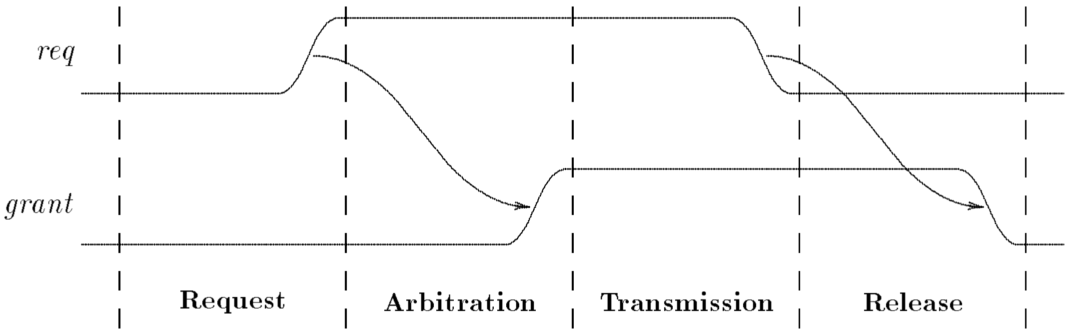

Figure 2: Arbitration Protocol

Request A processor desiring the use of the bus informs the arbiter of its request by setting the appropriate request line high. A request may be made at arbitrary times *(Independence of inputs)*, so long as *grant* is low.

Arbitration If the request does not conflict with a request from another processor, it will be granted *(Liveness)*. Only after *grant* has been set high may the requesting processor proceed.

Transmission Once a processor has acquired the use of the bus, it may transmit as long as it pleases by leaving *req* asserted *(Dominance)*. The arbiter's duty is to exclude both processors from being granted the use of the bus simultaneously *(Mutual Exclusion)*. Upon completing its transmission, a processor relinquishes the bus by setting the *req* line low.

Release The arbiter acknowledges the processor's having relinquished the bus by setting *grant* low, whereupon it is ready to grant further requests *(Reset)*. Now the processor is also ready to issue further requests.

This procedure corresponds exactly to the arbiters described, for example, in [PFL75] and [CAV86] and it captures the essence of such arbiters as in [Plu72] and [vB82] whose task includes not only the exclusion of simultaneous access to shared resources, but the transfer of data as well. In order to avoid unnecessary complications we shall restrict our investigation to the single, crucial task of arbitrating between possibly competing requests. Furthermore, since our argument does not depend on the number of processors involved, we shall simplify the discussion by restricting the investigation to a two-input arbiter. Thus we may concentrate on formalizing the two-way handshake procedure outlined above without compromising the generality of the results.

Our formalization of such properties as liveness and mutual exclusion will consist of axioms concerning the input-output behavior of a mathematical model of an arbiter. Since we wish to model physical voltage levels by means of real numbers and input signals as continuous, real-valued functions of time, we shall assume that the arbiter is a *continuous automaton*, a species of dynamic system closely related to those often used for investigations into the stability of feedback systems in control theory [BS70]. Such models have been applied, for instance, to study metastability in bistable systems [Mar81], [KC87]. By contrast with the discrete automata familiar in computer science, continuous automata have a continuous state space and they react to continuous inputs by producing continuous outputs. They are thus adequate to model classical physical phenomena such as continuously variable voltage changes and their use to implement switching circuits as feedback systems with delay.

Let us agree to denote by $[S \to T]$ the set of all continuous functions from a topological space, S, to another, T. Then the trajectory over a state space X of a physical process through time is modelled in a dynamical system as a function in $[\mathbf{R}^+ \to X]$ called a *motion* over X, where X is often Euclidean n-space $\mathbf{R}^n$, and $\mathbf{R}^+$ is the non-negative reals, which serves as a simple model of time, each with its usual topology. It is assumed that product spaces $S \times T$ carry the product topology of the topologies on S and T, subspaces $S \subseteq T$ are equipped with the topology induced by T, and function spaces $[S \to T]$ carry some arbitrary topology with the so-called *splitting* property: If for every continuous function $\alpha : S \times T \to U$ the function $\hat{\alpha} : S \to [T \to U]$ obtained from α by currying is continuous then the topology on $[T \to U]$ is said to be splitting. This is a very general property enjoyed, for instance, by the *compact-open* topology or the topology of *pointwise convergence*.

A continuous automaton, then, is an automaton whose function is described in terms of motions over its input, output, and state spaces.

Definition 2.1 *A* continuous automaton Ξ *is given by*

$$\Xi \;=\; \langle \mathbf{R}^n, \Sigma, \mathbf{R}^m, U, \varphi, \lambda \rangle,$$

whose components are:

$\mathbf{R}^n$	*input space*
Σ	*state space*
$\mathbf{R}^m$	*output space*
$U \subseteq [\mathbf{R}^+ \to \mathbf{R}^n]$	*admissible input motions*
$\varphi \in [\Sigma \times U \times \mathbf{R}^+ \to \Sigma]$	*transition or system function*
$\lambda \in [\Sigma \to \mathbf{R}^m]$	*output function*

where Σ *is some topological space.*

In our setting, where we regard a continuous automaton as a physical device that reacts to continuously changing, rather than discrete inputs, an element of the first component, $v \in \mathbf{R}^n$, represents the simultaneous input voltages on n input lines. Σ is the set of the automaton's internal states, *e.g.* $\mathbf{R}^k$ for suitable k. The third component, $\mathbf{R}^m$, is the set of simultaneous output voltages on m output lines. U is the set of continuous input motions, voltage trajectories through time, actually accepted by the system, *i.e.* to which it responds in the manner prescribed by the system and output functions. The system function φ specifies in what state $\varphi(s, u, t) \in \Sigma$ the system Ξ arrives by time t starting from state $s \in \Sigma$ given the input motion $u \in U$. Finally, the output function λ describes how the internal state affects the voltages observed on the output lines.

In the following we assume that an arbiter (see Fig. 3) is just such a continuous auto-

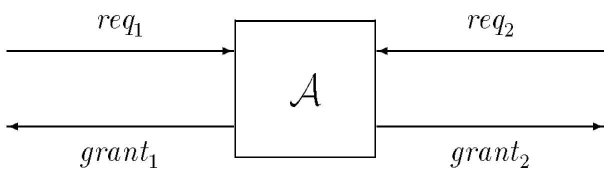

Figure 3: An Arbiter

maton $\mathcal{A} = \langle \mathbf{R}^2, \Sigma_{\mathcal{A}}, \mathbf{R}^2, U_{\mathcal{A}}, \phi_{\mathcal{A}}, \lambda_{\mathcal{A}} \rangle$ that fulfills one or more of the several assumptions, to be formalized below, concerning the arbitration protocol. We shall occasionally write $\mathcal{A} \models \Phi$ to express that the device $\mathcal{A}$ satisfies a predicate Φ. We refer in the following to the components of a specific continuous automaton $\mathcal{A}$ without mentioning the index $\mathcal{A}$. It will also be useful to assume throughout that the input and output signals req_i and $grant_i$ for $i \in \{1, 2\}$ are $req \stackrel{def}{=} \langle req_1, req_2 \rangle \in U$ and $grant = \langle grant_1, grant_2 \rangle \stackrel{def}{=} \lambda \circ \varphi$ so that $req : \mathbf{R}^+ \to \mathbf{R}^2$ and $grant : \Sigma \times U \times \mathbf{R}^+ \to \mathbf{R}^2$. Thus $req(-)$ and, for fixed parameters s and req, $grant(s, req, -)$ are input and output motions in the real plane. We shall assume that the subsets $\mathbf{0}, \mathbf{1} \subseteq \mathbf{R}$ that represent the signals 'low' and 'high' are disjoint opens, which we shall depict, for purposes of illustration, as intervals in $\mathbf{R}$, the 'undefined' range being the (closed) remainder $\mathbf{X} \stackrel{def}{=} \mathbf{R} \setminus (\mathbf{0} \cup \mathbf{1})$. Our choice of $\mathbf{0}$ and $\mathbf{1}$ to be open, rather than arbitrary, sets is mathematically expedient and appears to be physically innocuous. Finally, we confuse functions systematically with their extensions to intervals, so that we may write, for example, $req_i[t, t') \subseteq \mathbf{1}$ to mean $\forall t \leq \tau < t'.\ req_i(\tau) \in \mathbf{1}$.

