Technical Report

UCAM-CL-TR-96
ISSN 1476-2986

Number 96

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Models and logic of MOS circuits

Lectures for the Marktoberdorf Summerschool, August 1986

Glynn Winskel

October 1986

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

https://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

https://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Models and logic of MOS circuits

Lectures for the Marktoberdorf Summerschool, August 1986

by
Glynn Winskel
University of Cambridge,
Computer Laboratory,
Corn Exchange Street,
Cambridge CB2 3QG.

Abstract

Various models of hardware have been proposed though virtually all of them do not model circuits adequately enough to support and provide a formal basis for many of the informal arguments used by designers of MOS circuits. Such arguments use rather crude discrete notions of strength—designers cannot be too finicky about precise resistances and capacitances when building a chip—as well as subtle derived notions of information flow between points in the circuit. One model, that of R.E.Bryant, tackles such issues in reasonable generality and has been used as the basis of several hardware simulators. However Bryant's model is not compositional. These lectures introduce Bryant's ideas and present a compositional model for the behaviour of MOS circuits when the input is steady, show how this leads to a logic, and indicate the difficulties in providing a full and accurate treatment for circuits with changing inputs.

Models and logic of MOS circuits

by
Glynn Winskel
University of Cambridge,
Computer Laboratory,
Corn Exchange Street,
Cambridge CB2 3QG.

Abstract

Various models of hardware have been proposed though virtually all of them do not model circuits adequately enough to support and provide a formal basis for many of the informal arguments used by designers of MOS circuits. Such arguments use rather crude discrete notions of strength—designers cannot be too finicky about precise resistances and capacitances when building a chip—as well as subtle derived notions of information flow between points in the circuit. One model, that of R.E.Bryant, tackles such issues in reasonable generality and has been used as the basis of several hardware simulators. However Bryant's model is not compositional. These lectures introduce Bryant's ideas and present a compositional model for the behaviour of MOS circuits when the input is steady, show how this leads to a logic, and indicate the difficulties in providing a full and accurate treatment for circuits with changing inputs.

0. Introduction.

There are some tricky issues in the verification of hardware. We all know that verification of any device can only be done in terms of a model for its behaviour. However it is very easy to forget that verification depends crucially on the accuracy of the model. The verification of hardware has been very successful when the model assumes there are basic trusted functional devices out of which all other devices are built (see *e.g.* [Gor1. She, Mos] and the classical work on implementing Boolean functions). The mathematics of functions and functional programs is well-understood so it makes good sense to translate hardware into functions; they can be reasoned about or even run as functional programs to simulate the behaviour of the hardware. Fortunately this kind of model is often appropriate and proofs of properties of hardware amount to large but essentially trivial manipulations of expressions for functions or relations. Sometimes, however, the physical nature of the device intrudes. Designers, generally indifferent to slick proofs of correctness, use their knowledge of the physics of devices to improve performance or layout. Designers in metal oxide semiconductor (MOS) technology can use a variety of techniques. They exploit bidirectionality and the fact that signals do not have uniform strengths to improve their designs. Most approaches model such designs in an *ad hoc* way; directionality is often imposed, rather than derived, and effects due to signal strength are fudged. In what sense can a verification, based on such a model be trusted? After all directions can only be assigned correctly when the circuit behaviour is understood thoroughly and an incorrect assignment can easily lead to an incorrect prediction about the circuit's behaviour. As a last resort there are the precise models of

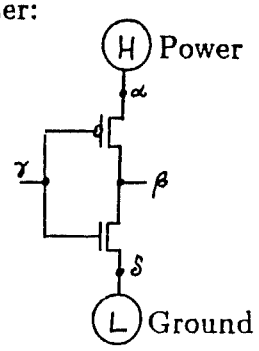
physics but, even aside from whether these are tractable or not, they are often far too detailed. A designer cannot be finicky about the precise resistance or capacitance to be realised in a VLSI chip; designs should be fairly robust in order to tolerate variations in manufacture. There is a need for a model and proof system for circuits which while close to the logical behaviour of hardware devices also captures the effects used by designers.

In these lectures we shall look at the most striking problems with many of the models in use and attempt to remedy their faults. In addressing these problems we shall make use of the ideas in R.E.Bryant's model of MOS circuits. Bryant's ideas have been developed chiefly with the aim of accurate simulation in mind and they are not directly suitable as a basis for a proof system for circuits. We shall work towards achieving this. Firstly, we illustrate some of the problems that arise in other approaches to modelling circuits.

1. Relational models and their problems.

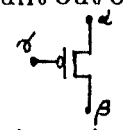
Modelling circuits as functions can often impose an unnatural directionality and lead to inaccurate models which predict the wrong behaviour. Instead we might try to model circuits as some form of relation. This is the course followed in [Gor] and less explicitly in [Mos]. It should be mentioned that although we base our criticism on the work [Gor] this is largely because it is there the problems are shown-up clearly, in their basic form. Essentially the same difficulties arise in the treatments [Mos, Mi]. Incidentally, all the approaches [Gor, Mi, Mos, Sh] seem to cope well at higher levels of abstraction.

Consider a CMOS inverter:

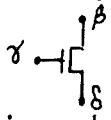


The effect of the inverter is to output at β the inverse of the value input at γ. We illustrate for this simple example how one expresses and argues in the framework of [Gor2,3] that the circuit drawn meets this specification.

The circuit is built out of two kinds of transistors. A p-type transistor $ptran(\alpha, \beta, \gamma)$, generally drawn as



is a device which, when the voltage value at the gate is low, *i.e.* $V\gamma = L$, connects α and β, so $V\alpha = V\beta$. When the value at γ is high, $V\gamma = H$, the points α and β are disconnected so we cannot say what the relation is between α and β—it all depends

on what they are connected to. The other kind of transistor, an n-type transistor $ntran(\beta, \delta, \gamma)$, drawn 

behaves in a converse fashion; when $V\gamma = H$, β and δ are connected and $V\beta = V\delta$, and when $V\gamma = L$ they are disconnected. In approaches like [Gor2,3, Mos] a circuit (and hardware in general) is modelled by a relation between values at the significant points of the circuit. This is expressed as an assertion. So in the work [Gor2,3] the assertions associated with the two transistors are

$$ptran(\alpha, \beta, \gamma) \equiv (V\gamma = L \rightarrow V\alpha = V\beta),$$
$$ntran(\beta, \delta, \gamma) \equiv (V\gamma = H \rightarrow V\beta = V\delta).$$

(We use $\equiv$ to mean definitional equality; the left-hand side stands for the right-hand side.)
Two kinds of sources are used in the CMOS inverter. *Power* connected at α is regarded as maintaining the voltage value as high at α and *ground* (or *earth*) at δ maintains the value at low. The corresponding assertions are:

$$Pow\ \alpha \equiv (V\alpha = H),$$
$$Gnd\ \delta \equiv (V\delta = L).$$

So each component is described by an assertion about values at the points with which it is associated. Their composition, got by joining points in common, meets all the relations of the components, and so satisfies the conjunction:

$$ptran(\alpha, \beta, \gamma) \wedge ntran(\beta, \delta, \gamma) \wedge Pow\ \alpha \wedge Gnd\ \delta$$

For the inverter we wish to hide the points α and δ from the environment. This is achieved by existential quantification to give the following assertion for the CMOS inverter:

$$\mathrm{Inv}(\gamma, \beta) \equiv \exists V\alpha \exists V\delta.\ ptran(\alpha, \beta, \gamma) \wedge ntran(\beta, \delta, \gamma) \wedge Pow\ \alpha \wedge Gnd\ \delta$$

The CMOS inverter is intended to implement the specification:

$$\mathrm{Spec}(\gamma, \beta) \equiv (V\gamma = H \rightarrow V\beta = L) \wedge (V\gamma = L \rightarrow V\beta = H)$$

A simple proof using well-known rules of logic shows

$$\mathrm{Inv}(\gamma, \beta) \rightarrow \mathrm{Spec}(\gamma, \beta).$$

We can prove the assertion $\mathrm{Inv}(\gamma, \beta)$ implies $\mathrm{Spec}(\gamma, \beta)$. In this sense we can prove $\mathrm{Inv}(\gamma, \beta)$ implements $\mathrm{Spec}(\gamma, \beta)$. To recap: We have described the behaviour of circuits by assertions, their composition by conjunction, hiding of points by existential quantification and taken implementation as implication between the assertions for the circuit and its specification.

This general scheme can be followed equally well for dynamically changing circuits subject to voltages which change over time [Gor2, 3]. The only change is to model a

circuit as a relation between histories of voltage values at the points of interest (histories are functions from time $0, 1, \cdots, t, \cdots$ to $\{H, L\}$) taking *e.g.* the assertion for a p–type transistor to be

$$ptran(\alpha, \beta, \gamma) \equiv \forall t.\ V(\gamma, t) = L \rightarrow V(\alpha, t) = V(\beta, t).$$

The approach is noteworthy because it does not impose an unrealistic directionality on devices as would for example be forced in the approach [Sh] were it to tackle such low–level circuits. The model and logic are also compositional; one can reason about the behaviour of circuits in terms of the behaviour of their components. A similar scheme is followed in [Mos], but there assertions in a temporal logic are used instead.

Unfortunately, there is a major deficiency in this approach—natural and useful as it is in many examples. According to this scheme a "short circuit" implements any specification! A short circuit, achieved most simply, by joining power and ground together at a point α is described by the assertion

$$Pow\ \alpha \wedge Gnd\ \alpha$$

But $Pow\ \alpha \wedge Gnd\ \alpha$ is equivalent to $V\alpha = H \wedge V\alpha = L$, and because H and L are assumed unequal, this is equivalent to ff, the logical value false. Thus $Pow\ \alpha \wedge Gnd\ \alpha$, like ff, implies every assertion and so "implements" any specification whatsoever.

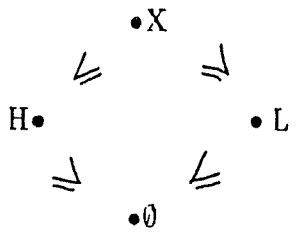
Several ways have been suggested to get around this undesirable situation. Sticking with the above method of modelling circuits by assertions, the only course is to use some other notion of implementation. In some contexts it appears reasonable to say a circuit implements a specification if the associated assertions are equivalent (see [Gor2]). But following that line would, in general, lead to unnecessarily detailed specifications. Another suggestion, discussed in [CGM], is to say a circuit $\mathrm{circ}(\iota, o)$ with input values $V\iota$ and output values Vo, correctly implements a specification $\mathrm{spec}(\iota, o)$ iff

$$(\forall V\iota,\ Vo.\ \mathrm{circ}(\iota, o) \rightarrow \mathrm{spec}(\iota, o))\ \wedge (\forall V\iota \exists Vo.\ \mathrm{circ}(\iota, o)).$$

A circuit meeting such a requirement cannot be equivalent to ff for any particular input value. But this solution depends on having clearly defined input and output points. It is hard to see, if this were so at every stage in constructing a design, how the method of construction could allow short–circuits to be formed, and any method which bans short–circuits outright is too restrictive to provide a general model. Another possibility, suggested in [F], is to use the power of higher order logic and make specifications of higher type than circuit behaviours. This proposal has promise but has not yet led a calculus for reasoning about circuits. Most significant of all, each of these suggestions fails to face the fact that voltage values other than high and low can appear in designs, often quite innocently, without trivialising their behaviour.

We look for a model of circuits which can handle voltage values other than just high and low, and in particular treat short circuits. When a source of power and ground are connected together they give rise to a voltage which has an indeterminate effect when applied to the gates of transistors. We can take a voltage to have value X when it lies

in a region between those corresponding to H and L. Similarly we can take a voltage at a point to have a value $\emptyset$ when the point is not connected to any significant sources of charge. It will be useful to order the values $H, L, X, \emptyset$ as



(Note this order is not directly related to the underlying order on voltages, measured as reals.) A point connected just to power takes the value H, just to ground the value L, to both the value X and to no sources the value $\emptyset$. Can our earlier ideas be adapted to cope with these extra values? For the composition of assertions to be conjunction we now require

$$Pow\ \alpha \equiv V\alpha \geq H,$$
$$Gnd\ \alpha \equiv V\alpha \geq L.$$

In this way we allow for the effect of the environment on the value at α. Then the conjunction $Pow\ \alpha \wedge Gnd\ \alpha$ is equivalent to $V\alpha = X$, as required, and not to ff. Unfortunately hiding can no longer be treated as simple $\exists$-quantification. For example, connecting and hiding a power source to the gate γ of an n-type transistor $ntran(\alpha, \beta, \gamma)$ should yield a circuit equivalent in behaviour to a wire between α and β. However $ntran(\alpha, \beta, \gamma)$ is still described by

$$ntran(\beta, \delta, \gamma) \equiv (V\gamma = H \rightarrow V\beta = V\delta),$$

for the same reasons as before, so using $\exists$-quantification to hide we obtain

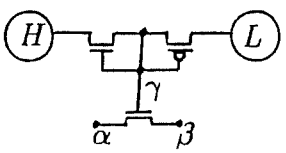
$$\exists V\gamma.\ Pow\ \gamma \wedge ntran(\alpha, \beta, \gamma)$$

which is

$$\exists V\gamma.\ V\gamma \geq H \wedge (V\gamma = H \rightarrow V\alpha = V\beta).$$

This does not imply $V\alpha = V\beta$, the assertion we would like, because of the possibility that $V\gamma = X$.

