An analysis of determinacy using a trace-theoretic model of asynchronous circuits

Mark B. Josephs

Centre for Concurrent Systems & VLSI
School of CISM
South Bank University, London

Overview

- A Question about Determinism
- Hoare's Answer
- Milner's Answer
- Concepts from Receptive Process Theory
- Deterministic Processes should have Non-Disabling Events
- Deterministic Processes should allow Neighbour-Swaps
- Deterministic Processes should be Maximally Transient
- Conclusion

A Question about Determinism

Will an asynchronous network built out of deterministic components be deterministic?

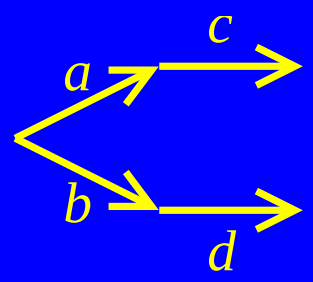
As designers of asynchronous circuits we rather take this for granted.

"It is not known how, and is believed to be physically impossible, to build a Mutual Exclusion element that is all that one might want. It cannot be made entirely from digital circuits ... The problem here is the same one that plagues synchronizers ... When the precaution of an analog difference circuit or some threshold network is not taken, the outputs may not be reliably interpreted as either high or low." (Seitz 1980)

Hoare's Answer

"The introduction of concurrency does not by itself introduce any element of nondeterminism ... Nondeterminism arises naturally from the decision to ignore or conceal those aspects of the behaviour of a system in which we are no longer interested." (Hoare 1985)

In Hoare's theory of CSP, the behaviour pattern (*process*)

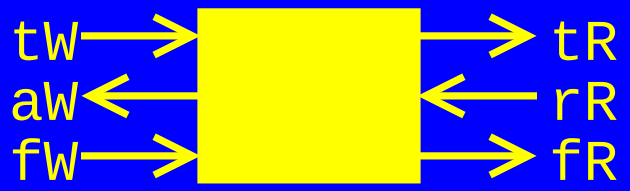


is considered to be deterministic, but hiding a and b introduces nondeterminism.

Milner's Answer

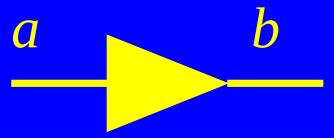
Milner (1989) proved that a stronger property (*confluence*) is preserved both by parallel composition and by hiding. In a confluent process, any two finite sequences (*traces*) of events can each be extended by a further sequence (containing the extra events present in the other one) to reach a common state.

But a Boolean register (Martin 1986) cannot be modelled by a confluent process!



`tW aW tW aW fW aW fW aW` *versus* `fW aW fW aW tW aW tW aW`

Concepts from Receptive Process Theory



safe-trace-set	$(ab)^*(\varepsilon + a)$	... safety
divergence-set	$(ab)^*aa(a+b)^*$	
trace-set	$(ab)^*(\varepsilon + a + aa(a+b)^*)$	
quiescent-trace-set	$(ab)^*$	... progress
failure-set	$(ab)^*(\varepsilon + aa(a+b)^*)$	

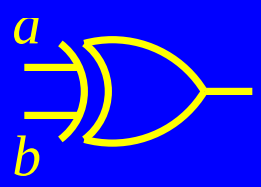
Nondeterministic choice between processes is modelled by union of their respective divergence-sets, trace-sets and failure-sets.

Determinism (conformance or refinement) ordering between processes corresponds to set-inclusion between their respective divergence-sets, trace-sets and failure-sets.

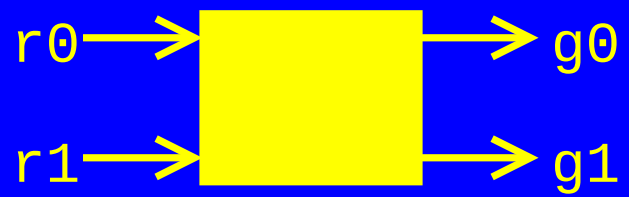
Deterministic Processes should have Non-Disabling Events

If $a \neq b$ and both sa and sb are traces, then sba is a trace ... (ND)

Note that safe-trace-sets are not necessarily closed in this way!



a and b are both safe-traces, but ab is not.



Both `r0 r1 g0` and `r0 r1 g1` are traces of a Mutex, but `r0 r1 g0 g1` is not.

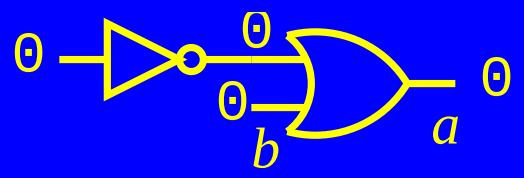
Assuming (ND) is preserved both by parallel composition and by hiding, we must refine away this nondeterminism in order to build a Mutex from gates modelled by deterministic processes.

In fact, the assumption that hiding preserves (ND) relies upon deterministic processes satisfying another property ...

Deterministic Processes should allow Neighbour-Swaps

If sa is a trace, then
$sbat$ is a safe-trace implies $sabt$ is a trace ... (NS)

Note that the safety of $sbat$ is critical!



a is a trace and, since bab is a divergence, $baba$ is a trace, but $abba$ is not.

(ND) and (NS) are jointly preserved both by parallel composition and by hiding.

If we were to refine our model of a Mutex to make its behaviour dependent upon the order in which concurrent requests are received, this would violate (NS).

So, according to our circuit-switching model, a Mutex cannot be built from gates.

Note that:
a stronger version (NS') of (NS) predicates that $sbat$ is a quiescent-trace and $sabt$ is a failure;
(Q)DI processes allow neighbour-swapping under slighly different conditions.

Deterministic Processes should be Maximally Transient

If a is an output and sa is a trace, then s is not a quiescent-trace ... (MT)

(ND), (NS') and (MT) are jointly preserved both by parallel composition and by hiding.

Our model of a Mutex with one output concealed would violate (MT).

Conclusion

In general, safety and progress properties of asynchronous circuits (whether speed-independent, quasi-delay-insensitive or delay-insensitive) can be modelled by receptive processes.
As in CSP, one can formally verify that a circuit implements a specification.

We have discovered a deterministic class of receptive processes that is suitable for modelling asynchronous circuits without arbiters and synchronizers.
As in CSP, determinism is preserved by parallel composition.
In contrast with CSP, determinism is also preserved by hiding!

The theses of Udding (1984) and Verhoeff (1994) also address the classification of processes modelling delay-insensitive circuits.