We turn our attention now to the class of input signals whose effect on the arbiter we intend to investigate. These will be pairs, $\langle req_1, req_2 \rangle$, of single, persistent requests, each of which may occur at an arbitrary time, and is never again withdrawn. If we assume that there is an upper bound $\varepsilon > 0$ on the duration of a signal change, then a persistent request at time $\tau > \varepsilon$ on the input line i is characterized by

$$\exists \sigma \in [\tau - \varepsilon, \tau].\, req_i[0, \sigma) \subseteq \mathbf{0} \quad \& \quad req_i[\sigma, \tau] \subseteq \mathbf{X} \quad \& \quad req_i(\tau, \infty) \subseteq \mathbf{1}$$

or graphically, using the conventional representations of $\mathbf{0}$, $\mathbf{X}$, and $\mathbf{1}$ common in real hardware, as in Fig. 4. By the occurrence of a signal change *at* τ is meant that the

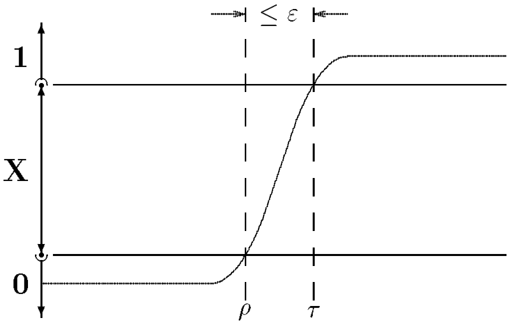

Figure 4: Trace of a Persistent Request Signal

change from $\mathbf{0}$ to $\mathbf{1}$ or vice versa has been completed by time $t = \tau$, as depicted in the figure. A request that persists for at least some duration $\delta > 0$, *i.e.* $req_i(t, t+\delta) \subseteq \mathbf{1}$ for some $t \in \mathbf{R}^+$, will be called *δ-persistent* at t and a ∞-persistent request at t is one that remains in $\mathbf{1}$ over (t, ∞).

Now let $R_i^\varepsilon(\tau) \subseteq [\mathbf{R}^+ \to \mathbf{R}^2]$ for $i \in \{1, 2\}$ and $0 < \varepsilon < \tau$ be the set of all input motions whose ith component is a single, ∞-persistent request at time τ and whose maximum rise time is ε:

$$\begin{aligned} R_i^\varepsilon(\tau) \ \stackrel{def}{=}\ & \{\langle f_1, f_2\rangle \in [\mathbf{R}^+ \to \mathbf{R}^2] \mid \exists \sigma \in [\tau - \varepsilon, \tau]. \\ & \qquad f_i[0, \sigma) \subseteq \mathbf{0} \ \& \ f_i[\sigma, \tau] \subseteq \mathbf{X} \ \& \ f_i(\tau, \infty) \subseteq \mathbf{1}\}. \end{aligned}$$

Then the set of ∞-persistent request pairs, the first of which occurs at σ and the second at τ and each of whose maximum rise times is ε, where $0 < \varepsilon < min\{\sigma, \tau\}$, will be written $R^\varepsilon(\sigma, \tau)$, *i.e.*

$$R^\varepsilon(\sigma, \tau) \ \stackrel{def}{=}\ R_1^\varepsilon(\sigma) \cap R_2^\varepsilon(\tau)$$

One final notation will be useful for simple affine transformations of motions in the real plane. For $f = \langle f_1, f_2\rangle \in [\mathbf{R}^+ \to \mathbf{R}^2]$ and $\sigma, \tau \in \mathbf{R}^+$ let $f^{(\sigma,\tau)} \in [\mathbf{R}^+ \to \mathbf{R}^2]$ be f displaced by the vector (σ, τ):

$$f^{(\sigma,\tau)}(t) \ \stackrel{def}{=}\ \langle f_1^{(\sigma)}(t), f_2^{(\tau)}(t)\rangle \ \stackrel{def}{=}\ \langle f_1(t+\sigma), f_2(t+\tau)\rangle.$$

We are now ready to formalize our intuitive description of the ideal arbiter's input-output behavior so as to arrive at a set of axioms whose satisfiability we shall examine in the next section.

The arbiter's purpose being to deal with requests from independently clocked processors, it must be ready for a request to arrive at any time from any processor not already holding the bus. A genuinely asynchronous arbiter also may not assume any regular timing relationship, *i.e.* any synchronization, between request signals from different processors. A hardware element operating under these assumptions is often said to operate in *unrestricted input change* mode [Ung71]. The two assumptions can be formalized as a closure condition on the arbiter's input space U.

CLOSURE

$$\forall \alpha > \varepsilon .\, \exists \sigma, \tau \in \mathbf{R}^+ .\, min\{\sigma, \tau\} > \alpha \quad \& \quad R^\varepsilon(\sigma, \tau) \cap U \neq \emptyset$$
$$\forall u \in U .\, \forall \sigma, \tau \in \mathbf{R}^+ .\, u^{(\sigma,\tau)} \in U$$

The first condition means the arbiter's admissible inputs U include ∞-persistent pairs of requests occurring at σ and τ, where the occurrence times may be arbitrarily late. Note that this condition depends on the parameter ε expressing the maximum rise time of the individual requests that U is to contain. When in the sequel we say an arbiter $\mathcal{A}$ satisfies the CLOSURE axiom we shall mean that there is a maximal rise time $\varepsilon > 0$ for which $U_{\mathcal{A}}$ fulfills the first condition. This characteristic parameter ε is taken, from here on, to be fixed. The second condition says that the input space is closed under arbitrary displacements of input signals to the left along the time axis. Since each component u_i of $u = \langle u_1, u_2 \rangle$ is displaced independently, this condition expresses the absence of any systematic timing relationship between signals arriving on distinct input lines. We could of course have required that the input space U be closed as well under displacements to the right. Such a requirement would mean that any device satisfying the axiom admits the same set of requests over its entire life. But we shall not need the additional assumption to prove the results of the next section.

An important property of an ideal arbiter is that it not leave persistent requests unattended: Of requests that persist long enough, the arbiter must eventually grant one. Note that this liveness condition covers the case where there is only one request and that the CLOSURE conditions ensure that there are requests in U that are nothing if not persistent.

LIVENESS

$$\begin{aligned} &\forall \emptyset \neq E \subseteq \{1,2\} .\, \forall p \in \Sigma .\, \forall req \in U .\, \forall t \in \mathbf{R}^+ . \\ &\quad (\ \forall i \in E .\, req_i(t, t+\Delta) \subseteq \mathbf{1} \quad \& \quad \forall j \in \{1,2\} \setminus E .\, req_j(t, t+\Delta) \subseteq \mathbf{0}\ ) \\ &\qquad \Rightarrow\ \exists k \in E .\, \exists s > t .\, grant_k(p, req, s) \in \mathbf{1} \end{aligned}$$

The LIVENESS axiom says that whenever there has been a set E of requests pending for some time at least as long as Δ then eventually one of them will be granted. Requests that fail to persist as long as Δ need not elicit any reaction. For $E = \{1\}$ the requirement is illustrated by the accompanying Fig. 5.

It is important to note that the axiom places no upper bound on the time, s above, when the request is finally granted. In the particular case where there are two competing requests, *i.e.* $E = \{1, 2\}$, the arbiter is allowed to take as much time as it needs to resolve the situation and decide to grant a particular request. Thus LIVENESS accommodates the fact that real arbiters can go metastable for indefinite periods upon receipt of simultaneous requests. All we require is that the arbiter eventually emit a signal granting some pending, Δ-persistent request.

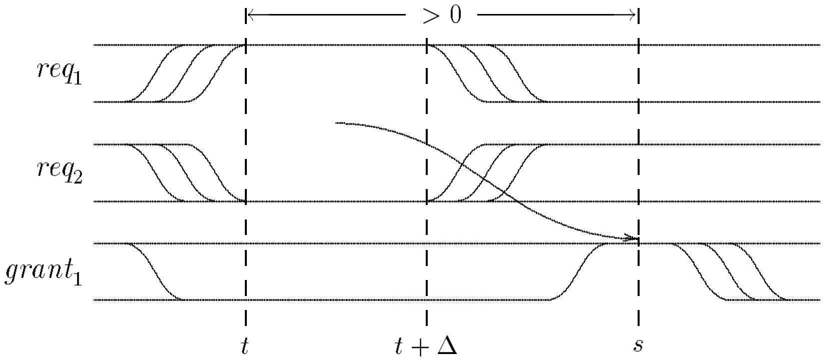

Figure 5: Signals satisfying LIVENESS ($E = \{1\}$)

This axiom depends on a parameter Δ, the minimum hold time for input requests. Requiring that an arbiter $\mathcal{A}$ satisfy LIVENESS means there is some $\Delta > 0$ for which the above formula is true of $\mathcal{A}$. Like ε, Δ is taken to be fixed throughout the paper.

The dominance property ensures that every processor is allowed exclusive access to the bus for as long as it wishes, once the arbiter has granted its request, since according to the four-phase handshake protocol the arbiter may not withdraw a grant of its own accord. Such behavior is sometimes called *non-preemptive*. We require that

DOMINANCE

$$\begin{aligned} &\forall p \in \Sigma.\, \forall req \in U.\, \forall r, t \in \mathbf{R}^+.\, \forall s, \hat{t} \in [r, t). \\ &\quad req_i[r, s] \subseteq \mathbf{X} \quad \& \quad req_i(s, t) \subseteq \mathbf{1} \quad \& \quad grant_i(p, req, \hat{t}\,) \rightsquigarrow 1 \\ &\qquad \Rightarrow\ grant_i(p, req, (\hat{t}, t)) \subseteq \mathbf{1} \end{aligned}$$

where $i \in \{1, 2\}$ and $grant_i(p, req, t) \rightsquigarrow 1$ abbreviates the predicate

$$grant_i(p, req, t) \in \mathbf{X} \quad \& \quad \exists t' > t \in \mathbf{R}^+.\, grant_i(p, req, (t, t')) \subseteq \mathbf{1},$$

which means that $grant_i$ is imminent at t, *i.e.* in $\mathbf{X}$ at time t, but on its way into $\mathbf{1}$. Thus DOMINANCE reads as follows: If during some interval $[r, t)$ a request arrives that persists until t and a grant is imminent at $\hat{t} \in [r, t)$, then the grant may not be withdrawn during $(\hat{t}, t)$. One such set of signals is illustrated in Fig. 6.