The above example might suggest that when we hide a node the value associated with the node should be the least possible—to rule out the possibility that $V\gamma = X$ in the example above. However, there are examples where this does not work. Consider hiding γ in the following circuit:

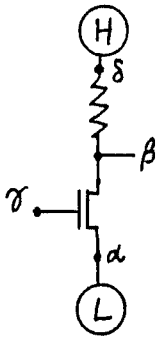


It is unclear whether we should hide γ with $V\gamma$ taking the value H, L, X or $\emptyset$—all are possible.

One way to manage the new value X and still treat hiding by $\exists$–quantification is to impose enough directionality on devices so they can be modelled as functions. Again, this suffers the drawback of leading to an unrealistic model. There is no easy fix without complicating the relational model. We need somehow to express the fact that the values associated with a hidden point are precisely those which are maintained by connections to sources. We shall address this problem and the related one of providing a model which deals correctly with signal strengths.

2. Signal strengths.

In NMOS technology p–type transitors are not available so inverters are constructed in a different way. An NMOS inverter has the following design



The design uses a strong resistance

connected to power. In NMOS this is implemented as a pull–up transistor. The behaviour of the inverter is remarkably subtle for its size (see [MC, MD]). In those environments where the point β is not connected to any other sources, the inverse of the value $V\gamma$, input at γ, is output as $V\beta$ at β, so as a makeshift description of its behaviour we can take:

$$V\gamma = L \rightarrow V\beta = H \wedge$$
$$V\gamma = H \rightarrow V\beta = L.$$

Later in section 4, after we have given the semantics of circuits and enriched the class of assertions, we can provide a complete description of its behaviour and, in particular, see what value $V\beta$ takes for input $V\gamma = X$. The informal English description of how the inverter works is sometimes given as follows:

> When γ is low the n–type transistor disconnects so the only voltage contribution to β is from the power source so β is high.
>
> When γ is high the transistor connects so there is a voltage contribution of low from ground and a voltage contribution of high from power, weakened however by the large resistor, so the contribution from ground dominates and the net effect is to make β low.

Though this argument is perhaps not convincing at first, it can be justified by a simple application of Ohm's law. Suppose γ is high. Then α and β are connected with a very small resistance r relative to the large resistance R between β and δ. Let $v_\alpha, v_\beta, v_\delta$ be the voltage values (real numbers) at the corresponding points. By Ohm's law we see

$$\frac{v_\beta - v_\alpha}{v_\delta - v_\beta} = \frac{Ir}{IR} = \frac{r}{R}$$

which we have assumed is very small. Because v_α is safely inside the half–open interval qualifying as high, taking R large enough ensures v_β is high too.

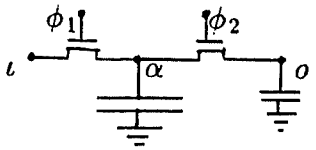
It is often convenient to use the inverse notion of conductance instead of resistance. In the literature (*e.g.* [B, Hay]), sources are pictured as transmitting signals to points of a circuit. The signals not only have a value—$H, L, X, \emptyset$—but also a strength depending on the conductance strength of the path along which the signal has travelled. These ideas generalise when signals from capacitance, another source of charge, are considered.

The NMOS inverter illustrates a principle used in circuit design:

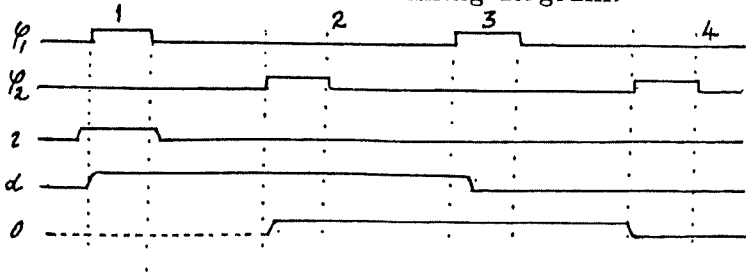
> A signal from power or ground via a strong conductance overrides a signal via a relatively weak conductance.

It is not unusual for designs to use three ranks of conductance strengths, a signal via a conductance in one rank is overridden by a signal via a conductance in a rank strictly above it.

Extra signal strengths arise from capacitance. A simple dynamic register takes the form



in which use is made of the high capacitance cap^α at the node α. The points ϕ_1 and ϕ_2 are connected to clocks which alternately send pulses of high and low voltage. They are out of phase as shown in the timing diagram below. If a signal, say from power, is present at a pulse of ϕ_1, then the left transistor connects and the right transistor disconnects and whatever charge was stored at cap^α is overridden by the current supplied from ι. This assumes the clock pulse of ϕ_1 is long enough for the opposite charge stored in cap^α to drain away. Then when ϕ_1 goes low both transistors are disconnecting and cap^α stores a high voltage. This is delivered at o at the next rise of the clock ϕ_2. Assuming node α has a very high capacitance relative to o, the net effect is to produce a high voltage at o. This is illustrated in the timing diagram.



More succinctly we can describe the behaviour by

$$\forall t.\ V(o, t \div 1) = V(\iota, t)$$

where we take a discrete model of time with the high pulses of ϕ_1 and ϕ_2 corresponding to alternate numbers.

We used two principles to explain the behaviour of the dynamic register:

> A signal from power or ground overrides a signal from a capacitance.
>
> A signal from a large capacitance overrides a signal from a relatively weak capacitance.

A circuit design may involve a range of signal strengths which give a discrete measure of the current driving capability in the analogue circuit. The strength order is derived from a deliberately crude ranking of resistances and capacitances. Two resistances R and R' are assigned conductance strengths g, g' respectively, with $g < g'$, if the ratio R/R' is *very large*. Then, arguing by Ohm's law, as we did before, if two resistances R and R' of strength g and g' are connected in series the resulting resistance $R + R'$ should be assigned conductance strength $g \cdot g'$, the *minimum* of g and g'. This is because, for example, if R/R' is very large then so is $(R + R')/R'$. Connected in parallel their resulting resistance, $RR'/(R + R')$, should be assigned strength $g + g'$, the *maximum* of g and g'. We would like to conclude that whenever we encounter a chain of resistances, of strengths $g_1, g_2, \ldots, g_n$, in series then we can regard it as equivalent to a single resistance of strength $\Pi\{g_1, \ldots, g_n\}$, the minimum strength along the chain. We cannot quite do this because "very large" is a vague concept; even though R/R' is very large R/nR' need not be. The point is that with respect to a particular design a very large number L can be chosen (to stand for "very large") so that the number of times resistances are placed in series or parallel *in the design* has no significant effect. Still, we should bear in mind that such problems can arise through our choosing to work with an abstract strength order, and that without care they can lead to inaccuracies in the model.

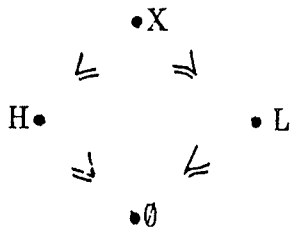
In a similar way capacitances are ranked in a total order of capacitance strengths. The signal stored by a capacitance of strength k is overridden by one from a capacitance of strength k' with $k < k'$. As we have observed, signals from sources override those due to capacitance so we arrive at the concept of a *strength order* as consisting of two finite sets $K = \{k_1, \ldots, k_m\}$ and and $G = \{g_1, \ldots, g_n\}$ in a total order

$$0 < k_1 < \cdots < k_m < g_1 < \cdots < g_n < \infty.$$

We use 0, zero strength, to stand for a strength of a negligible signal, from zero capacitance or a non–conductance, and ∞ to be the strength of a signal from a source via a perfect conductance. The restriction $\mathbf{K} = (K \cup \{0\}, \leq)$, is called the *capacitance order* and its elements are called *capacitance strengths*. The restriction $\mathbf{G} = (G \cup \{0, \infty\}, \leq)$ is called the *conductance order*, with elements called *conductance strengths*. Often we shall write a strength order as $\mathbf{S}$, and sometimes, when we wish to emphasise the conductance and capacitance strengths as $\mathbf{S}_{K,G}$. We shall use $s \cdot s'$ for the minimum and

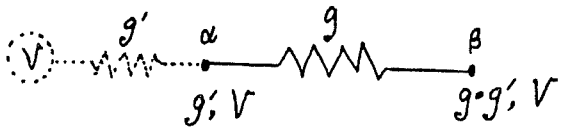
$s + s'$ for the maximum of two strengths s and s'. We can write the minimum of a set of strengths A as ΠA and their maximum as ΣA.

Recall the lattice of values:

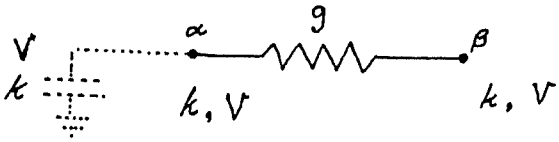


The value $\emptyset$ is associated with points which are at 0 strength so they are not connected to any significant sources of current or charge. Sometimes it is said that such points "float" because they adopt, or float to, the value of whatever they are connected to. We can call the $\emptyset$ value *floating*, though sometimes we use it in contexts where the intuition suggesting this name is not appropriate. Values now form a finite lattice. We use $U + U'$ for the join (or least upper bound) of two values U, U', $U \cdot U'$ for their meet. and ΣA, and ΠA, for the join, respectively meet, of a set of values A. (Our notation is thus consistent with that for the strength order considered as a lattice.)

Consider how signals are transmitted through a resistance between α and β of strength g. Suppose a signal of strength g', a conductance strength, and value $V \in \{H, L\}$ is applied to one end α of the resistance. By our earlier observation concerning resistances in series, the resulting signal at β has strength $g \cdot g'$ and value V.



If instead a signal of strength k, a capacitance strength, and value V is applied at α. assuming the resistance has negligible capacitance, the resulting signal at β is the same, with strength still k and value V.



In both situations a signal of strength s and value V, applied at α gives rise to a signal of strength $s \cdot g$ and value V at β. A signal of strength $s > g$ is cut–down to a signal of strength g while a signal of strength $s \leq g$ is unchanged.

Notice that these assumptions are dependent on connections lasting long enough for the charges in capacitances to reach stable levels. We shall have cause to examine this assumption more carefully later in the conclusion.

We have said that directionality in circuits should be derived rather than imposed. Where is this directionality to come from? It comes from the effects of resistance. Consider a resistance of conductance strength g between points α and β in some environment in which the strength of the signal at α is $S\alpha$ and that at β is $S\beta$. Assume too

that the associated values are $V\alpha$ and $V\beta$ in $\{H, L, X, \emptyset\}$. We know from the behaviour of resistances, considering the transmission of signals from α to β that

$$S\alpha \cdot g \leq S\beta$$

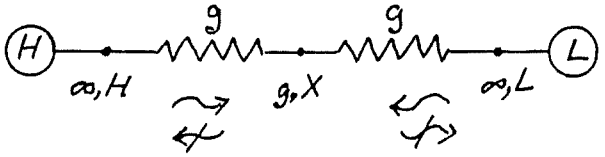
(and similarly that $S\beta \cdot g \leq S\alpha$).
In the case when $S\alpha \cdot g < S\beta$ the signal from α is overridden and so has no effect on that at β. On the other hand, if $S\alpha \cdot g = S\beta$ then the signal from α is not overridden and has an effect on that at β. Values "flow" from α to β and so $V\alpha \leq V\beta$. In particular if $V\alpha = H$ then $V\beta = H$, or X if β happens also to be connected to ground. There is a connection, possibly one–way, between α and β, and we can write this suggestively as $\alpha \rightsquigarrow \beta$. We have $\beta \rightsquigarrow \alpha$ as well only if $S\beta \cdot g = S\alpha$. It is easy to check that elements of this "flow relation" compose, so if $\alpha \rightsquigarrow \beta$ and $\beta \rightsquigarrow \gamma$ then $\alpha \rightsquigarrow \gamma$. For this more general understanding of the flow relation we still have

$$\alpha \rightsquigarrow \beta \rightarrow V\alpha \leq V\beta.$$

It is helpful to think of the strength function S as giving a "height" of each point, and indeed

$$\alpha \rightsquigarrow \beta \rightarrow S\beta \leq S\alpha,$$

so flow is never "uphill". This understanding accounts for the following assignment of flows, strengths and values:



It is the flow relation, rather than the graph of conductances, which plays the central control in analysing the behaviour of circuits.

This exposition has been based largely on Bryant's work (see [B]). The paper [Hay] deals with similar ideas but the model it presents seems only to apply in situations where the components can be understood as functions with definite input and output ports. Our "flow relation" corresponds to Bryant's idea of "unblocked path" in a steady state. Note our subsequent presentation will be markedly different than Bryant's. This is because we shall develop a compositional model, one on which we can base a proof system structured by the way circuits are built–up.

3. States of circuits—static configurations.

Assume a particular strength order $\mathbf{S} = \mathbf{S}_{K,G}$.

We explain the intuition behind the definition of static configuration. Imagine a circuit connected to some environment via points Λ. Assume that in this environment the circuit has settled into a steady state. The definition of static configuration is intended to formalise this notion, picking out those features essential for the compositional account of circuit behaviour that follows. Note here we ignore the possibility that a circuit may never settle into a steady state in an environment. This model of circuits can be seen as analogous to those models of programs which only capture their partial correctness.

In a static configuration each point of a circuit is associated with a signal with a certain strength and value. So each point α is associated with with a strength $S\alpha \in \mathbf{S}$ and a resultant value $V\alpha \in \mathbf{V}$. Some of this signal may be contributed by sources inside the circuit; the internal contribution at α can be recorded by a value $I\alpha \in \mathbf{V}$.

Of course points are connected to each other according to the state of transistors and the connections have certain conductance strengths. This gives rise to a flow relation $\rightsquigarrow$ between points, though in the rather indirect way we saw in the last section. It is this relation, derived from the more basic and detailed conductance relation between points, which plays the central role in our model of the behaviour of circuits. Intuitively, the relation $\rightsquigarrow$ captures the flow of information in a circuit; it expresses how the values of signals flow (or are transmitted) from points at high strength to points at lower or equal strength along flows of conductance.

3.1 Definition. Let Λ be a set of points. A *static configuration* of sort Λ is a 4-tuple

$$\langle S, V, I, \rightsquigarrow \rangle$$

where

$S : \Lambda \to \mathbf{S}$ (the *strength function*),
$V : \Lambda \to \mathbf{V}$ (the value function),
$I : \Lambda \to \mathbf{V}$ (the *internal* value function) and
$\rightsquigarrow$ is a reflexive, transitive relation on Λ (the *flow relation*),

which satisfy

(i) $\alpha \rightsquigarrow \beta \to S\alpha \geq S\beta$,
(ii) $\alpha \rightsquigarrow \beta \wedge S\alpha = S\beta \to \beta \rightsquigarrow \alpha$,
(iii) $\alpha \rightsquigarrow \beta \wedge S\beta \in K \to S\alpha = S\beta$,
(iv) $S\alpha = 0 \leftrightarrow V\alpha = \emptyset$,
(v) $I\alpha \leq V\alpha$,
(vi) $\alpha \rightsquigarrow \beta \to I\alpha \leq I\beta \wedge V\alpha \leq V\beta$.

Write $\mathrm{sort}(\sigma)$ for the sort of a configuration σ. Write $\mathrm{Sta}_S[\Lambda]$ for the set of static configurations of sort Λ. We say a static configuration is *finite* when it has finite sort.

Property (i) says a signal cannot flow from a point at weaker strength to a point at stronger strength. Property (ii) expresses the fact that if $\alpha \rightsquigarrow \beta$, so information can

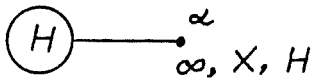
flow from α to β, and α and β are at the same strength then $\beta \rightsquigarrow \alpha$, so information can flow in the other direction too. To justify (ii) assume that $\alpha \rightsquigarrow \beta$ and $S\alpha = S\beta$. Then, according to the last section, $\alpha \rightsquigarrow \beta$ arises iff there is a conductance, of strength g say, between α and β so that $S\alpha \cdot g = S\beta$. Hence $S\beta \cdot g = S\alpha$ too making $\beta \rightsquigarrow \alpha$. Property (iii) states that if $\alpha \rightsquigarrow \beta$ and $S\beta$ is a capacitance strength then $S\alpha$ is the same strength. This follows capacitance strengths are always ranked below those of conductance in the strength order. Assume $\alpha \rightsquigarrow \beta$ and $S\beta \in K$. Then $S\alpha \cdot g = S\beta$ for some conductance strength g for which $S\beta \leq g$. This can only occur with $S\alpha = S\beta$. Property (iv) says a signal of no strength has no content and *vice versa*. Naturally the internal contribution cannot exceed the resultant value—property (v). The final property (vi) formalises the intention that $\rightsquigarrow$ should represent the direction in which information is transmitted through the circuit.

The strength and value functions are used later to specify when two static configurations can sensibly be linked together in parallel. It is necessary to keep track of the internal contribution to the value function and flow relation to give a satisfactory treatment of hiding. They determine when a point may be insulated from the environment without changing its resultant signal.

To make these somewhat abstract ideas a little clearer we present some static configurations of basic devices.

3.2 Example. *A source:*

A source Pow^{α} supplies an internal contribution of strength ∞ and value H to a point α which may well receive a contribution of L from the environment to yield a resultant value $V\alpha = X$.



$$S\alpha = \infty \wedge V\alpha = X \wedge I\alpha = H$$

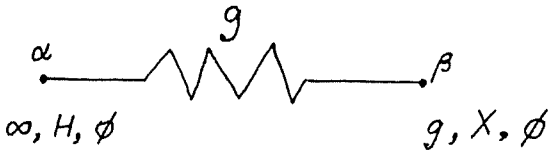
3.3 Example. *A wire:*

A resistance of perfect conductance $res_{\infty}^{\alpha,\beta}$ can be regarded as a wire between α and β in which signals flow unimpaired between the two points. There are no internal sources.

$$S\alpha = S\beta = s \wedge V\alpha = V\beta = U \wedge I\alpha = I\beta = \emptyset \wedge \alpha \rightsquigarrow \beta \wedge \beta \rightsquigarrow \alpha$$

3.4 Example. *A resistance:*

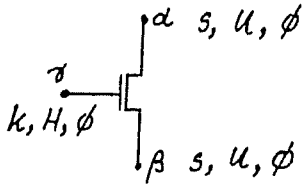
A resistance of strength g contains no internal sources. If (in the environment) power and no other source is applied at α and β is connected to ground via conductance g then the static configuration shown would result.



$$S\alpha = \infty \wedge S\beta = g \wedge$$
$$V\alpha = H \wedge V\beta = X \wedge$$
$$I\alpha = I\beta = \emptyset \wedge$$
$$\alpha \rightsquigarrow \beta \wedge \neg\beta \not\rightsquigarrow \alpha.$$

3.5 Example. *A transistor:*

If an n–type transistor $ntran^{\alpha,\beta,\gamma}$ is placed in an environment in which positive charge is applied to the gate γ then α and β are connected and behave like a wire.



$$S\alpha = k \wedge S\alpha = S\beta = s \wedge$$
$$V\alpha = V\beta = U \wedge$$
$$I\alpha = I\beta = I\gamma = \emptyset \wedge$$
$$\alpha \not\rightsquigarrow \beta \wedge \beta \not\rightsquigarrow \alpha \wedge$$
$$\neg(\alpha \not\rightsquigarrow \gamma \vee \gamma \not\rightsquigarrow \alpha) \wedge \neg(\beta \not\rightsquigarrow \gamma \vee \gamma \not\rightsquigarrow \beta).$$

In a static configuration, two points α and β may be both in the relation $\alpha \not\rightsquigarrow \beta$ and $\beta \not\rightsquigarrow \alpha$. In this case the points α and β receive the same signals both in value and strength. Sometimes neither $\alpha \not\rightsquigarrow \beta$ nor $\beta \not\rightsquigarrow \alpha$—the two points are incomparable—with respect to the flow relation $\not\rightsquigarrow$. Intuitively this means that the signal at one point does not affect the signal at the other. We introduce notation to describe these circumstances.

3.6 Definition. Let $\sigma = \langle S, V, I, \not\rightsquigarrow \rangle$ be a static configuration of sort Λ. For $\alpha, \beta \in \Lambda$ define

$$\alpha \sim \beta \equiv \alpha \not\rightsquigarrow \beta \wedge \beta \not\rightsquigarrow \alpha$$
$$\alpha \parallel \beta \equiv \neg(\alpha \not\rightsquigarrow \beta) \wedge \neg(\beta \not\rightsquigarrow \alpha).$$

In the special case where the effects of resistance are negligible there is only only one positive conductance strength ∞, corresponding to perfect conductance, the flow

relation $\nrightarrow$ coincides with the equivalence relation $\sim$. The proof is a little exercise in using the axioms which define a static configuration.

3.7 Proposition. *Assume there are only the two conductance strengths $0 < \infty$. Then for any static configuration σ the relation $\nrightarrow$ is symmetric and so an equivalence relation on points with $x \nrightarrow y$ iff $x \sim y$ for all points x, y of σ.*

Proof. Suppose $\alpha \nrightarrow \beta$.

Assume $S\beta = 0$. Then $S\alpha = 0$ by (iv) and (vi). Assume $S\beta \in K$. Then $S\alpha = S\beta$ by (iii). Assume the remaining case that $S\beta = \infty$. Then $S\alpha = \infty$ by (i). Therefore, as in all cases $S\alpha = S\beta$, we see $\beta \nrightarrow \alpha$ by (ii).

Thus $\alpha \nrightarrow \beta$ implies $\beta \nrightarrow \alpha$, making $\nrightarrow$ symmetric. By definition $\nrightarrow$ is reflexive and transitive, so $\nrightarrow$ is an equivalence relation. Hence the relations $\nrightarrow$ and $\sim$ are equal. ∎

Proposition 3.7 may increase our level of confidence in the axioms proposed for a static configuration. Still there are sufficiently many axioms, and the idea of static configuration sufficiently complicated, to raise the question of their completeness. We have used arguments from physics for the soundness of the axioms. To show completeness we need an argument showing that there are no properties shared by all static configurations of real (or buildable) circuits which do not follow from those written down in the definition? Afterall, axiom (ii) does not spring to mind immediately from idea of static configuration or the examples we have considered. Later in proposition 4.4 we shall show that every structure $\langle S, V, I, \nrightarrow \rangle$ on a finite set of points which satisfies all the axioms of 3.1 can be realised as the static configuration of a circuit built-up from resistances and sources connected to those points. Then any property of structures $\langle S, V, I, \nrightarrow \rangle$ which holds of all static configurations of circuits must also hold of all finite structures in 3.1.

We make the static configurations with sorts Λ a subset of some universe of points Π into a partial algebra with operations associated with composition and hiding.

3.8 Notation. Let L be a complete lattice ordered by $\leq$ with binary join $+$ and arbitrary join Σ. Let M and Λ be sets. Let $f : M \to L$ and $g : \Lambda \to L$ be functions to the lattice. We define their join $f + g : M \cup \Lambda \to L$ by

$$(f+g)(x) = \begin{cases} f(x) & \text{if } x \in M \setminus \Lambda \\ g(x) & \text{if } x \in \Lambda \setminus M \\ f(x) + g(x) & \text{if } x \in M \cap \Lambda. \end{cases}$$

for $x \in M \cup \Lambda$.

If $f : M \to L$ we write $f \lceil \Lambda$ for the restriction of f to the subset $M \cap \Lambda$. If R is a binary relation on M we write $R \lceil \Lambda = R \cap (\Lambda \times \Lambda)$ for its restriction.

Let R be a binary relation on M. Let $f : \Lambda \to L$ be a function from $\Lambda \subseteq M$ to the lattice L. Define the *application* of relation R to f to be the function $(R \,.\, f) : M \to L$ given by

$$(R \,.\, f)(x) = \Sigma\{f(z) \mid z \in \Lambda \;\&\; (z, x) \in R\}.$$

We shall use this operation to transmit the values at a subset of points in a static configuration to other points in accord with the flow relation $\rightsquigarrow$.

Assume σ_0 is a static configuration of a circuit c_0 and σ_1 is a static configuration of a circuit c_1. When can σ_0 and σ_1 be composed to give a static configuration of c_0 and c_1? When their strengths and values agree at common points; only then do σ_0 and σ_1 make consistent assumptions about the environment. Then the resulting flow relation should be the transitive closure of the flow relations in the components and the internal contribution should be spread out accordingly.

3.9 Definition. Let $\sigma_0 = \langle S_0, V_0, I_0, \rightsquigarrow_0 \rangle$ be a static configuration of sort Λ_0 and $\sigma_1 = \langle S_1, V_1, I_1, \rightsquigarrow_1 \rangle$ be a static configuration of sort Λ_1. Define their *composition* to be

$$\sigma_0 \bullet \sigma_1 = \begin{cases} \langle S, V, I, \rightsquigarrow \rangle & \text{if } S_0\lceil\Lambda_1 = S_1\lceil\Lambda_0 \text{ and} \\ & \quad V_0\lceil\Lambda_1 = V_1\lceil\Lambda_0 \\ \text{undefined} & \text{otherwise} \end{cases}$$

where

$$\begin{aligned} S &= S_0 + S_1, \\ V &= V_0 + V_1, \\ \rightsquigarrow &= (\rightsquigarrow_0 \cup \rightsquigarrow_1)^* \quad \text{and} \\ I &= \rightsquigarrow .\ (I_0 + I_1). \end{aligned}$$

Suppose σ is a static configuration of a circuit c. When does σ restrict to a configuration of a circuit like c but in which all points but those in Λ are hidden? When all the points to be hidden have values (and strengths) which result from the combined effect of internal sources and the contribution from unhidden points Λ. More precisely when for all points α to be hidden we have $V\alpha = I\alpha + (\rightsquigarrow .\ V\lceil\Lambda)\alpha$. Then the hiding of points not in Λ will make no difference.