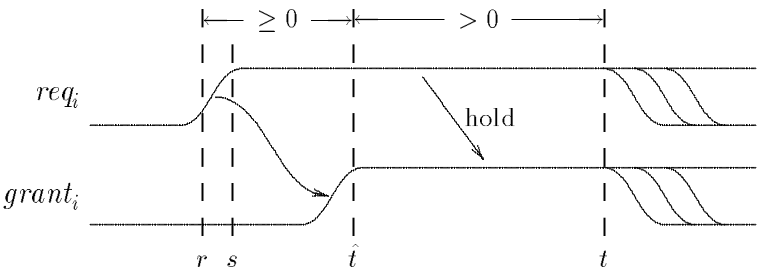

Figure 6: Signals satisfying DOMINANCE

Note that the second line of the axiom is being used to express a correspondence between the occurrence of the present grant and the arrival of the most recent request. It is these two events — and no others — whose whose causal connection the axiom is meant to constrain. The desired correspondence must be expressed in terms of signal changes, *i.e.* flanks, rather than signal levels, distinguishing this axiom from the rest.

Requiring that the signal granting a request remain stable for the duration of the request also ensures that processor i may interpret $grant_i(p, req, t) \in \mathbf{1}$ as an unambiguous sign to proceed, rather than a transitory or tentative response. The requirement is tantamount to assuming that an ideal arbiter's internal workings, including metastable or oscillatory crises, do not confuse its environment by having it issue fleeting signals.

The next axiom describes the reset phase of the handshake protocol: If a request is held in $\mathbf{0}$ long enough then any previous grant will end up in $\mathbf{0}$ as well.

RESET

$$\begin{aligned}&\forall p \in \Sigma.\, \forall req \in U.\, \forall r, t \in \mathbf{R}^+.\\ &\quad t - r \geq \Delta \quad \& \quad req_i(r, t) \subseteq \mathbf{0}\\ &\qquad\quad \Rightarrow\; \exists s \in [r, t).\, grant_i(p, req, (s, t)) \subseteq \mathbf{0}\end{aligned}$$

Here again, $i \in \{1, 2\}$. As before, the parameter Δ embodies the minimum hold time before the arbiter is guaranteed to react, in this case to the withdrawal of a request. Note that via the condition $s \in [r, t)$ RESET requires a finite response time of at most Δ for the resetting of $grant_i$, in contrast to the unbounded response time LIVENESS allowed for the grant to be issued. This restriction is innocuous since processors do not conflict during the protocol's reset phase: No provision need be made for unbounded response times due to metastable operation.

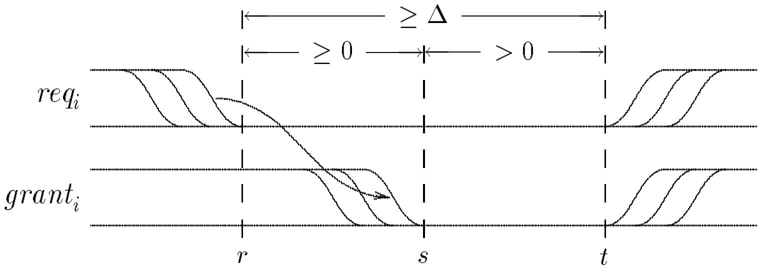

Figure 7: Signals satisfying RESET

We now come to the last and most obvious property of an ideal arbiter. If CLOSURE, DOMINANCE, LIVENESS, and RESET were the only axioms an arbiter had to fulfill then it would be only too simple to find a correct implementation: The trivial circuit consisting of just two wires, one connecting input req_1 with output $grant_1$ and the other connecting req_2 with $grant_2$ would do. (We leave the proof as an exercise to the reader.) The axiom which distinguishes an arbiter from this trivial circuit is the axiom of mutual exclusion: At no time may both output lines show $\mathbf{1}$. We choose to formulate this positively as follows:

MUTEX

$$\forall p \in \Sigma . \forall req \in U . \forall s,t \in \mathbf{R}^+ . \forall i,j \in \{1,2\},\ i \neq j .$$
$$grant_i(p, req, [s,t]) \subseteq \mathbf{1} \ \Rightarrow\ grant_j(p, req, [s,t]) \subseteq \mathbf{0}$$

Mutual exclusion is not, strictly speaking, a property of the arbiter. It is instead a property of the protocol, one side of which is illustrated in Fig. 2, in which of course the processors participate as well. In order to concentrate our attention exclusively on the arbiter, we simply assume that the processors keep their part of the bargain by observing the protocol's conventions. Under that assumption, mutual exclusion is guaranteed if the arbiter has the property we are calling MUTEX.

Having stated various explicit assumptions about arbitration and argued that each is reasonable in isolation, we should answer some standard questions about their global properties. To begin with, one might ask if any four of the axioms we have formulated can be satisfied by Newtonian automata. In fact, we have already suggested, in the discussion of the MUTEX axiom, one way to build a circuit that satisfies all four of the other axioms. Similarly, a circuit in which both output lines are tied to $\mathbf{0}$ satisfies all of the assumptions but LIVENESS. The reader may find it instructive to construct circuits that satisfy other combinations of four axioms. There are simple solutions if CLOSURE and DOMINANCE are dropped, the case of RESET remains open.

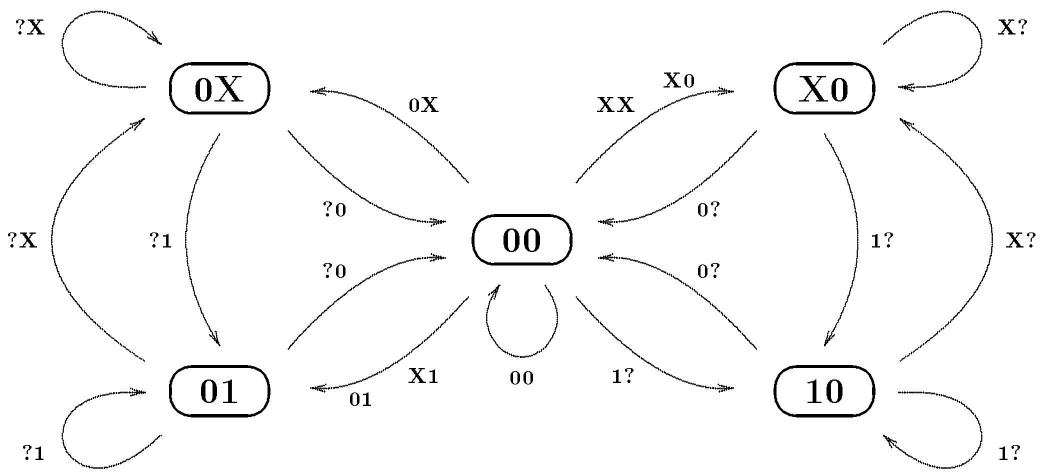

Figure 8: A Discrete Arbiter

Finally, the axioms are consistent: they can be shown to posess a model, if we take the liberty of interpreting them in such a way that they refer not indeed to *continuous* but rather to *discrete* automata. For if we take a discrete domain $\{0, X, 1\}$ of voltages with $\mathbf{0} = \{0\}$, $\mathbf{1} = \{1\}$ and $\mathbf{X} = \{X\}$, and a discrete domain $\mathbf{N}$ of time for $\mathbf{R}^+$, then with $\varepsilon = 0$ and $\Delta = 3$ it is not difficult to check, given appropriate interpretations of *req* and *grant*, that the above Moore automaton satisfies all five arbiter axioms. Since the axioms are consistent, it makes sense to use them to define formally what we shall mean by an "ideal arbiter".

Definition 2.2 *An* ideal arbiter *is an automaton that satisfies the axioms* CLOSURE, LIVENESS, DOMINANCE, RESET, *and* MUTEX.

We shall in the remaining considerations restrict our attention again to the standard interpretation of the axioms in the domain of continuous automata. Our assumptions are thus embodied in our choice of a class of models for Newtonian devices and in our choice of axioms governing the behavior of arbiters. The investigation asks, in the next section, whether there are such models that satisfy such axioms. In order that the results apply broadly we have made comparatively weak assumptions, on both the class of models in general and the axioms for arbiters in particular.

Our choice of model, continuous automata, for Newtonian devices was influenced by the investigations of [Mar81] and [KC87] into metastable operation of bistable elements. There, two additional assumptions about such automata are used to prove the results: that the system be non-anticipatory and that its future behavior be determined completely by its present state. Though such assumptions may well apply to devices reasonably called Newtonian, we do not need them to prove the results of the next section.

Our choice of axioms was influenced by the work of Barros and Johnson [BJ83]. But the differences are considerable. In particular, they stipulate the following additional requirements for an arbiter. All signal changes at the outputs are caused by certain changes at the inputs. Output changes are invariably complete transitions from one logical level to the other and occur with a certain minimal speed. High and low signals are always held for a minimum period. Finally, there is an upper bound on the arbiter's response time.

According to our definition an ideal arbiter may well emit arbitrarily short output pulses or, more importantly, respond arbitrarily slowly to input signals (the reset phase aside). Thus our axioms formulate rather weak, but common, requirements for an ideal arbiter. Any real device might be expected to satisfy further electrical or timing requirements, albeit not bounded response time, to function properly in a practical setting.

3 Newtonian Realizations of Ideal Arbiters

We shall now study the effect of input signals from $R^{\varepsilon}(\sigma, \tau) \cap U$ on the behavior of the device $\mathcal{A}$ (as in Fig. 3), a continuous automaton, which we assume fixed throughout. Recall that the set $R^{\varepsilon}(\sigma, \tau) \cap U$ consists of admissible input signals in which each processor issues an ∞-persistent request, the one at time σ and the other at τ. Assuming that $\mathcal{A}$ satisfies the axioms Closure, Liveness, Dominance, and Reset we derive an input-output behavior which violates the axiom MutEx.

Our first lemma is a simple consequence of the Liveness assumption. It asserts that if σ and τ are far enough apart, the appropriate grant line will eventually be set high; furthermore, one of the requests will be granted no matter how they are timed.

Lemma 3.1 *Let $\mathcal{A}$ satisfy the* Liveness *axiom, $p \in \Sigma$ be an arbitrary initial state, and req be an input signal in $R^{\varepsilon}(\sigma, \tau) \cap U$. Then the following propositions hold:*

(1) $$\tau \geq \sigma + \Delta + \varepsilon \;\;\Rightarrow\;\; \exists t > \sigma .\, grant_1(p, req, t) \in \mathbf{1}$$

(2) $$\sigma \geq \tau + \Delta + \varepsilon \;\;\Rightarrow\;\; \exists t > \tau .\, grant_2(p, req, t) \in \mathbf{1}$$

(3) $$\exists k \in \{1, 2\} .\, \exists t > max\{\sigma, \tau\} .\, grant_k(p, req, t) \in \mathbf{1}$$

Proof: Suppose we are given $req \in R^{\varepsilon}(\sigma,\tau)$ and $p \in \Sigma$.

Ad (1): Let σ and τ be such that $\tau \geq \sigma + \Delta + \varepsilon$. Since $\langle req_1, req_2\rangle = req \in R^{\varepsilon}(\sigma,\tau)$ we have on the one hand $req_1(\sigma,\infty) \subseteq \mathbf{1}$, *i.e.* in particular

$$req_1(\sigma, \sigma+\Delta) \subseteq \mathbf{1}, \tag{i}$$

and on the other that there exists a ρ in the interval $[\tau-\varepsilon,\tau]$ such that $req_2[0,\rho) \subseteq \mathbf{0}$. This means, since $\rho \geq \tau - \varepsilon \geq \sigma + \Delta$, that

$$req_2(\sigma, \sigma+\Delta) \subseteq \mathbf{0}. \tag{ii}$$

Given (i) and (ii) we may now specialize LIVENESS with $E = \{1\}$ to obtain $t > \sigma$ such that $grant_1(p, req, t) \in \mathbf{1}$. The proof for (2) is analogous.

Ad (3): Let $\rho \overset{def}{=} max\{\sigma,\tau\}$. Then since $\langle req_1, req_2\rangle = req \in R^{\varepsilon}(\sigma,\tau)$ we have, as before, that $req_1(\sigma,\infty) \subseteq \mathbf{1}$, whence $req_1(\rho,\rho+\Delta) \subseteq \mathbf{1}$ from which, with the analogous argument for req_2, the conclusion follows by specializing LIVENESS to $E = \{1,2\}$. ∎

We shall regard the quadrant $\mathbf{R}^+ \times \mathbf{R}^+$ of the real plane as a parameter space from which we may choose arbitrary request times σ and τ, for which we then consider input motions from $R^{\varepsilon}(\sigma,\tau) \cap U$. Viewed in this way, the conclusion of Lemma 3.1 admits a more geometrical statement. Its first clause says, if we choose $\langle\sigma,\tau\rangle$ from $A_1 \overset{def}{=} \{\langle\sigma,\tau\rangle \mid \tau \geq \sigma + \Delta + \varepsilon\}$, the region at least $\Delta + \varepsilon$ above the diagonal, then any pair of requests from $R^{\varepsilon}(\sigma,\tau) \cap U$ will drive a device $\mathcal{A} \models$ LIVENESS from an arbitrary initial state into a state in which it issues a grant on line 1. Its second clause says the grant will be issued on line 2 for any choice of $\langle\sigma,\tau\rangle$ from $A_2 \overset{def}{=} \{\langle\sigma,\tau\rangle \mid \sigma \geq \tau + \Delta + \varepsilon\}$, the region at least $\Delta+\varepsilon$ to the right of the diagonal. The third clause asserts that arbitrary requests, including those whose arrival times lie in the strip $(\mathbf{R}^+ \times \mathbf{R}^+) \setminus (A_1 \cup A_2)$ along the diagonal, will drive the device to issue a grant. Thus the device arbitrates even between simultaneous requests.

Our next order of business is to build a choice function Req that, given parameters σ and τ, chooses, for any device $\mathcal{A} \models$ CLOSURE, a pair of requests from $R^{\varepsilon}(\sigma,\tau) \cap U$. The following lemma asserts that Req can in fact be constructed in such a way that it is continuous in each of its arguments. This in effect reformulates the logical predicates of the CLOSURE axiom in a form that permits us to investigate their consequences using topological reasoning. The proof makes use of very general properties of the topology on the function space U.

Lemma 3.2 *Let $\mathcal{A}$ satisfy the* CLOSURE *axiom. Then for each interval $I = [\alpha,\beta] \subseteq \mathbf{R}^+$ with $\varepsilon < \alpha \leq \beta$ there exists a continuous function $Req : I \times I \to U$ such that*

$$\forall \sigma,\tau \in I.\, Req(\sigma,\tau) \in R^{\varepsilon}(\sigma,\tau).$$

Proof: Let $\mathcal{A}$ satisfy CLOSURE, and let $I = [\alpha,\beta] \subseteq \mathbf{R}^+$ with $\varepsilon < \alpha \leq \beta$ be given. By the first CLOSURE condition there is a pair $s,t \in \mathbf{R}^+$ with $min\{s,t\} > \beta$ and an input $req = \langle req_1, req_2\rangle \in R^{\varepsilon}(s,t) \cap U$. Consider the function $\hat{r} : I \times I \longrightarrow [\mathbf{R}^+ \to \mathbf{R}^2]$ defined as

$$\hat{r}(\sigma,\tau) \overset{def}{=} req^{(s-\sigma,t-\tau)}$$

The expression $req^{(s-\sigma,t-\tau)}$ is well defined if $s-\sigma,\ t-\tau\ \in \mathbf{R}^+$, which is the case since $\sigma,\tau \in [\alpha,\beta]$ and $\beta < s,t$. Now we have $\hat{r}(I\times I) \subseteq U$ by the second CLOSURE condition. So we take the required function $Req : I \times I \to U$ to be $\hat{r}$. It remains to be shown that $\hat{r}$ is continuous and that for all σ,τ in I we have $\hat{r}(\sigma,\tau) \in R^\varepsilon(\sigma,\tau)$.
We begin with the continuity of $\hat{r}$. The function $\tilde{r} : I\times I\times \mathbf{R}^+ \longrightarrow \mathbf{R}^2$ given by

$$\tilde{r}(\sigma,\tau,v) \stackrel{def}{=} \hat{r}(\sigma,\tau)(v)$$

is continuous since it is composed, as the equation $\hat{r}(\sigma,\tau)(v) = req^{(s-\sigma,t-\tau)}(v) = \langle req_1(v+s-\sigma), req_2(v+t-\tau)\rangle$ shows, from continuous functions by continuous construction. Since the function $\hat{r}$ is obtained from $\tilde{r}$ by the process of currying and the topology on $[\mathbf{R}^+ \to \mathbf{R}^2]$ has the splitting property, $\hat{r}$ is also continuous. The continuity of Req now follows from the assumption that U carries the topology induced by $[\mathbf{R}^+ \to \mathbf{R}^2]$.
Now we show $\hat{r}(\sigma,\tau) \in R^\varepsilon(\sigma,\tau)$ for all $\sigma,\tau \in I$. Let $\sigma,\tau \in I$ be given. Assume further that $\hat{r}_i = \pi_i \circ \hat{r}$ for $i \in \{1,2\}$ so that $\hat{r} = \langle \hat{r}_1, \hat{r}_2\rangle$. Since $req \in R^\varepsilon(s,t)$ we have $req_1 \in R_1^\varepsilon(s)$. This implies there exists $\rho \in [s-\varepsilon, s]$ with

$$req_1[0,\rho) \subseteq \mathbf{0} \quad \& \quad req_1[\rho,s] \subseteq \mathbf{X} \quad \& \quad req_1(s,\infty) \subseteq \mathbf{1}.$$