3.10 Definition. Let $\sigma = \langle S, V, I, \rightsquigarrow \rangle$ be a static configuration of sort $\mathcal{M}$ and Λ a set of points. Define the *restriction* of σ to Λ to be

$$\sigma\lceil\Lambda = \begin{cases} \langle S\lceil\Lambda,\ V\lceil\Lambda,\ I\lceil\Lambda, \rightsquigarrow\lceil\Lambda \rangle & \text{if } V = I + (\rightsquigarrow .\ V\lceil\Lambda) \\ \text{undefined} & \text{otherwise.} \end{cases}$$

3.11 Notation. We indicate $\sigma_0 \bullet \sigma_1$ and $\sigma\lceil\Lambda$ are defined by writing $\sigma_0 \bullet \sigma_1 \downarrow$ and $\sigma\lceil\Lambda \downarrow$.

4. The semantics of static circuits.

Assume the strength order is $\mathbf{S} = \mathbf{S}_{K,G}$. Also assume a countably infinite set of *point names* Π.

4.1 Definition. A little language for static circuits—circ.
The syntax of circ is given by:

$$c ::= Pow\ \alpha \mid Gnd\ \alpha \mid cap_{kU}\alpha \mid res_g(\alpha,\beta) \mid ntran(\alpha,\beta,\gamma) \mid ptran(\alpha,\beta,\gamma) \mid c \bullet c \mid c\lceil\Lambda$$

where $k \in K$, $\emptyset \neq U \in \mathbf{V}$, $g \in G \cup \{\infty\}$ and α,β,γ are distinct point names in Π and $\Lambda \subseteq \Pi$.

The constant terms $Pow\ \alpha$ and $Gnd\ \alpha$ stand for sources providing a signal of strength ∞ at point α with values H and L respectively. Another kind of source arises through charge storage, when the strength s is an element of K. A term $cap_{kU}\alpha$ represents a capacitance of strength k, charged up with value U. The constant term $res_g(\alpha,\beta)$ stands for a resistance connecting points α and β with conductance $g \in G$. The constant $ntran(\alpha,\beta,\gamma)$ stands for an n–type transistor with a gate γ which when it is high connects points α and β with perfect conductance. The constant term $ptran(\alpha,\beta,\gamma)$ stands for a p–type transistor with a gate γ which connects points α and β, again with perfect conductance, when the gate γ is low. Of course, should the effect of resistance be significant we can insert a suitable resistance between α and β.

We take the behaviour of a circuit in circ to be the set of possible static configurations it can settle into in some static environment. For compactness, in the following definition we assume that a static configuration σ is $\langle S, V, I, \nrightarrow\rangle$.

4.2 Definition. The semantics of circ.
Let $\mathrm{Sta} = \bigcup_{\Lambda\subseteq\Pi}\mathrm{Sta}[\Lambda]$, the set of static configurations with sorts which are subsets of Π.

Define the semantic function $[\![\ ]\!] : \mathbf{circ} \to P(\mathrm{Sta})$ to be the map defined by the structural induction

$$[\![Pow\ \alpha]\!] = \{\sigma \in \mathrm{Sta}[\alpha] \mid S\alpha = \infty \wedge I\alpha = \mathrm{H}\}$$

$$[\![Gnd\ \alpha]\!] = \{\sigma \in \mathrm{Sta}[\alpha] \mid S\alpha = \infty \wedge I\alpha = \mathrm{L}\}$$

$$[\![cap_{kU}\alpha]\!] = \{\sigma \in \mathrm{Sta}[\alpha] \mid S\alpha \geq k \wedge (S\alpha = k \to I\alpha = U) \wedge (S\alpha > k \to I\alpha = \emptyset)\}$$

$$[\![res_g(\alpha,\beta)]\!] = \{\sigma \in \mathrm{Sta}[\alpha,\beta] \mid I\alpha = \emptyset \wedge I\beta = \emptyset \wedge S\alpha \cdot g \leq S\beta \wedge S\beta \cdot g \leq S\alpha \wedge (S\alpha \cdot g = S\beta \leftrightarrow \alpha \nrightarrow \beta) \wedge (S\beta \cdot g = S\alpha \leftrightarrow \beta \nrightarrow \alpha)\}$$

$$[\![ntran(\alpha,\beta,\gamma)]\!] = \{\sigma \in \mathrm{Sta}[\alpha,\beta,\gamma] \mid I\alpha = \emptyset \wedge I\beta = \emptyset \wedge I\gamma = \emptyset \wedge \gamma \parallel \alpha \wedge \gamma \parallel \beta \wedge (\alpha \parallel \beta \vee \alpha \sim \beta) \wedge (V\gamma = \mathrm{H} \rightarrow \alpha \sim \beta) \wedge (V\gamma = L \rightarrow \alpha \parallel \beta)\}$$

$$[\![ptran(\alpha,\beta,\gamma)]\!] = \{\sigma \in \mathrm{Sta}[\alpha,\beta,\gamma] \mid I\alpha = \emptyset \wedge I\beta = \emptyset \wedge I\gamma = \emptyset \wedge \gamma \parallel \alpha \wedge \gamma \parallel \beta \wedge (\alpha \parallel \beta \vee \alpha \sim \beta) \wedge (V\gamma = \mathrm{L} \rightarrow \alpha \sim \beta) \wedge (V\gamma = H \rightarrow \alpha \parallel \beta)\}$$

$$[\![c \bullet d]\!] = \{\sigma \bullet \rho \mid \sigma \in [\![c]\!] \ \& \ \rho \in [\![d]\!] \ \& \ \sigma \bullet \rho \downarrow\}$$

$$[\![c\lceil \Lambda]\!] = \{\sigma\lceil \Lambda \mid \sigma \in [\![c]\!] \ \& \ \sigma\lceil \Lambda \downarrow\}.$$

Further terms can be defined. For example, we can define a wire $wre(\alpha,\beta)$ between α and β, with denotation

$$[\![wre(\alpha,\beta)]\!] = \{\sigma \in \mathrm{Sta}[\alpha,\beta] \mid I\alpha = \emptyset \wedge I\beta = \emptyset \wedge \alpha \sim \beta\},$$

which can be realised by a resistance with perfect conductance, *i.e.*

$$[\![wre(\alpha,\beta)]\!] = [\![res_\infty(\alpha,\beta)]\!].$$

More interestingly, We can define a general source $sce_{sU}\alpha$, at α, of strength s and value $U \in \mathbf{V}$, to have denotation

$$[\![sce_{sU}\alpha]\!] = \{\sigma \in \mathrm{Sta}[\alpha] \mid S\alpha \geq s \wedge (S\alpha = s \rightarrow I\alpha = U) \wedge (S\alpha > s \rightarrow I\alpha = \emptyset)\}$$

Such a source only makes a contribution at α if the strength of α is exactly s. We only consider sources $sce_{sU}\alpha$ for which $s = 0 \Leftrightarrow U = \emptyset$—any others could have no static configurations. If s is a capacitance strength k then such a weakened source can be realised by the charged capacitance cap^{α}_{kU}. If s is a conductance strength g it can be realised by passing signals from ground or power through a resistance of strength g. For example, if $U = H$, we have

$$[\![sce_{gH}\alpha]\!] = [\![(Pow\ \beta \bullet res_g(\beta,\alpha))\lceil \alpha]\!].$$

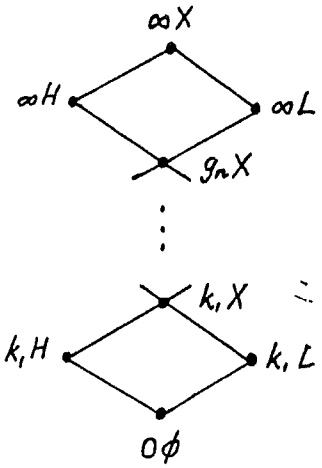
The one remaining case, $sce_{0\emptyset}\alpha$ can be realised as a single connection point standing alone, and this can be made by hiding one end of a resistance, *i.e.*

$$[\![sce_{0\emptyset}\alpha]\!] = [\![res_g(\alpha,\beta)\lceil \alpha]\!].$$

It is helpful to explain the ways sources combine using an ordering between pairs sU, where $s \in \mathbf{S}$ and $U \in \mathbf{V}$ and $s = 0$ iff $U = \emptyset$. On such pairs define

$$sU \leq s'U' \text{ iff } s < s' \text{ or } (s = s' \ \& \ U \leq U').$$

This forms a finite distributive lattice (meet $\cdot$ and join $+$), mentioned in [B1] and [Hay], which may be drawn as:



Now we can list some basic equivalences between circuit terms, which can be proved from the denotational semantics.

4.3 Proposition. *Some equivalences on circuits*

Write $c = c'$ iff $[\![c]\!] = [\![c']\!]$, for circuit terms c, c'.

Realising sources: If k is a capacitance strength and g a conductance strength then

$$\begin{aligned}
sce_{0\emptyset}\alpha &= res_g(\alpha,\beta)\lceil\alpha,\\
sce_{kU}\alpha &= cap_{kU}\alpha,\\
sce_{gH}\alpha &= (Pow\ \beta \bullet res_g(\beta,\alpha))\lceil\alpha,\\
sce_{gL}\alpha &= (Gnd\ \beta \bullet res_g(\beta,\alpha))\lceil\alpha,\\
sce_{gX}\alpha &= sce_{gH}\alpha \bullet sce_{gL}\alpha.
\end{aligned}$$

Composing sources: $sce_{sU}\alpha \bullet sce_{s'U'}\alpha = sce_{sU+s'U'}\alpha$.

Resistances in series and parallel:

$$(res_g(\alpha,\beta) \bullet res_{g'}(\beta,\gamma))\lceil\{\alpha,\gamma\} = res_{g\cdot g'}(\alpha,\gamma).$$
$$(res_g(\alpha,\beta) \bullet res_{g'}(\alpha,\beta)) = res_{g+g'}(\alpha,\beta).$$

Wires, resistances and transistors:

$$\begin{aligned}
wre(\alpha,\beta) &= res_\infty(\alpha,\beta)\\
&= (Pow\ \alpha \bullet ntran(\alpha,\beta,\gamma))\lceil\{\alpha,\beta\}\\
&= (Gnd\ \alpha \bullet ptran(\alpha,\beta,\gamma))\lceil\{\alpha,\beta\}.
\end{aligned}$$

Earlier we pointed out the problem of knowing whether or not we had written down sufficient axioms for static configurations. From the remarks in the last section it is sufficient to show every finite static configuration can be realised as a static configuration of a circuit term. This is established in the following proposition.

4.4 Proposition. *Any static configuration σ of sort Λ a finite subset of Π is the static configuration of some circuit c i.e. $\sigma \in [\![c]\!]$.*

Proof. We sketch the proof. Given σ, we define the required circuit to be the composition of the following finite set of components:

$\{sce_{S\alpha U}\alpha \mid \alpha \in \Lambda \ \& \ I\alpha = U \neq \emptyset\} \cup$
$\{res_g(\alpha,\beta) \mid \alpha \rightsquigarrow \beta \ \& \ g$ is the minimum conductance strength s.t. $S\alpha \cdot S\beta \leq g\}$.

This uses our knowledge of how to build all sources $sce_{sU}\alpha$ as circuits. From the definition of sources, resistances and composition, it can be seen $\sigma \in [\![c]\!]$. ∎

Now we can see how to give a more accurate specification of the NMOS inverter of section 2. Its circuit is constructed by the term

$$c \equiv (ntran(\alpha,\beta,\gamma) \bullet res_g(\beta,\delta) \bullet Pow\ \delta \bullet Gnd\ \alpha)\lceil\{\gamma,\beta\}.$$

Informally, its output β behaves like a direct connection to ground when its input γ is high and like a weakened power source when γ is low. Formally, if we define

$$\begin{aligned} Sce_{sU}x \equiv & Sx \geq s \wedge \\ & Sx = s \rightarrow Ix = U \wedge \\ & Sx = s \rightarrow Ix = U, \end{aligned}$$

then we can say c satisfies the assertion

$$\begin{aligned} & V\gamma = H \rightarrow Sce_{\infty L}\beta \wedge \\ & V\gamma = L \rightarrow Sce_{gH}\beta. \end{aligned}$$

This illustrates how the model is closely associated with assertions for specifying the behaviour of circuits. Of course we should give a more rigorous treatment of their syntax and semantics. This is done in the next section where a complete proof system is presented for proving a circuit satisfies an assertion.