This will now be used to show $\hat{r}_1(\sigma,\tau) \in R_1^\varepsilon(\sigma)$. Note that $\sigma \in I = [\alpha,\beta]$, $\rho \in [s-\varepsilon,s]$, and $\varepsilon < \alpha$ implies $0 < \sigma+\rho-s$. This means the interval $[0,\sigma+\rho-s)$ is nonempty, so we may compute:

$$\begin{array}{rcl} \hat{r}_1(\sigma,\tau)[0,\sigma+\rho-s) & = & {req_1}^{(s-\sigma)}[0,\sigma+\rho-s) \\ & = & req_1[s-\sigma,\rho) \\ & \subseteq & req_1[0,\rho) \\ & \subseteq & \mathbf{0}. \end{array} \tag{α}$$

Further, $\rho \leq s$ implies that $[\sigma+\rho-s,\sigma]$ is nonempty, as is (σ,∞) and we obtain by similar calculations

$$\hat{r}_1(\sigma,\tau)[\sigma+\rho-s,\sigma] \ \subseteq \ \mathbf{X} \tag{β}$$

and

$$\hat{r}_1(\sigma,\tau)(\sigma,\infty) \ \subseteq \ \mathbf{1}. \tag{γ}$$

Observe that $\sigma+\rho-s \in [\sigma-\varepsilon,\sigma]$ since $\rho \in [s-\varepsilon,s]$, so (α), (β), and (γ) together yield $\hat{r}_1(\sigma,\tau) \in R_1^\varepsilon(\sigma)$. Thus, with an analogous computation to show that $\hat{r}_2(\sigma,\tau) \in R_2^\varepsilon(\tau)$, we have $\hat{r}(\sigma,\tau) \in R^\varepsilon(\sigma,\tau)$. ∎

We shall assume in the following that $Req : I\times I \to U$ is a fixed function defined for an appropriate $I \subseteq \mathbf{R}^+$ as in Lemma 3.2. Of particular interest will be intervals $I = [\alpha,\beta]$ long enough for $I\times I$ to intersect A_1 and A_2 and far enough from the origin that Lemma 3.2 applies. Since the main diagonal of $\mathbf{R}^+\times\mathbf{R}^+$ bisects $I\times I$, that will be the case whenever $\beta \geq \alpha+\Delta+\varepsilon$ and $\alpha > \varepsilon$, which we assume in the sequel of some I, taken to be fixed.

With the above preparation we can proceed to prove our main result: If $\mathcal{A}$ satisfies CLOSURE and LIVENESS then there necessarily exist combinations of request times σ, τ and input signals in $R^\varepsilon(\sigma,\tau)$ for which it eventually grants both requests. The proof is based on a simple idea: Lemma 3.1 and 3.2 ensure that if we choose $\langle\sigma,\tau\rangle \in I\times I$ then the pair of requests $Req(\sigma,\tau)$ causes any $\mathcal{A} \models$ CLOSURE & LIVENESS to issue a grant on line 1 or line 2. We know moreover from Lemma 3.1 that any $\langle\sigma,\tau\rangle$ from $A_1 \cap I\times I$ must yield a grant on line 1 and any $\langle\sigma,\tau\rangle$ from $A_2 \cap I\times I$ on line 2 instead. Now if the request parameters $\langle\sigma,\tau\rangle$ are made to pass through $I\times I$ continuously, starting in $A_1 \cap I\times I$ and ending in $A_2 \cap I\times I$, then $\mathcal{A}$, being a continuous automaton, must change its outputs in response to the inputs $Req(\sigma,\tau)$ continuously as well. Theorem 3.6 below will exhibit signals in the passage from $A_1 \cap I\times I$ to $A_2 \cap I\times I$ for which $\mathcal{A}$ cannot help but grant both requests.

Abstractly, the main result can be understood as an *intermediate value* argument. In this respect it is similar to the proofs, such as [Mar81], that metastable operation is unavoidable. Its proof depends crucially on the following characterization of the request times $\langle\sigma,\tau\rangle \in I\times I$ that cause an $\mathcal{A} \models$ CLOSURE & LIVENESS to issue a grant on a particular line, given the input $Req(\sigma,\tau)$.

Lemma 3.3 *Let $\mathcal{A}$ satisfy the* CLOSURE *and* LIVENESS *axioms. Then for arbitrary $p\in\Sigma$ the sets*

$$B_1^p \overset{def}{=} \{\,\langle\sigma,\tau\rangle \in I\times I \mid \exists t>\sigma.\, grant_1(p, Req(\sigma,\tau), t) \in \mathbf{1}\,\}$$
$$B_2^p \overset{def}{=} \{\,\langle\sigma,\tau\rangle \in I\times I \mid \exists t>\tau.\, grant_2(p, Req(\sigma,\tau), t) \in \mathbf{1}\,\}$$

form a non-trivial open covering of $I\times I$.

Proof: Let $\mathcal{A}$ satisfy the CLOSURE and LIVENESS axioms and assume that $p\in\Sigma$ is a fixed but arbitrary initial state. That B_1^p and B_2^p form a non-trivial open covering of $I\times I$ means that for $i\in\{1,2\}$ we have (i) $I\times I = B_1^p \cup B_2^p$, (ii) $B_i^p \neq \emptyset$, and (iii) B_i^p open in $I\times I$.

Ad (i): It suffices to show $I\times I \subseteq B_1^p \cup B_2^p$, since $B_i^p \subseteq I\times I$ by definition. But if $\langle\sigma,\tau\rangle \in I\times I$, so that $Req(\sigma,\tau) \in R^\varepsilon(\sigma,\tau)\cap U$, Lemma 3.1 ensures that there exist $k\in\{1,2\}$ and $t > max\{\sigma,\tau\}$ such that $grant_k(p, Req(\sigma,\tau), t)\in\mathbf{1}$, which implies $\langle\sigma,\tau\rangle \in B_1^p \cup B_2^p$.

Ad (ii): It is easy to see that $A_1 \cap I\times I \subseteq B_1^p$. Now by choice of I we have $A_1 \cap I\times I \neq \emptyset$, hence B_1^p is nonempty. The same argument yields $B_2^p \neq \emptyset$.

Ad (iii): We shall characterize B_1^p and B_2^p in a way that ensures that they are indeed open. To this end let the function $G_i^{p,t} : I\times I \to \mathbf{R}^+$ for $p\in\Sigma$ as above and fixed $t\in\mathbf{R}^+$ be given by

$$G_i^{p,t}(\sigma,\tau) \overset{def}{=} grant_i(p, Req(\sigma,\tau), t)$$

for $i\in\{1,2\}$. Intuitively, $G_i^{p,t}(\sigma,\tau)$ is the device's output on $grant_i$ at time t in response to the input signal $Req(\sigma,\tau)\in R^\varepsilon(\sigma,\tau)\cap U$. We observe that $G_i^{p,t}$ is continuous, for so is Req, as is $grant = \lambda\circ\varphi$, since $\mathcal{A}$ is a continuous automaton. But then the inverse image of $\mathbf{1}$ under $G_1^{p,t}$

$$(G_1^{p,t})^{-1}(\mathbf{1}) = \{\,\langle\sigma,\tau\rangle \mid grant_1(p, Req(\sigma,\tau), t)\in\mathbf{1}\,\}$$

is open because $\mathbf{1}$ is open, as is its intersection with the open half-plane $\{\,\langle\sigma,\tau\rangle \mid t>\sigma\,\}$, with t fixed as above. Now the union over all $t\in\mathbf{R}^+$ of these sets

$$\bigcup_{t\in\mathbf{R}^+} \{\,(G_1^{p,t})^{-1}(\mathbf{1}) \cap \{\,\langle\sigma,\tau\rangle \mid t>\sigma\,\}\,\}$$

is just B_1^p, which is therefore open. By an analogous argument B_2^p is also open. ∎

The main theorem is now a simple consequence of Lemma 3.3 and the fact that $I \times I$ is a connected space.

Theorem 3.4 *Let $\mathcal{A}$ satisfy the* CLOSURE *and* LIVENESS *axioms. Fix an arbitrary initial state $p \in \Sigma$. Then there is a non-empty open subset K of $I \times I$ such that if $\langle \sigma, \tau \rangle \in K$, there are times $s > \sigma$ and $t > \tau$ in $\mathbf{R}^+$ for which*

$$grant_1(p, Req(\sigma, \tau), s) \in \mathbf{1} \quad and \quad grant_2(p, Req(\sigma, \tau), t) \in \mathbf{1}.$$

Proof: Let $\mathcal{A}$ satisfy CLOSURE and LIVENESS, and let the initial state $p \in \Sigma$ be given. Now $I \times I$ is connected, since it is the product of the connected set $I = [\alpha, \beta] \subseteq \mathbf{R}^+$ with itself. By Lemma 3.3 the sets B_1^p and B_2^p form a non-trivial open covering of $I \times I$ so we must have $B_1^p \cap B_2^p \neq \emptyset$ since a connected set cannot be partitioned non-trivially into nonempty open subsets. So taking $K \stackrel{def}{=} B_1^p \cap B_2^p$ we have $K \subseteq I \times I$ is non-empty and open. Now if $\langle \sigma, \tau \rangle \in K$, then by the definitions of B_1^p and B_2^p there are indeed $s > \sigma$ and $t > \tau$ in $\mathbf{R}^+$ with the desired property. ∎

The last step will be to derive a contradiction to the mutual exclusion axiom MUTEX. Recall that MUTEX is the property

$$grant_i(p, req, [t_0, t_1]) \subseteq \mathbf{1} \;\Rightarrow\; grant_j(p, req, [t_0, t_1]) \subseteq \mathbf{0}$$

for $i, j \in \{1, 2\}$ and $i \neq j$. It implies that at no time can both grants be in $\mathbf{1}$. Now Theorem 3.4 certainly expresses highly undesirable behavior in an arbiter $\mathcal{A}$, but it does not necessarily mean that $\mathcal{A}$ violates this condition. The theorem does not say that for 'critical' input signals from $Req(K)$ the device eventually grants both requests at the *same* time; it says rather that each of them will be granted eventually. But if, in this situation, the device is also to satisfy DOMINANCE, then once it has granted the bus on an ∞-persistent request, it cannot withdraw it, so we may conclude that eventually *both* grants will be set $\mathbf{1}$ simultaneously.

Let us examine the consequences of DOMINANCE for the ∞-persistent request signals we are considering.

Lemma 3.5 *Let $\mathcal{A}$ satisfy the* DOMINANCE *and* RESET *axioms and let $p \in \Sigma$ be an arbitrary initial state. Further, let req be an input signal from $R_k^{\varepsilon}(\tau) \cap U$, for $k \in \{1, 2\}$ fixed but arbitrary, with $\tau \geq \Delta + \varepsilon$. Then for all $t \in \mathbf{R}^+$ with $t > \tau$ we have*

$$grant_k(p, req, t) \in \mathbf{1} \;\Rightarrow\; grant_k(p, req, [t, \infty)) \;\subseteq\; \mathbf{1}.$$

Proof: Assume $k \in \{1, 2\}$ to be fixed, $p \in \Sigma$, $t > \tau \geq \Delta + \varepsilon$, $req \in R_k^{\varepsilon}(\tau) \cap U$, and $grant_k(p, req, t) \in \mathbf{1}$. The choice of req yields, by definition of R_k^{ε}, a $\sigma \in [\tau - \varepsilon, \tau]$ such that

$$req_k[0, \sigma) \subseteq \mathbf{0} \quad \& \quad req_k[\sigma, \tau] \subseteq \mathbf{X} \quad \& \quad req_k(\tau, \infty) \subseteq \mathbf{1}.$$

Now we must prove that $grant_k(p, req, [t, \infty)) \subseteq \mathbf{1}$. But first we show that the assumptions imply that $grant_k$ has a rising edge in the interval $[\sigma, t)$. For with $req_k[0, \sigma) \subseteq \mathbf{0}$ and $\sigma \geq \tau - \varepsilon \geq \Delta$ the RESET axiom guarantees the existence of $s \in [0, \sigma)$ such that $grant_k(p, req, (s, \sigma)) \subseteq \mathbf{0}$. Hence, $grant_k$ remains within $\mathbf{0}$ during the interval $(s, \sigma) \neq \emptyset$, while it is in $\mathbf{1}$ at time $t > \tau > \sigma$. But since $grant_k$ is continuous and $\mathbf{1}$ open, $grant_k$ must pass through $\mathbf{X}$ in the interval $[\sigma, t)$ and there is a time $r \in [\sigma, t)$ when the grant is imminent:

$$grant_k(p, req, r) \in \mathbf{X} \quad \& \quad grant_k(p, req, (r, t]) \subseteq \mathbf{1}.$$

In other words there is an $r \in [\sigma, t)$ such that $grant_k(p, req, r) \rightsquigarrow \mathbf{1}$ is true, *i.e.* $grant_k$ has a rising edge at time r. (Implicitly this argument involves the completeness of the real numbers: Every nonempty set of reals with an upper bound has a supremum.) Given the specific shape of the input req we can now use DOMINANCE to conclude from this rising edge that the grant can no longer be withdrawn, *i.e.*

$$grant_k(p, req, (r, \infty)) \subseteq \mathbf{1}.$$

In particular, since $r < t$ we have $grant_k(p, req, [t, \infty)) \subseteq \mathbf{1}$. ∎

We would like to interject two technical remarks here concerning the lemma and its proof. First, the condition $\tau \geq \Delta + \varepsilon$ is necessary to ensure that the device is safely reset by the leading $\mathbf{0}$ phase of the input request $req \in R_k^\varepsilon(\tau)$. This resetting, in turn, ensures that any grant issued must in fact be a response to the request at time τ of req and not to some extraneous previous request 'stored' in the initial state. Secondly, the proof uses the continuity of the transition function $\varphi : \Sigma \times U \times \mathbf{R}^+ \longrightarrow \Sigma$ in the third argument $\mathbf{R}^+$ only, *i.e.* that φ is continuous over time. By contrast, the proof of Theorem 3.4 made a different use of continuity in exploiting the fact that φ is continuous over input signals (the second argument), which is a higher-order continuity property.

We are finally in a position to produce the promised contradiction to MUTEX as a corollary of the main theorem using the above lemma.

Corollary 3.6 *Let $\mathcal{A}$ satisfy the* CLOSURE*,* LIVENESS*,* DOMINANCE *and* RESET *axioms. Assume an initial state $p \in \Sigma$. Assume further that the interval $I = [\alpha, \beta]$ is such that $\alpha > \Delta + \varepsilon$. Then there is a non-empty open subset K of $I \times I$ such that if $\langle \sigma, \tau \rangle \in K$, there are times $s > \sigma$ and $t > \tau$ in $\mathbf{R}^+$ for which*

$$grant_1(p, Req(\sigma, \tau), [s, \infty)) \subseteq \mathbf{1} \quad \textit{and} \quad grant_2(p, Req(\sigma, \tau), [t, \infty)) \subseteq \mathbf{1}.$$

Proof: That a non-empty K exists and is open follow directly from Theorem 3.4. Assume, then that $\langle \sigma, \tau \rangle \in K$. Again by Theorem 3.4 we have $s, t \in \mathbf{R}^+$ with $s > \sigma$ and $t > \tau$ such that $grant_1(p, Req(\sigma, \tau), s) \in \mathbf{1}$ and $grant_2(p, Req(\sigma, \tau), t) \in \mathbf{1}$. In particular $Req(\sigma, \tau) \in R_1^\varepsilon(\sigma) \cap R_2^\varepsilon(\tau)$ and $\sigma, \tau \geq \Delta + \varepsilon$, since $\sigma, \tau \in [\alpha, \beta]$ and $\alpha > \Delta + \varepsilon$. So we can invoke Lemma 3.5 to obtain the conclusion. ∎

We conclude this section with some remarks on the mathematical character and the logical structure of these results. These issues concern the mathematical objects we have called continuous automata. What inferences may be drawn from them concerning real hardware, or its verification, is another matter, to be dealt with in part in the final section.

Logically, we have investigated the consequences of the axioms CLOSURE, LIVENESS, DOMINANCE, and RESET among the class of continuous automata. Corollary 3.6 demonstrates that any $\mathcal{A}$ satisfying those postulates cannot invariably — *i.e.* for all of its admissible inputs — satisfy MUTEX. But Corollary 3.6 does not mean, to take a more subtle point, that any continuous automaton which does indeed behave like an ideal arbiter for 'most' admissible inputs must fail to arbitrate between some requests by failing to satisfy MutEx. It could, for example, satisfy MutEx invariably, but occasionally violate one of the response properties like Liveness. The argument demonstrates merely that MutEx and the response properties are incompatible: They cannot be satisfied simultaneously.

We believe that a similar result could be obtained without the use of the Reset axiom, albeit by a more complicated argument and further assumptions like a definite start state. It is, on the other hand, not obvious how to weaken Closure significantly while claiming that a device $\mathcal{A}$ satisfying the weakened postulate admits genuinely asynchronous requests. Of the two remaining response axioms, Liveness and Dominance, we are thus led to ask whether there are weaker assumptions that still capture the essence of ideal arbitration, but are nonetheless compatible with MutEx. Since, in fact, $\mathcal{A} \models$ Closure & Liveness already leads to the undesirable behavior of Theorem 3.4, one could argue that any search for weaker postulates should concentrate on Liveness.