Technically, it will be simpler to work with more basic predicates than Vx and Ix. Say σ satisfies $H\alpha$ if $H \leq V\alpha$, meaning α is connected to a source of positive charge (either in the environment or internal). Similarly, say σ satisfies $L\alpha$ if $L \leq V\alpha$. The assertion $V\alpha = H$ can then be expressed equivalently by $H\alpha \wedge \neg L\alpha$, while $V\alpha = X$ is equivalent to $H\alpha \wedge L\alpha$. For internal signals say σ satisfies $h\alpha$ if $H \leq I\alpha$ and $l\alpha$ if $L \leq V\alpha$. Then, for instance, σ satisfies $h\alpha$ if α is connected to an internal source of positive charge.

To conclude this section, we note we have not completely eliminated the kind of problems raised in the section 1. Certainly we now have an adequate treatment of short circuits. In particular

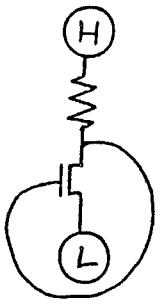
$$[\![Pow\ \alpha \bullet Gnd\ \alpha]\!] \neq \emptyset,$$

and so, when we come to the logic, the circuit $Pow\ \alpha \bullet Gnd\ \alpha$ will not satisfy ff. However, there are other circuit terms which do denote $\emptyset$ and so will satisfy ff. Roughly speaking,

these are terms which represent circuits whose only possible behaviour is to oscillate. For example the term

$$c \equiv (ntran(\alpha, \beta, \gamma) \bullet res_g(\beta, \delta) \bullet Pow\ \delta \bullet Gnd\ \alpha \bullet wre(\gamma, \beta))\lceil \emptyset,$$

with g a conductance strength strictly between 0 and ∞, "ties–back" the output of an NMOS inverter to its input, and then insulates all points from the environment. It can be drawn as:



The circuit c denotes the emptyset, $[\![c]\!] = \emptyset$. (The circuit is a little peculiar in that it has an empty sort. A trivial modification—connecting the wire to the gate of another transistor—produces a term with nonempty sort and empty denotation.) The logic of circuits will thus be akin to the logic of partial correctness assertions (Hoare logic); a purely oscillating circuit will satisfy any assertion just as a diverging program satisfies all partial correctness assertions.

5. A proof system.

We show how to construct a proof system for circuits. In this section we highlight its main features, and refer to the appendix for the full syntax, semantics and proof rules. The proof system consists of a complete set of rules to prove formally that a circuit, described by a term c, satisfies a property described by an assertion A. It is compositional in that proving an assertion holds of a compound circuit of the form $c\lceil\Lambda$ or $c_0 \bullet c_1$ reduces to proving assertions about the components, c of $c\lceil\Lambda$, and c_0 and c_1 of $c_0 \bullet c_1$. Assertions describe the possible static configurations the circuit can settle into. We have seen several examples where a static configuration satisfies some particular logical assertion. Of course, an assertion determines all those configurations which satisfy it, and in the formal treatment we take the meaning, or denotation, of an assertion A, written $[\![A]\!]$, to be the set of all those static configurations which satisfy it. Thus we seek a way to prove relations $[\![c]\!] \subseteq [\![A]\!]$, which we write as $c \models A$, hold between circuit terms c and assertions A. In order to establish such relations it is useful to treat circuit terms as just another kind of assertion in our semantics and proof rules. We write $\sigma \models c$ to mean σ is a static configuration of the circuit c, just as we do for more usual assertions. We make use of relations $\Gamma \models A$ between a set Γ and A where Γ and A are *circuit-assertions* (see L.1 in the appendix) which may include, or be built out of circuit terms. The relation $\Gamma \models A$ means any static configuration σ which satisfies all of Γ satisfies A too. Such relations have their syntactic counterpart in the proof system. The proof rules are written in a natural-deduction style, keeping track of the assumptions in sequents of the form $\Gamma \vdash A$.

We settle on a countably infinite set of point names Π, with typical members $\alpha, \beta, \cdots$, and a fixed strength order $\mathbf{S}$. It is simplest to assume that point names are in 1–1 correspondence with points, so two distinct names cannot be associated with the same point (we shall axiomatise the equality relation between points accordingly). Our previous work suggests the form assertions should take. A static configuration σ, with sort $\Lambda \subseteq \Pi$, is a structure $\langle S, V, I, \rightarrowtail\!\!\!\!\!\!\rightarrow \rangle$ over individuals Λ. We could treat σ as a structure for a first order logic with function symbols S, V, I and relation symbols $\rightarrowtail\!\!\!\!\!\!\rightarrow$ and $=$. We do not quite follow this course. There is first of all the problem that the sorts are not all the same, and in particular we allow the trivial empty static configuration—a source of trouble should we follow a traditional treatment where it is usual to ban empty structures. We could use a family of logics, one for each sort, but instead it is much more convenient and much less messy to use a *free logic* in which it is not necessary that terms are defined, or denote existing things. In the free logic a static configuration σ satisfies $\forall x.A$ when $A[\alpha/x]$ holds for all points α in $\mathrm{sort}(\sigma)$, and similarly σ satisfies $\exists x.A$ if for some α in $\mathrm{sort}(\sigma)$ the assertion $A[\alpha/x]$ is satisfied by σ; quantification is only taken to be over existing, or defined, elements. On the other hand, all variables are understood to range over all potential individuals Π. Our style of free logic is based on [S] and includes an existence predicate E; $E\alpha$ holds in a static configuration precisely when $\alpha \in \mathrm{sort}(\sigma)$. We have constant symbols from Π, a function symbol S, as well as relations $\rightarrowtail\!\!\!\!\!\!\rightarrow$ and $=$ in the logical language. Instead of using functions V and I we use the predicates $H x$, $L x$ and $h x$, $l x$ mentioned in the last section 4.

The *point relations*, *prel*, have the form

$$H\pi \mid L\pi \mid h\pi \mid l\pi \mid \pi_0 = \pi_1 \mid \pi_0 \rightarrowtail\!\!\!\!\!\!\rightarrow \pi_1 \mid E\pi$$

where π, π_0, π_1 range over points and variables.

To reason formally about strengths we need *strength expressions* of the form

$$s \mid S\pi \mid e_0 \cdot e_1 \mid e_0 + e_1$$

where $s \in \mathbf{S}$, π is a point or variable and e_0, e_1 are strength expressions. Note we need an existence predicate for strength expressions as well as points. This is because $S\alpha$ only makes sense, with respect to a static configuration σ, if $\alpha \in \mathrm{sort}(\sigma)$. Define the *strength relations*, *srel*, to have the form

$$Ee \mid e_0 \leq e_1 \mid e_0 = e_1$$

where e, e_0 and e_1 are strength expressions.

Now the *first order assertions* of our free logic are:

$$A ::= \mathit{prel} \mid \mathit{srel} \mid \mathrm{tt} \mid \mathrm{ff} \mid A \wedge A \mid A \vee A \mid A \rightarrow A \mid \exists x.A \mid \forall x.A$$

with atoms which are point and strength relations. The full semantics and proof system can be seen in the appendix (for the moment ignore the second order assertions and rules). Notable, special to a free logic, are the quantifier rules which must take account of existence and the rule (refl) the axiom for existence—the equality $\alpha = \alpha$ only holds of an existing point α. The paper [S] gives an excellent discussion of the axioms and rules.

5.1 Theorem. *A first-order assertion is provable using the proof system in the appendix iff it is satisfied by all static configurations.*

Proof. The proof is omitted. Unfortunately, I do not know an adequate reference for a completeness result in the form we want it, though completeness can be proved rather indirectly from results in [FS] and [Gr]. More direct proofs can be got by following the lines of Henkin's completeness proof for the predicate calculus. ∎

It is as well to get a basic fact about strength relations out of the way. It will be useful to observe that any strength relation can be reduced to ones of a special form. described in the following.

5.2 Proposition. *Let R be a strength relation. There is a propositional assertion A which includes strength assertions solely of the form $S\pi = s$ such that $\vdash R \leftrightarrow A$.*

Proof.

By structural induction,

$$\vdash Ee \leftrightarrow \bigwedge_{\pi \ in \ e} E\pi$$

for any strength expression e, where π ranges over the variables or points mentioned in e. This establishes the proposition for strength relations of the form Ee.

By structural induction,

$$\vdash Ee \leftrightarrow \mathbb{W}_{s \in S} e = s$$

for any strength expression e. It follows from the axioms that

$$\vdash e_0 \cdot e_1 = s \leftrightarrow \mathbb{W}_{s_0, s_1 \in S}(e_0 = s_0 \wedge e_1 = s_1 \wedge s_0 \cdot s_1 = s)$$
$$\vdash e_0 + e_1 = s \leftrightarrow \mathbb{W}_{s_0, s_1 \in S}(e_0 = s_0 \wedge e_1 = s_1 \wedge s_0 + s_1 = s)$$
$$\vdash e_0 \leq e_1 \leftrightarrow \mathbb{W}_{s_0, s_1 \in S}(e_0 = s_0 \wedge e_1 = s_1 \wedge s_0 \leq s_1)$$
$$\vdash e_0 = e_1 \leftrightarrow \mathbb{W}_{s_0, s_1 \in S}(e_0 = s_0 \wedge e_1 = s_1 \wedge s_0 = s_1).$$

Whence, by structural induction on expressions, any strength relation of the form $e_0 \leq e_1$ or $e_0 = e_1$ is provably equivalent to a propositional assertion of the required form. ∎

We are still left with the major problem of how to incorporate rules for reasoning about circuit terms in the logic. As mentioned we can include them in the logic, treating them much the same way as assertions. They are built-up using restrictions $\lceil\Lambda$ and composition $\bullet$ from atoms like $Pow\ \alpha$ and $ntran(\alpha, \beta, \gamma)$. It is a simple matter to incorporate atoms. For example, the rules (L.10) include an elimination rule of the form

$$(Pow\ E) \qquad Pow\ \alpha \vdash S\alpha = \infty \wedge h\alpha \wedge \neg l\alpha \wedge \forall x.x = \alpha$$

whose role is to replace proving a property of a power source $Pow\ \alpha$ to proving a consequence of an assertion expressing its behaviour. But how are we to treat compound terms? Our approach is to associate modal operations, analogous to weakest liberal preconditions, with the operators $\lceil\Lambda$ and $\bullet$. This line was inspired by the general treatment in [A], though, of course, their specific use in the semantics of imperative programming language is well known, largely due to [D]. We also extend the logic to include second-order quantifiers, to obtain the following syntactic category of *second-order assertions*:

$$A ::= prel \mid srel \mid \mathrm{tt} \mid \mathrm{ff} \mid A \wedge A \mid A \vee A \mid A \rightarrow A \mid \exists x.A \mid \forall x.A$$
$$\{x : A\}\pi \mid P\pi \mid \exists P.A \mid \forall P.A \mid$$

Though the reason for this will not become clear until we deal with the preconditions for composition.

The treatment of the restriction operator $\lceil\Lambda$, for a finite subset Λ of Π, is easier to explain first. The use of preconditions arises naturally from the requirement that the proof system be compositional. The requirement of compositionality begs the question:

> What has to be true of a circuit c in order to be guaranteed that $c\lceil\Lambda$ satisfies an assertion A?

And this question amounts to:

> What must be true of a static configuration σ so that if $\sigma\lceil\Lambda$ is defined then $\sigma\lceil\Lambda$ satisfies A?

Rather tautologously, we can answer that σ must satisfy the $\lceil\Lambda$–precondition of A, written ${}^{\Lambda}A$, which has the denotation

$$[\![{}^{\Lambda}A]\!] = \{\sigma \mid \sigma\lceil\Lambda\downarrow \Rightarrow \sigma\lceil\Lambda \models A\}.$$

We might instead of ${}^{\Lambda}A$ write $[\lceil\Lambda]A$ because $[\![{}^{\Lambda}A]\!]$ could have been written equally well as

$$\{\sigma \mid \forall\sigma'.\ \sigma\lceil\Lambda = \sigma' \Rightarrow \sigma' \models A\},$$

making the precondition look even more like a modal operator of the kind used in dynamic logic [Ha]. This precondition is analogous to weakest liberal preconditions because if $\sigma\lceil\Lambda$ is undefined then σ satisfies ${}^{\Lambda}A$. We could have worked instead with the analogue of weakest precondition. We temporarily introduce it as $\langle\lceil\Lambda\rangle A$ with the meaning

$$[\![\langle\lceil\Lambda\rangle A]\!] = \{\sigma \mid \exists\sigma'.\ \sigma\lceil\Lambda = \sigma' \ \&\ \sigma' \models A\},$$

but do not make any further use of it. It turns out to be definable in terms of ${}^{\Lambda}A$, and is not quite as directly useful. For ${}^{\Lambda}A$ we have the following fact:

5.3 Proposition.
Let c be a circuit term and A a second-order assertion. Then

$$c\lceil\Lambda \models A \text{ iff } c \models {}^{\Lambda}A.$$

Proof.

$$\begin{aligned}
c\lceil\Lambda \models A \text{ iff } & [\![c\lceil\Lambda]\!] \subseteq [\![A]\!] \\
\text{iff } & \forall\sigma'.\ \sigma' \models c\lceil\Lambda \Rightarrow \sigma' \models A \\
\text{iff } & \forall\sigma.\ (\sigma \models c \ \&\ \sigma\lceil\Lambda\downarrow) \Rightarrow \sigma\lceil\Lambda \models A \\
\text{iff } & \forall\sigma.\ \sigma \models c \Rightarrow (\sigma\lceil\Lambda\downarrow \Rightarrow \sigma\lceil\Lambda \models A) \\
\text{iff } & \forall\sigma.\ \sigma \models c \Rightarrow \sigma \models {}^{\Lambda}A \\
\text{iff } & [\![c]\!] \subseteq [\![{}^{\Lambda}A]\!] \\
\text{iff } & c \models {}^{\Lambda}A.
\end{aligned}$$

∎

We extend the second–order assertions by preconditions and call the resulting syntactic category simply *assertions*. Proposition 5.3 suggests an obvious elimination rule for preconditions:

$$\frac{c \vdash {}^{\Lambda}A}{c\lceil\Lambda \vdash A}$$

in which the semantic relation of satisfaction has been replaced by the syntactic one of entailment in a sequent calculus. Standing alone this proof rule would not get us very far. True we can eliminate an occurrence of the restriction operator, but only at the expense of introducing a precondition. Fortunately it is a purely mechanical process, captured mainly in the distribution laws for ${}^{\Lambda}(\ )$, listed in L.6, to eliminate all occurrences of $\lceil\Lambda$–preconditions; an assertion containing them can be proved equivalent to an assertion without any.

We explain the rules for ${}^{\Lambda}(\ )$, leaving the detailed proofs of soundness to the reader. Regarding preconditions as modal operators suggests the introduction rule

$$\frac{\vdash A}{\vdash {}^{\Lambda}A}$$

familiar from proof systems with modal operators. The other introduction rule for such preconditions accompanies the fact that if $\sigma\lceil\Lambda$ is undefined then $\sigma \models {}^{\Lambda}A$. Take

$$G \equiv \forall x.\ \neg\Lambda x \to [\mathrm{H}x \to hx \vee (\exists y : \Lambda.\ \mathrm{H}y \wedge y \rightsquigarrow x)] \wedge [\mathrm{L}x \to lx \vee (\exists y : \Lambda.\ \mathrm{L}y \wedge y \rightsquigarrow x)].$$

The assertion G expresses definedness in the sense that

$$\sigma \models G \text{ iff } \sigma\lceil\Lambda\downarrow.$$

The other introduction rule for preconditions is

$$\neg G \vdash {}^{\Lambda}A.$$

The remaining ${}^{\Lambda}(\ )$ rules say how the operator distributes over logical operators. Their role is to enable preconditions to be pushed through and finally eliminated from assertions. Using them we can for instance derive the rule

$$\frac{\Gamma \vdash A}{{}^{\Lambda}\Gamma \vdash {}^{\Lambda}A}$$

where by ${}^{\Lambda}\Gamma$ we mean $\{{}^{\Lambda}B \mid B \in \Gamma\}$, for a finite set of assumptions Γ. From $\Gamma \vdash A$ we derive $\vdash \bigwedge\Gamma \to A$ and hence $\vdash {}^{\Lambda}(\bigwedge\Gamma \to A)$. Using the distribution rules we deduce $G \vdash \bigwedge{}^{\Lambda}\Gamma \to {}^{\Lambda}A$. This gives $G, \bigwedge{}^{\Lambda}\Gamma \vdash {}^{\Lambda}A$ which combined with $\neg G \vdash {}^{\Lambda}A$ yields $\bigwedge{}^{\Lambda}\Gamma \vdash {}^{\Lambda}A$, and finally ${}^{\Lambda}\Gamma \vdash {}^{\Lambda}A$. In particular using this we can for instance show $\vdash {}^{\Lambda}A \leftrightarrow {}^{\Lambda}B$ if $\vdash A \leftrightarrow B$; equivalent assertions have equivalent preconditions. This means that preconditions of strength relations can be replaced by preconditions of assertions of the form given in proposition 5.2 from which all preconditions can be eliminated using the distribution rules.

5.4 Theorem.
Let A be an assertion, which may include restriction preconditions. There is an assertion B, which does not contain any preconditions, such that $\vdash A \leftrightarrow B$.

Proof. By the remarks above, using the distribution laws, we obtain

$$G \vdash {}^{\Lambda}A \leftrightarrow C$$

where C contains no preconditions. But $\neg G \vdash {}^{\Lambda}A$. Therefore $\vdash {}^{\Lambda}A \leftrightarrow (G \to C)$, with $B \equiv (G \to C)$ an assertion of the required form. ∎

Proving $c\lceil\Lambda \models A$ reduces to proving $c \models {}^{\Lambda}A$ which, by the metatheorem and modus ponens, reduces to proving $c \models B$ where B contains no preconditions—and this

can be done formally in the proof system. We have reduced the problem of proving an assertion A holds of $c\lceil\Lambda$ to proving an assertion B holds of c. We have achieved compositionality for restriction.

The term c may well contain compositions using $\bullet$. For our proof system to be compositional we must "decompose" the proof that an assertion A is satisfied by a composition $c_0 \bullet c_1$ to proofs that assertions A_0 and A_1, are satisfied by the components c_0 and c_1, respectively. Problems like this have received a great deal of attention recently (see *e.g.* [deR, OH, St, W]) and our approach throws some light on the problem of obtaining compositional proof systems for parallel processes, and even suggests a general approach. Again the use of a modal operator plays a central role. this time associated with composition. Composition $\bullet$ is a binary operator so the $\bullet$–precondition of an assertion A is satisfied by those pairs of static configurations (σ, ρ) whose composition $\sigma \bullet \rho$, when defined, satisfies A , *i.e.*

$$[\![^\bullet A]\!] = \{(\sigma, \rho) \mid \sigma \bullet \rho \downarrow \Rightarrow \sigma \bullet \rho \models A\}.$$

Thus the assertion $^\bullet A$ is of a different type than those we have encountered previously. It is satisfied by *pairs* of static configurations. To emphasise that it has a different type we call it, and other assertions satisfied by pairs of configurations, product assertions. It is useful to define another operator for forming atomic product assertions. For assertions A and B, take $A \times B$ to be satisfied by those pairs (σ, ρ) where σ satisfies A and ρ satisfies B, *i.e.* so

$$[\![A \times B]\!] = [\![A]\!] \times [\![B]\!].$$

The full syntactic category of *product assertions* is:

$$\begin{aligned} D ::= & A \times B \mid {}^\bullet A \mid E\pi \mid \pi_0 = \pi_1 \\ & \mathrm{tt} \mid \mathrm{ff} \mid D_0 \wedge D_1 \mid D_0 \vee D_1 \mid D_0 \rightarrow D_1 \mid \exists x.D \mid \forall x.D \mid \\ & \{x : D\}\pi \mid \exists P.D \mid \forall P.D \end{aligned}$$

which includes the apparatus of first and second order free logic with equality.

We can treat circuit terms similarly, and with the obvious definition of $\models$ between product assertions, obtain the following proposition relating the $\bullet$–precondition to composition.

5.5 Proposition.
Let c_0 and c_1 be circuit terms and A an assertion. Then

$$c_0 \bullet c_1 \models A \text{ iff } c_0 \times c_1 \models {}^\bullet A.$$

Proof.

$$\begin{aligned} c_0 \bullet c_1 \models A \text{ iff } & [\![c_0 \bullet c_1]\!] \subseteq [\![A]\!] \\ \text{iff } & \forall \sigma'.\ \sigma' \models c_0 \bullet c_1 \Rightarrow \sigma' \models A \\ \text{iff } & \forall \sigma, \rho.\ (\sigma \models c_0 \ \&\ \rho \models c_1 \ \&\ \sigma \bullet \rho \downarrow) \Rightarrow \sigma \bullet \rho \models A \\ \text{iff } & \forall \sigma, \rho.\ \sigma \models c_0 \ \&\ \rho \models c_1 \Rightarrow (\sigma \bullet \rho \downarrow \Rightarrow \sigma \bullet \rho \models A) \\ \text{iff } & \forall \sigma, \rho.\ (\sigma, \rho) \models c_0 \times c_1 \Rightarrow (\sigma, \rho) \models {}^\bullet A \\ \text{iff } & [\![c_0 \times c_1]\!] \subseteq [\![^\bullet A]\!] \\ \text{iff } & c_0 \times c_1 \models {}^\bullet A. \quad \blacksquare \end{aligned}$$

The proposition asserts the soundness of the elimination rule for composition:

$$\frac{c_0 \times c_1 \vdash {}^\bullet A}{c_0 \bullet c_1 \vdash A}$$

We want the proof of $c_0 \times c_1 \models {}^\bullet A$ to "factor" into proving

$$c_0 \models A_0 \;\&\; c_1 \models A_1 \;\&\; A_0 \times A_1 \models {}^\bullet A.$$

The key to a compositional proof system for $\bullet$ is to obtain such assertions so that we can prove formally the relation between assertions, that $A_0 \times A_1 \vdash {}^\bullet A$. Then we have reduced proving a property holds of a composition $c_0 \bullet c_1$ to proving properties hold of its components c_0 and c_1: If there are such assertions then by the rules $(\times \vdash)$ and (tran), and provided $c_0 \vdash A_0$ and $c_1 \vdash A_1$, we can deduce $c_0 \times c_1 \vdash {}^\bullet A$ and so $c_0 \bullet c_1 \vdash A$. Obtaining suitable assertions A_0 and A_1 is quite involved.

The rules for $\bullet$–preconditions are like those for $\lceil\Lambda$–preconditions and are presented in L.9. There are two introduction rules, one of which makes use of a definedness assertion D, with

$$\begin{aligned} D \equiv \forall x.\, (\lfloor Ex \times \mathrm{tt}\rfloor \wedge \lfloor \mathrm{tt} \times Ex\rfloor) \rightarrow \\ (\lfloor Hx \times \mathrm{tt}\rfloor \leftrightarrow \lfloor \mathrm{tt} \times Hx\rfloor \wedge \\ \lfloor Lx \times \mathrm{tt}\rfloor \leftrightarrow \lfloor \mathrm{tt} \times Lx\rfloor \wedge \\ \textstyle\bigvee_{s \in S} \lfloor Sx = s \times Sx = s\rfloor), \end{aligned}$$

so $(\sigma, \rho) \models D$ iff $\sigma \bullet \rho \downarrow$. This is where the cost of making assertions second-order pays-off. Because we can quantify over subsets (or properties), we can reduce ${}^\bullet A$ to a provably equivalent product assertion which contains no preconditions. The quantification over sets is used to express the fact that the flow relation in a composition $\sigma \bullet \rho$, assumed defined, is the transitive closure of the flow relations contributed by the components σ and ρ. This is the role of axiom $(d^\bullet \rightsquigarrow)$. Suppose $\sigma \bullet \rho$ is defined. Then $(\sigma, \rho) \models {}^\bullet(\alpha \rightsquigarrow \beta)$ means simply that $\sigma \bullet \rho \models \alpha \rightsquigarrow \beta$ for $\alpha, \beta \in \mathrm{sort}(\sigma \bullet \rho)$. Let $\Lambda \subseteq \mathrm{sort}(\sigma \bullet \rho)$. Note we can represent the set Λ as the term Λ which is $\{x : \bigvee_{\lambda \in \Lambda} x = \lambda\}$ in the logic. Say Λ is *closed*, with respect to σ, ρ, if

$$\sigma \bullet \rho \models \Lambda\gamma \;\&\; (\sigma \models \gamma \rightsquigarrow \delta \;\text{ or }\; \rho \models \gamma \rightsquigarrow \delta) \Rightarrow \sigma \bullet \rho \models \Lambda\delta$$

for all $\gamma, \delta \in \mathrm{sort}(\sigma \bullet \rho)$. Now, the flow relation in $\sigma \bullet \rho$ is characterised as being the transitive closure of the flow relations of the components, and this is expressed by the property that

$$\Lambda\alpha \;\&\; \Lambda \text{ is closed } \Rightarrow \Lambda\beta$$

for all $\alpha, \beta \in \mathrm{sort}(\sigma \bullet \rho)$ and for any Λ. This gives the gist of axiom $(d^\bullet \rightsquigarrow)$. Notice it depends on quantifying over subsets. Once we have gone second-order, the distribution axioms for ${}^\bullet(\ )$ enable us to eliminate occurrences of $\bullet$–preconditions from product assertions.

5.6 Theorem.

Let A be a product assertion which may include composition preconditions. There is a product assertion B, which does not contain any preconditions such that $\vdash A \leftrightarrow B$.