Mathematically, the result is, as remarked above, a form of intermediate value theorem. The classical result asserts that any continuous, real-valued function assuming a value a at some point and value $b > a$ at another must pass through all intermediate values between a and b. A simple analogon in the setting of continuous automata asserts that any continuous signal u starting at time s in a closed set C_1 and ending in some closed set C_2 at $t > s$ with $C_1 \cap C_2 = \emptyset$ must assume a value in $C = \complement C_1 \cap \complement C_2$ somewhere in the interval $[s, t]$. For if we regard u as a continuous function on the interval $[s, t]$, the assumptions imply that the inverse images $u^{-1}(\complement C_1)$ and $u^{-1}(\complement C_2)$ are nonempty, open sets which cover $[s, t]$. Now $[s, t]$ is a connected set, so by definition of connectedness the components of the open covering must intersect nontrivially, *i.e.* $u^{-1}(\complement C_1) \cap u^{-1}(\complement C_2) \neq \emptyset$. But this is equivalent to $u^{-1}(C) \neq \emptyset$ which was to be shown.

The assumptions of this simple argument may be translated to the more involved setting of Theorem 3.4 as follows. First, replace the continuous signal u by the input-output behavior of the arbiter, or more precisely, the continuous function mapping an input request pair $\langle \sigma, \tau \rangle$ to the output signal resulting from input $Req(\sigma, \tau)$. Take s and t to be the request pairs $\langle \alpha, \beta \rangle$ and $\langle \beta, \alpha \rangle$, respectively, and let the interval $[s, t]$ correspond to the subspace $I \times I$. The sets B_1^p and B_2^p in Lemma 3.3 play the role of $u^{-1}(\complement C_1)$ and $u^{-1}(\complement C_2)$.

This mathematical resemblance to the type of argument used by Marino in [Mar81] to prove that metastable behaviour afflicts bistable elements does not mean that our result has anything to do with metastability. We propose, following Marino, that 'metastable operation' in the strict sense refer to an element's failure to settle in bounded time into one the stable regions $S \subseteq \Sigma$ among its internal states. The literature also uses 'metastability' in a looser sense to refer to the observable effects of such unfortunate behavior. Note that the results in this section refer in no way to what we are calling metastable operation in the strict sense. Since Corollary 3.6 does not identify the *internal* behavior that leads a device $\mathcal{A}$ satisfying an arbiter's response properties to violate MutEx, we refrain from calling such malfunctions 'metastable operation', even in the loose sense.

4 Conclusion

In this paper we have formalized several explicit assumptions concerning the input-output behavior of an arbiter: the response properties CLOSURE, LIVENESS, DOMINANCE, RESET plus the additional, albeit crucial synchronization property MUTEX. We then investigated the models of these axioms in the class of continuous automata. The upshot is that any continuous automaton satisfying the response properties possesses a class of critical, conflicting requests that cause it to violate the mutual exclusion property.

Our result bears directly on the theory of hardware design and the practice of its formal verification. These conclusions are best motivated by comparing our work with other theoretic investigations into synchronization hardware.

By deducing logical consequences from their axioms Barros and Johnson [BJ83] and Strom [Str79] demonstrate the interrealizability of three ideal hardware elements: flipflops, inertial delays and arbiters. Their results mean that metastability of flipflops impinges on the realizability of the other elements, in particular the arbiter, as well. Their results are based, like ours, on axioms describing the input-output behavior of the various elements but they make much stronger assumptions than we concerning the arbiter, the crucial one being an upper bound on its response time. We conjecture that Barros and Johnson's results can be extended to the interrealizability of time-*un*bounded flipflops and arbiters.

The most recent qualitative results on metastable operation of bistable elements were achieved by Marino in [Mar81] and improved in [KC87]. In a careful investigation of the internal state of bistable dynamic systems they succeed in showing that if the input space is connected and contains two inputs that drive the system to different stable states, then there exist inputs that excite metastable behavior. Our work differs from theirs in concentrating on the realizability of an independently characterized class of input-output behaviors. In the absence of an axiomatically specified, abstract class of input-output behaviors that such bistable dynamic systems fail to realize, it is not clear, for example, what the published results imply for the externally observable behavior of flipflops augmented by some kind of 'metastability detector' along the lines of [Pěc76] or [KW76]. As such augmented flipflops are supposed to be free of ambiguous outputs [Pěc76], [SC79], [Mar86] and are used to implement 'metastable-free' time-unbounded arbiters [SD88] the question is of some practical relevance: If the conjecture be true that time-*un*bounded flipflops and arbiters are interrealizable, then our result implies that augmented flipflops as well as arbiters, metastable-free or not, can be expected to malfunction intermittently in ways not yet detected under laboratory conditions.

While the literature on synchronizer failures recognizes that metastable operation precludes arbitration in bounded time it also appears — perhaps by omission — to assume that perfectly reliable arbitration in *un*bounded time is possible in principle using Newtonian devices. Our result casts doubt on such an assumption.

As this discussion suggests, one way to interpret the result succinctly is as an assertion about the realizability of certain kinds of hardware: The classical synchronization problem of mutual exclusion is not soluble without intermittent failures in a genuinely asynchronous setting using Newtonian physical phenomena. That interpretation, though it speaks in guarded absolutes, does not support hasty conclusions about real arbiters. It obviously does not mean, for example, that virtually trouble-free arbiters cannot be built. Commercial systems do contain arbitration devices which, by rigorous practical standards, can be considered hugely reliable. Moreover, both theoretical and practical techniques are known that reduce the incidence of failure in synchronization hardware systematically to rates below any given, nonzero limit; see, *e.g.* [Pěc76], [EW77], [SC79], [CR79], [Cha83], [Mar86], [SD88], [Hes89]. Such failure rates may in fact be below practical or even theoretical limits of common laboratory equipment to measure them.

Another way to interpret the result is as an assertion about the formal verifiability of certain kinds of hardware. For formal verifications our result, like earlier results on synchronizer failures, has the obvious consequence that behaviors with verifiably correct discrete realizations may fail to possess verifiably correct Newtonian realizations. That problem, once recognized, may not cause serious difficulty in practice. More pernicious is a subtler potential effect. Formal verification tools are based on libraries of predefined primitive components whose behavior is described axiomatically. If such a library includes an arbiter modelled at the level of continuous time and continuous signal values, it may be tempting to assume further, perhaps indirectly, that the arbiter's input-output behavior is continuous. But, as we have shown, this would render the assumptions inconsistent. Similar problems presumably afflict the axiomatic descriptions of other simple hardware elements.

Some of the difficulties arising out of the passage from discrete specifications to continuous realizations might best be solved by devising a mathematically precise notion of *approximate implementation*. A suitable definition might be obtained by considering sequences of continuous machines that approximate a discrete behavior in the limit. Another possibility would be to constrain a partly satisfactory circuit design, say by adding timing information, until it satisfies the desired specification.

Acknowledgements. We would like to thank the Erlanger Dienstagsclub and the Edinburgher Formal System Design Club for helpful discussions, Rainer Klar for thoughtful criticism, and Mike Fourman and John Longley for their written reponse.

References

[BJ83] J. C. Barros and B. W. Johnson. Equivalence of the arbiter, the synchronizer, the latch and the inertial delay. *IEEE Transactions on Computers*, C-32(7):603–614, July 1983.

[BS70] N. P. Bhatia and G. P. Szegö. *Stability Theory of Dynamical Systems*, volume 161 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1970.

[CAV86] J. Calvo, J. I. Acha, and M. Valencia. Asynchronous modular arbiter. *IEEE Transactions on Computers*, C-35(1):67–70, January 1986.

[Cha83] T. J. Chaney. Measured flip-flop responses to marginal triggering. *IEEE Transactions on Computers*, C-32(12):1207–1209, December 1983.

[CM73] T. J. Chaney and C. E. Molnar. Anomalous behavior of synchronizer and arbiter circuits. *IEEE Transactions on Computers*, C-22(3):421–422, March 1973.

[CR79] T. J. Chaney and F. U. Rosenberger. Characterization and scaling of MOS flip-flop performance in synchronizer applications. In *Proceedings of the Caltech Conference on Very Large Scale Integration*, pages 357–374, Pasadena, California, January 1979. Caltech Computer Science Department.

[CW75] G. R. Couranz and D. F. Wann. Theoretical and experimental behavior of synchronizers operating in the metastable region. *IEEE Transactions on Computers*, C-24(6):604–616, June 1975.

[EW77] G. Elineau and W. Wiesbeck. A new J-K flip-flop for synchronizers. *IEEE Transactions on Computers*, C-26(12):1277–1279, December 1977.

[FD79] W. Fleischhammer and O. Dörtok. The anomalous behavior of flip-flops in synchronizer circuits. *IEEE Transactions on Computers*, C-28(3):273–276, March 1979.