Proof. The proof follows the same lines as for restriction preconditions. ∎

But, of course, this was at the expense of making our logic second-order and it is well-known that there is no complete effective axiomatisation of second-order logic. Fortunately we can use the fact that circuits are finite to get around this difficulty. In contexts where it is assumed that all points lie within some finite set we can reduce second-order assertions to provably equivalent first-order or even propositional assertions. (An assertion is *propositional* if it contains no quantifiers or preconditions.)

It is easy to define the sort of circuit term by structural induction:

$$\begin{aligned}
&\mathrm{sort}(\mathit{Pow}\ \alpha) = \mathrm{sort}(\mathit{Gnd}\ \alpha) = \{\alpha\}\\
&\mathrm{sort}(\mathit{res}_g(\alpha,\beta)) = \{\alpha,\beta\}\\
&\mathrm{sort}(\mathit{ntran}(\alpha,\beta,\gamma)) = \mathrm{sort}(\mathit{ptran}(\alpha,\beta,\gamma)) = \{\alpha,\beta,\gamma\}\\
&\mathrm{sort}(c \bullet d) = \mathrm{sort}(c) \cup \mathrm{sort}(d)\\
&\mathrm{sort}(c\lceil\Lambda) = \mathrm{sort}(c) \cap \Lambda.
\end{aligned}$$

In the logic, the fact that a circuit term c has sort Λ, a finite set, is expressed by

$$c \models \bigwedge\!\!\!\bigwedge_{\lambda\in\Lambda} E\lambda \wedge \forall x.\Lambda x$$

where Λx abbreviates $\{y : \bigvee_{\lambda\in\Lambda} y = \lambda\}x$ and so is equivalent to $\bigvee_{\lambda\in\Lambda} x = \lambda$. This can be proved in the formal system, though all we need to show completeness is the following.

5.7 Proposition.

(i) *Let c be a circuit term of sort Λ. Then $c \vdash \forall x.\Lambda x$.*

(ii) *Let c_0, c_1 be circuit terms of sort Λ_0 and Λ_1 respectively. Then $c_0 \times c_1 \vdash \forall x.\ (\Lambda_0 \cup \Lambda_1)x$.*

Proof. The proof of (i) is by structural induction on circuit terms.

For basic components like transistors the sorts are specified in their associated assertions so for a basic component c we have $c \vdash \forall x.\ (\mathrm{sort}(c))x$.

Assume we already know $c \vdash \forall x.Mx$ where M is the sort of c. Then $c \vdash Ex \rightarrow Mx$. Hence $c \vdash \Lambda x \rightarrow \Lambda x \wedge Mx$. But this yields $c \vdash {}^{\Lambda}(Ex \rightarrow (\Lambda\cap M)x)$ from which we derive $c\lceil\Lambda \vdash Ex \rightarrow (\Lambda \cap M)x$. Therefore $c\lceil\Lambda \vdash \forall x.(\Lambda \cap M)x$ and, of course, $\Lambda \cap M$ is the sort of $c\lceil\Lambda$.

Suppose we already know $c_0 \vdash \forall x.\Lambda_0 x$ and $c_1 \vdash \forall x.\Lambda_1 x$ where Λ_0 and Λ_1 are the sorts of c_0 and c_1. Thus

$$c_0 \vdash Ex \rightarrow \Lambda_0 x \text{ and } c_1 \vdash Ex \rightarrow \Lambda_1 x.$$

By the $(\times \vdash)$ rule, $c_0 \times c_1 \vdash |(Ex \to \Lambda_0 x) \times (Ex \to \Lambda_1 x)|$. From the rule $(\times E)$ for product assertions we derive

$$c_0 \times c_1 \vdash Ex \to (|\Lambda_0 x \times \mathbb{t}| \vee |\mathbb{t} \times \Lambda_1 x|)$$

from which

$$c_0 \times c_1 \vdash Ex \to (\Lambda_0 \cup \Lambda_1)x$$

follows. We get

$$c_0 \times c_1 \vdash \forall x.(\Lambda_0 \cup \Lambda_1)x$$

directly, and from the $^\bullet(\ )$ rules we obtain

$$c_0 \times c_1 \vdash {}^\bullet(\forall x.(\Lambda_0 \cup \Lambda_1)x).$$

Now we see $c_0 \bullet c_1 \vdash \forall x.(\Lambda_0 \cup \Lambda_1)x$, as required.

This completes the structural induction, to show (i). It also makes clear how (ii) follows. ∎

In contexts where we know every point belongs to some finite set we can eliminate all quantifiers, both first and second order, to obtain an equivalent propositional assertion (without any quantifiers).

5.8 Proposition. *Let Λ be a finite set of points.*

(ia) $\forall x.\Lambda x \vdash \exists x.A \leftrightarrow \bigvee_{\lambda \in \Lambda}(E\lambda \wedge A[\lambda/x])$.
(ib) $\forall x.\Lambda x \vdash \forall x.A \leftrightarrow \bigwedge_{\lambda \in \Lambda}(E\lambda \to A[\lambda/x])$.
(iia) $\forall x.\Lambda x \vdash \exists P.A \leftrightarrow \bigvee_{M \subseteq \Lambda} A[M/P]$.
(iib) $\forall x.\Lambda x \vdash \forall P.A \leftrightarrow \bigwedge_{M \subseteq \Lambda} A[M/P]$.

Proof.

(ia) Clearly

$$\vdash \bigvee_{\lambda \in \Lambda}(E\lambda \wedge A[\lambda/x]) \to \exists x.A. \tag{1}$$

As $\forall x.\Lambda x \vdash \bigvee_{\lambda \in \Lambda} x = \lambda$,

$$\exists x.A, \forall x.\Lambda x \vdash (Ex \wedge A) \wedge \bigvee_{\lambda \in \Lambda} x = \lambda.$$

Hence

$$\exists x.A, \forall x.\Lambda x \vdash \bigvee_{\lambda \in \Lambda}(Ex \wedge A \wedge x = \lambda).$$

Now, using (eq),

$$\exists x.A, \forall x.\Lambda x \vdash \bigvee_{\lambda \in \Lambda}(E\lambda \wedge A[\lambda/x]).$$

Thus

$$\forall x.\Lambda x \vdash \exists x.A \to \bigvee_{\lambda \in \Lambda}(E\lambda \wedge A[\lambda/x]). \tag{2}$$

Combining (1) and (2) we obtain (ia).

(ib) The proof is similar to that of (ia), but using $\forall$ in place of $\exists$ rules.

(iia) Let $P = Q$ abbreviate $\forall x.Px \leftrightarrow Qx$, where P and Q are second-order variables or terms. By structural induction on A the following substitution property for second-order terms:

$$\vdash P = Q \wedge A[P/R] \rightarrow A[Q/R].$$

For any finite set Λ, we know

$$\vdash \bigwedge_{\lambda\in\Lambda}(P\lambda \wedge \neg P\lambda).$$

By the distribution of $\bigwedge$ over $\vee$ (a metaresult about the proof system),

$$\vdash \bigvee_{M\subseteq\Lambda}([\bigwedge_{\lambda\in M} P\lambda] \wedge [\bigwedge_{\lambda\in\Lambda\setminus M} \neg P\lambda]),$$

or better,

$$\vdash \bigvee_{M\subseteq\Lambda}\bigwedge_{\lambda\in\Lambda}(P\lambda \leftrightarrow M\lambda).$$

Hence by part (i),

$$\forall x.\Lambda x \vdash \bigvee_{M\subseteq\Lambda} P = M.$$

Now (iia) and (iib) follow by proofs on the same lines as in part (i). ∎

It now follows that for a circuit term c and possibly second-order assertion A that

$$c \vdash A \leftrightarrow B$$

where B is purely first-order. The second-order quantifiers may be useful but they are not essential, and can be provably eliminated. Indeed, this is also true for the first order quantifiers, but having already a complete proof system for the underlying first-order free logic there is no need to eliminate them in our earlier work on restriction. However at present obtaining a complete system of proof rules for composition seems to require a stronger elimination for product assertions.

5.9 Corollary. *Let Λ be a finite set of points. Let A be an assertion. There is a propositional assertion p such that*

$$\forall x.\Lambda x \vdash A \leftrightarrow p.$$

Let c be a circuit term, and A an assertion. There is a propositional assertion p such that

$$c \vdash A \leftrightarrow p$$

Proof. By theorem 5.4 we can eliminate preconditions. Then the first part is proved by successively applying the results above. The second part, for circuits, follows by 5.7(i) because entailment is transitive. ∎

It follows from corollary 5.9 that product assertions of interest can be reduced to provably equivalent propositional assertions, in the context of reasoning about circuits. Just as before we can eliminate first-order quantifiers, and second-order quantifiers can be eliminated at least for the product assertions of concern, those of the form $^\bullet A$.

5.10 Proposition. *Let Λ be a finite set of points.*

For product assertions C,

(a) $\forall x.\Lambda x \vdash \exists x.C \leftrightarrow \bigvee_{\lambda\in\Lambda}(E\lambda \wedge C[\lambda/x])$.

(b) $\forall x.\Lambda x \vdash \forall x.C \leftrightarrow \bigwedge_{\lambda\in\Lambda}(E\lambda \rightarrow C[\lambda/x])$.

For an assertion A,

$\forall x.\Lambda x \vdash {}^\bullet A \leftrightarrow p$ *where p is a propositional product assertion .*

Proof.

Parts (a) and (b) follow as earlier.

Arguing in the same way as we did for restriction preconditions, we can derive the rule

$$\frac{\Gamma \vdash A}{{}^\bullet\Gamma \vdash {}^\bullet A},$$

from which we observe it follows that if $\Gamma \vdash A \leftrightarrow B$ then ${}^\bullet\Gamma \vdash {}^\bullet A \leftrightarrow {}^\bullet B$. By corollary 5.9, $\forall x.\Lambda x \vdash A \leftrightarrow r$ where r is a propositional assertion. By the observation, we see

$${}^\bullet\forall x.\Lambda x \vdash {}^\bullet A \leftrightarrow {}^\bullet r.$$

From the distribution rules for ${}^\bullet(\ )$,

$$D, {}^\bullet\forall x.\Lambda x \vdash {}^\bullet A \leftrightarrow q,$$

where q is propositional. Hence

$$\forall x.\Lambda x \vdash {}^\bullet A \leftrightarrow (D \rightarrow q),$$

and now by (a) we can eliminate all the quantifiers in D to obtain the required propositional product assertion p. ∎

5.11 Corollary. *Let c_0 and c_1 be circuit terms. Let A be a second order assertion. Then there is a propositional product assertion p such that*

$$c_0 \times c_1 \vdash {}^\bullet A \leftrightarrow p.$$

Proof. By the transitivity of entailment using 5.7(ii). ∎

Product assertions which are propositional can be put into a useful normal form. The purpose of the distribution rules for product (L.8) is to enable this to be done formally in the proof system.

5.12 Proposition. *Let A be a product assertion which is propositional. Then*

$$\vdash A \leftrightarrow \bigvee_{i\in I} B_i \times C_i$$

where I is a finite set, indexing assertions B_i and C_i.

Proof.

Firstly the logic is classical so $\vdash (A \to B) \leftrightarrow (\neg A \vee B)$ and so we can eliminate occurrences of $\to$ in favour of $\neg$ and $\vee$. Thus without loss of generality we can assume that A is built up solely using $\wedge, \vee, \neg$. The proposition now follows by structural induction on A using basic distributivity properties of the logical connectives. For example, to deal with the hardest case of the induction, we show if we assume the proposition holds for assertion D then it holds for assertion $\neg D$.