[HE75] M. Hurtado and D. L. Elliott. Ambiguous behavior of logic bistable systems. In *Proceedings of the 13th Annual Allerton Conference on Circuit & Systems Theory*, pages 605–611, Urbana-Champaign, October 1–3 1975.

[Hes89] H. Hessenauer. Zuverlässigkeit elementarer Mechanismen zur Realisierung der Synchronisation bzw. des gegenseitigen Ausschlusses. In *Workshop Grundlagen Verteilter und Paralleler Systeme – Teil I: Kommunikation, Koordinierung und Parallele Algorithmen*, pages 99–111. Arbeitsberichte IMMD Bd.22 Nr.13, University of Erlangen-Nürnberg, October 1989.

[KC87] L. Kleeman and A. Cantoni. On the unavoidability of metastable behavior in digital systems. *IEEE Transactions on Computers*, 36(1):109–112, January 1987.

[KW76] D. J. Kinniment and J. V. Woods. Synchronization and arbitration circuits in digital systems. *Proc. Inst. Elec. Eng.*, 123:961–966, October 1976.

[Mar77] L. R. Marino. The effect of asynchronous inputs on sequential network reliability. *IEEE Transactions on Computers*, 26(11):1082–1090, November 1977.

[Mar81] L. R. Marino. General theory of metastable operation. *IEEE Transactions on Computers*, 30(2):107–115, February 1981.

[Mar86] L. R. Marino. *Principles of Computer Design*. Computer Science Press, Rockwell, 1986.

[May74] D. Mayne. Minimize computer 'crashes'. *Electronic Design*, 9:168–172, April 1974.

[Pěc76] M. Pěchouček. Anomalous response times of input synchronizers. *IEEE Transactions on Computers*, 25(2):133–139, February 1976.

[PFL75] R. C. Pearce, J. A. Field, and W. D. Little. Asynchronous arbiter module. *IEEE Transactions on Computers*, 24(9):931–932, September 1975.

[Plu72] W. W. Plummer. Asynchronous arbiters. *IEEE Transactions on Computers*, 21(1):37–42, January 1972.

[SC79] H. J. Stucki and J. R. Cox, Jr. Synchronization strategies. In *Proceedings of the Caltech Conference on VLSI*, pages 375–393, Pasadena, California, January 22–24 1979.

[SD88] N. Siddique and C. Dike. Metastable-free arbitrator coordinates processors. *Electronic Design*, 14:107–112, April 1988.

[Str79] B. I. Strom. Proof of the equivalent realization of a time-bounded arbiter and a runt-free inertial delay. In *Proceedings of the 6th Annual IEEE Symposium on Computer Architecture*, pages 179–181, New York, 1979. IEEE.

[Ung71] S. H. Unger. Asynchronous sequential switching circuits with unrestricted input changes. *IEEE Transactions on Computers*, 20(12):1437–1444, December 1971.

[VA75] M. K. Vosbury and D. N. Arden. Hazards in asynchronous sequential circuits due to unrestricted input changes. Unpublished, State University of New York, 1400 Washington Ave., Albany, New York 12222, 1975.

[vB82] Gregor von Bochmann. Hardware specification with temporal logic: an example. *IEEE Transactions on Computers*, C-31(3):223–231, March 1982.

Newtonian Arbiters Cannot be Proven Correct
— A Response —

Michael Fourman John Longley
Laboratory for Foundations of Computer Science
University of Edinburgh, UK.

The result and its repercussions. The principal result in this paper states that there can be no arbiter that behaves "perfectly" for all possible inputs, if we take a simple Newtonian model of physics in which certain minimal assumptions hold. The main difference between this and previous "impossibility" results, the authors suggest, is that other results have concentrated on problems for which there is a specified upper bound on the reaction time, whereas here we have an instance of a time-*unbounded* insoluble problem (the mutual exclusion problem). It should be noted, though, that the present result asserts a slightly weaker kind of impossibility than some others: whereas typically in time-bounded situations there is always a *positive* probability of failure, there might (according to the present model) exist an arbiter which satisfies the LIVENESS axiom with probability one (given a sensible measure on the space of inputs), and which always satisfies MUTEX and the other axioms.

Thus, as the authors point out in their Conclusion, the result does not signify that *practically* reliable arbiters cannot be built. Although we can never quite build the near-ideal arbiter whose failure probability is zero, and even accepting that there may be technological bounds on the reliability that can be achieved for a single component, there are well-known ways of combining components sequentially so that the failure probability diminishes multiplicatively, so that any desired degree of confidence can be attained. Moreover, if the set of failure states is small enough, in any physical system the state will (with probability one) eventually be pushed outside this set by unpredictable small perturbations.

As we see it, the importance of the result is not primarily in terms of the practical unrealizability of perfect arbiters—its main significance lies rather in its implications for formal verification. In particular, though a component may well behave "perfectly" in practice, any attempt to verify its behaviour formally at this low physical level will run into the theoretical obstacles discussed in the paper. Of course, in principle we could surmount these if we were prepared to refine our model still further to include an analysis of failure probabilities. Such an analysis would be unilluminating, however, and perhaps even misleading, since it might yield predictions at a greater level of detail than could be justified by the faithfulness of the Newtonian model. The force of the impossibility results, then, is to discourage us from trying to reason about such very low-level electrical phenomena, and to lead us to seek a more abstract model of the physical world as the basis for our reasoning (e.g. a discrete-time model), in which the existence of ideal arbiters, for example, might be assumed.

However, this idea of abstracting away from the physical detail raises (at least) two important issues. The first of these, noted briefly at the end of the paper, is the question of consistency in the abstract model. If we choose to incorporate physically inconsistent axioms into our model (such as the existence of a perfect arbiter), it *may* then become possible to "prove" absurdities within our formal system, albeit via abstruse reasoning of a kind not normally encountered in verification work. If this is the case, then even "correct" proofs somehow become suspect. The paradox, fortunately, is avoidable, since *formal* consistency is a weaker requirement than *physical* consistency. We could seek to show quite independently that our system was formally consistent, for example using model theory.

The second issue concerns the incorporation of components such as arbiters into a large design. However negligible the failure probability of a single arbiter on its own, in the context of a larger system this probability might be magnified if there were *systematic* interdependencies between the inputs. For example, owing to a bad design decision we might have the two input lines to an arbiter driven by the same clock, so that requests would be expected to arrive in phase. Clearly this would greatly increase the chance of a metastable crisis. It is at least conceivable that several such features working together could augment the failure probability explosively until it ceased to be negligible. The problem is that such a system could be pronounced "correct" by a verification that only reasoned about the abstract model, since the accumulation of failure probabilities would not be "seen" at this level. In this sense, the impossibility theorem can be interpreted as a warning that we ignore the physical level at our peril! One solution to the problem would be to find a set of "design rules" that would preclude the creation of situations in which this kind of interdependence arises.

Philosophical status of the physical assumptions. In the paper, the authors base their proof on a carefully chosen set of assumptions about the physical behaviour of realizable arbiters. Since these assumptions are so crucial to an understanding of the result, it seems important to give them close consideration. In most respects we find the underlying assumptions very mild—indeed, they embrace not only the behaviour of all practically realizable arbiters, but also many more unfamiliar and distinctly non-Newtonian forms of behaviour, such as the ability to look ahead in time.

At one point, however, it seems that a particular (and potentially controversial) philosophical stance is implied, but not explicitly stated. This is in the assumption that the sets denoted $\mathbf{0}$ and $\mathbf{1}$, consisting respectively of the signal values regarded as "low" and those regarded as "high", are *open* subsets of the real line. (This is really the central assumption that makes the proof work.) On the surface this certainly seems innocuous—if any other part of the system was required to act or behave in a certain way whenever a signal was high, it seems reasonable to demand that the property of highness be stable under small perturbations of the signal. However, this presupposes that the specification of what ought to happen as a result of the high signal itself determines an open set of possible behaviours. Continuing in this way, we may chain together open sets via continuous maps until we reach the top-level specification of the required "observable" behaviour of the whole system. (This is reminiscent of the propagation of small perturbations up to the macroscopic level in the "Schrödinger's cat" argument in quantum physics.)

Thus we see that we are justified in assuming $\mathbf{0}$ and $\mathbf{1}$ are open *provided* the top-level requirement takes the form of an open set of behaviours—that is, the specification is an *affirmative* predicate. Certainly this is reasonable in many cases. But it is the implicit preference of affirmative over refutative (or any other) specifications that seems to us to be the hidden assumption behind the hypothesis concerning $\mathbf{0}$ and $\mathbf{1}$. (Indeed, a game-theoretic understanding of the relationship between system designer and customer might even lead us to the view that refutative specifications are ultimately more satisfactory from a contractual viewpoint!)

Fortunately, the results of the paper do not stand or fall on this kind of debate. We propose that the philosophical difficulties here be sidestepped completely by accepting a weaker, and much less controversial, hypothesis: we merely assume that $\mathbf{0}$ and $\mathbf{1}$ are *contained in* some open sets U, V respectively, where U and V are disjoint. This is a familiar kind of topological separation condition, and merely corresponds to the fact that it should be physically feasible to discriminate between high and low.