Suppose $\vdash D \leftrightarrow \mathbb{W}_{i\in I} B_i \times C_i$. Then

$$\vdash \neg D \leftrightarrow \mathbb{M}_{i\in I} \neg[B_i \times C_i|,$$

and so

$$\vdash \neg D \leftrightarrow \mathbb{M}_{i\in I}([\neg B_i \times \mathrm{tt}| \vee [\mathrm{tt} \times \neg C_i|).$$

But then $\vdash \neg D \leftrightarrow \mathbb{W}W$ where W is the set of product assertions

$$\{[\mathbb{M}_{i\in I} B_i^* \times \mathbb{M}_{i\in I} C_i^* \| \quad (B_i^* \equiv \neg B_i \ \&\ C_i^* \equiv \mathrm{tt}) \text{ or } (B_i^* \equiv \mathrm{tt} \ \&\ C_i^* \equiv \neg C_i), \text{ all } i \in I\},$$

making D provably equivalent to an assertion of the right form. ∎

Now the problem of proving $c_0 \bullet c_1 \models A$ has reduced to proving $c_0 \times c_1 \models \mathbb{W}_{i\in I} B_i \times C_i$. To achieve compositionality we would like to reduce this further, to proving assertions hold of c_0 and c_1. We need to show:

5.13 Lemma. *Let c_0 and c_1 be circuit terms such that*

$$c_0 \times c_1 \models \mathbb{W}_{i\in I} B_i \times C_i$$

for indexed assertions B_i, C_i. Then there are assertions A_0 and A_1 for which

$$c_0 \models A_0 \ \&\ c_1 \models A_1 \ \&\ A_0 \times A_1 \vdash \mathbb{W}_{i\in I} B_i \times C_i.$$

Proof. It is simpler to argue that A_0 and A_1 exist in a nonconstructive way which shows there exists a proof, so $A_0 \times A_1 \vdash \mathbb{W}_{i\in I} B_i \times C_i$, without giving it explicitly. Of course, this is all that is needed to show completeness. We know $c_0 \times c_1 \models \mathbb{W}_{i\in I} B_i \times C_i$. Hence there is a function $i[\ ,\ |$ so that for any $\sigma \models c_0$ and $\rho \models c_1$ there is $i[\sigma, \rho| \in I$ such that

$$\sigma \models B_{i[\sigma,\rho|} \ \&\ \rho \models C_{i[\sigma,\rho|}.$$

(Note this does not determine $i[\ ,\ |$ uniquely).
Now

$$\sigma \models \mathbb{M}_{\rho \models c_1} B_{i[\sigma,\rho|} \ \&\ \rho \models \mathbb{M}_{\sigma \models c_0} C_{i[\sigma,\rho|}$$

for any $\sigma \models c_0$ and $\rho \models c_1$. We use *e.g.* $\mathbb{M}_{\rho \models c_1} B_{i[\sigma,\rho|}$ to mean the finite conjunction

$$\mathbb{M}\{B_{i[\sigma,\rho|} \mid \rho \models c_1\}.$$

Clearly

$$\bigwedge_{\rho \models c_1} B_{i[\sigma,\rho]} \vdash B_{i[\sigma,\rho]} \quad \text{for } \sigma \models c_0 \text{ and } \rho \models c_1, \text{ and}$$
$$\bigwedge_{\sigma \models c_0} C_{i[\sigma,\rho]} \vdash C_{i[\sigma,\rho]} \quad \text{for } \sigma \models c_0 \text{ and } \rho \models c_1,$$

as *e.g.* the conjunction $\bigwedge_{\rho \models c_1} B_{i[\sigma,\rho]}$ contains $B_{i[\sigma,\rho]}$ as a conjunct. Thus

$$\bigwedge_{\rho \models c_1} B_{i[\sigma,\rho]} \times \bigwedge_{\sigma \models c_0} C_{i[\sigma,\rho]} \vdash B_{i[\sigma,\rho]} \times C_{i[\sigma,\rho]}$$

for any $\sigma \models c_0$ and $\rho \models c_1$. Write

$$A_0 \equiv \bigvee_{\sigma \models c_0} \bigwedge_{\rho \models c_1} B_{i[\sigma,\rho]} \quad \text{and} \quad A_1 \equiv \bigvee_{\rho \models c_1} \bigwedge_{\sigma \models c_0} C_{i[\sigma,\rho]}.$$

By $(\times \vee)$,

$$A_0 \times A_1 \vdash \bigvee_{(\sigma,\rho) \models c_0 \times c_1} [\bigwedge_{\rho \models c_1} B_{i[\sigma,\rho]} \times \bigwedge_{\sigma \models c_0} C_{i[\sigma,\rho]}].$$

Also, obviously,

$$\bigvee_{(\sigma,\rho) \models c_0 \times c_1} [\bigwedge_{\rho \models c_1} B_{i[\sigma,\rho]} \times \bigwedge_{\sigma \models c_0} C_{i[\sigma,\rho]}] \vdash \bigvee_{i \in I} B_i \times C_i.$$

Therefore $A_0 \times A_1 \vdash \bigvee_{i \in I} B_i \times C_i$. Clearly $c_0 \models A_0$ and $c_1 \models A_1$. ∎

Now we can tidy up all the loose ends and prove soundness and completeness.

5.14 Theorem. *The proof system is sound i.e.*

$$\Gamma \vdash A \Rightarrow \Gamma \models A$$

for a finite set Γ of circuit-assertions and circuit-assertion A. It is complete in the restricted sense that

$$c \models A \Rightarrow c \vdash A$$

for a circuit term c and assertion A.

Proof. We omit the proof of soundness which is routine if tedious. The proof of completeness follows by structural induction on c.

Suppose the case where c is an atomic circuit term, so suppose for example $c \equiv ntran(\alpha,\beta,\gamma)$. The rule (*ntran* E) has the form $ntran(\alpha,\beta,\gamma) \vdash NT$, where NT is a first order assertion describing the behaviour of $ntran(\alpha,\beta,\gamma)$. In fact $[\![ntran(\alpha,\beta,\gamma)]\!] = [\![NT]\!]$. Suppose $ntran(\alpha,\beta,\gamma) \models A$ for an assertion A. Then, as we have shown, $ntran(\alpha,\beta,\gamma) \vdash A \leftrightarrow B$ where B is a first-order assertion. Thus $NT \models B$, so by the completeness theorem 5.1, $NT \vdash B$. Hence $ntran(\alpha,\beta,\gamma) \vdash A$. The other atomic cases are similar.

In the case of restriction suppose $c\lceil\Lambda \models A$ and inductively assume that $c \models B$ iff $c \vdash B$ for any assertion B. Then $c \models {}^{\Lambda}A$. By the metatheorem 5.4, we can eliminate all preconditions to obtain

$$\vdash {}^{\Lambda}A \leftrightarrow B$$

for some assertion B. But $c \models B$ and by the inductive assumption $c \vdash B$. This gives $c \vdash {}^{\wedge}A$ so $c \lceil \Lambda \vdash A$, as required in this case.

In the case of composition, suppose $c_0 \bullet c_1 \models A$ and inductively assume $c_0 \models A_0$ iff $c_0 \vdash A_0$ and $c_1 \models A_1$ iff $c_1 \vdash A_1$ for any assertions A_0 and A_1. Then $c_0 \times c_1 \models {}^{\bullet}A$. Our earlier results show that

$$c_0 \times c_1 \vdash {}^{\bullet}A \leftrightarrow \mathbb{W}_{i \in I} B_i \times C_i,$$

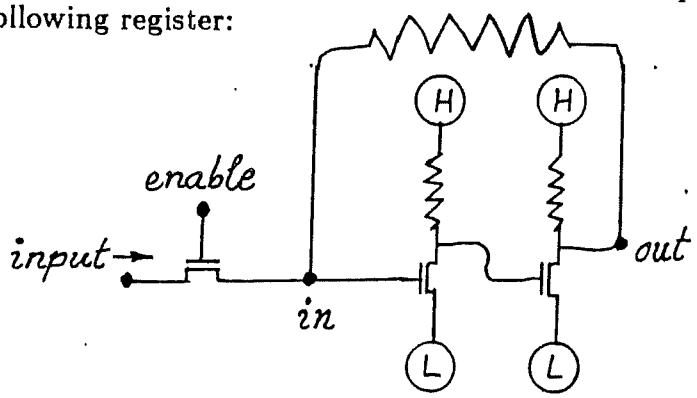
—and, the above lemma, that in such a case there are A_0 and A_1 with

$$c_0 \models A_0 \ \& \ c_1 \models A_1 \ \& \ A_0 \times A_1 \vdash \mathbb{W}_{i \in I} B_i \times C_i,$$

precisely what is needed to conclude $c_0 \times c_1 \vdash {}^{\bullet}A$ and hence $c_0 \bullet c_1 \vdash A$. ∎

6. Conclusion.

We have presented a compositional model for the behaviour of MOS circuits in a static environment. Certainly such a model is necessary for a satisfactory treatment of dynamically changing circuits. Their behaviour can often be viewed as going through a sequence of static configurations as data changes in synchrony with one or several clocks. The data and the clock pulses are held long enough for the circuit to settle into a static configuration. It seems natural therefore to represent a possible history for such a circuit as a sequence of static configurations. Copying the approach in [Gor2,3] we could try to extend our work to the dynamic case by simply including a time parameter $t \in \omega$, and associating devices with assertions expressing time dependencies between static configurations at different times. For example, then we could express facts like a capacitance would contribute a charge at time $t+1$ if it was precharged at time t. Although it is not hard to push through this programme, it is disappointing that there are difficulties in making the model accurate. A main problem it seems is that short-term capacitance effects influence the physical behaviour but are not captured easily in any extension of our model. The strength orders we use assume that the environment is stable long enough for sources of current to override charges stored by capacitance. This assumption breaks down in some stages needed to explain the behaviour of devices like the following register:



When *enable* is high, a strong signal at *in* overrides that already present, and its value is established, after two inversions, at *out*. When *enable* becomes low the value is preserved at *out* (the input value is stored). Speaking loosely, the latter stage relies on capacitance to maintain the value at *in* until the feedback loop takes over. This occurs over a very short time when the assumptions behind the strength order are violated. I cannot, at present, see how to extend the model to account for effects over such a short time. Through failing to cope adequately with such short-term effects, the models I have developed allow more possibilities than are physically possible. By the way, the work of Bryant does not address this problem; a simulator need only generate one possible course of behaviour and Bryant does this by making a unit-delay assumption. (In his simulators all transistors switch with the same delay after their gates change.)

It is unfortunate that the logic is so complicated even for static circuits. This may be in the nature of things. If so it makes even more pressing the task of relating models of the kind here to the models and logic like [Gor2,3] which are relatively simple to work with. There must be "correspondence principles" which express how and when one model reduces to another. In general relations between models may be quite complex. A physical model implements a discrete model only provided certain conditions are met.

So far there have been several successful attempts at relating the physical and the logical levels for specific pieces of hardware, *e.g.* [GH, HD], and carrying through the thesis proposal [Me] should lay bare some of the key issues.

Acknowledgements

I am grateful to many people for help at various times. I owe a debt to the thesis work of Luca Cardelli [Car]; there Luca attempted to give a semantics to circuits on roughly similar lines to section 4 though his model was not sufficiently detailed to support a satisfactory treatment of hiding. His approach and mine were guided by work on CCS and CSP. Mike Gordon's course on hardware verification exposed the problems which led to this work. I would like to thank him and members of the hardware verification group here for help and encouragement on many occasions in an area where I'm still finding my feet. Special thanks are due to Inder Dhingra, Edmund Ronald and Edmund Robinson, all of whom contributed ideas. The work [A] of Samson Abramsky provided a vital hint on how to make the proof system, and the work in [S] and [P] gave useful leads. Many thanks to Mogens Nielsen for his patient reading of this and his constructive criticism. Thanks too to Randy Bryant for encouragement.

Appendix. The logic of static circuits .

L.1 The assertion language for circuits.

Assume a particular strength order S (with typical members $s, s', \cdots$), a countably infinite set of point names Π (with typical members $\alpha, \beta, \gamma, \cdots$) and Var a set of variables. ranging over points (with typical members $x, y, z, \cdots$) and Pvar a set of second order variables ranging over sets (or properties) of points (with typical members $P, Q, R, \cdots$).

Define a *strength expression.* by induction, to have the form

$$s \mid S\pi \mid e_0 \cdot e_1 \mid e_0 + e_1$$

where $s \in \mathbf{S}$, $\pi \in \Pi \cup \mathrm{Var}$ and e_0, e_1 are strength expressions.

Define a *point relation* to have the form

$$H\pi \mid L\pi \mid h\pi \mid l\pi \mid \pi_0 = \pi_1 \mid \pi_0 \nrightarrow \pi_1 \mid E\pi$$

where $\pi, \pi_0, \pi_1 \in \Pi \cup \mathrm{Var}$.

Define a *strength relation* to have the form

$$Ee \mid e_0 \leq e_1 \mid e_0 = e_1$$

where e, e_0 and e_1 are strength expressions.

Define a *circuit-assertion*, by induction, to have the form

$$\begin{array}{l} \mathit{circ} \mid \mathit{srel} \mid \mathit{prel} \mid \\ \mathbb{tt} \mid \mathbb{ff} \mid A \wedge B \mid A \vee B \mid A \rightarrow B \mid \exists x.A \mid \forall x.A \mid \\ \{x : A\}\pi \mid P\pi \mid \exists P.A \mid \forall P.A \mid \\ {}^{\Lambda}A \end{array}$$

where *srel* is a strength relation, *prel* is a point relation, *circ* is a circuit term, and Λ is a finite subset of ports with $x \in \mathrm{Var}$ and $P \in \mathrm{Pvar}$.

L.2 Notation.

We shall take the treatment of free and bound variables, and open and closed strength expressions and assertions as understood. Write $\mathrm{FV}(A)$ for the set of free variables (first and second order) of an assertion A. We write $A[\pi/x]$ for the result of substituting π for all free occurrences of the variable x in A, and similarly $A[T/P]$ for the result of substituting a term denoting a property for the second order variable P—we assume changes are made in the naming of bound variables to avoid the binding of free variables in the substituted terms.
Take $\neg A$ to abbreviate the assertion $A \rightarrow \mathbb{ff}$, for an assertion A. Take $A \leftrightarrow B$ to abbreviate $A \rightarrow B \wedge B \rightarrow A$.

Let $\Gamma = \{A_0, \cdots, A_n\}$ be a finite set of assertions. We use $\bigwedge\Gamma$ to abbreviate their conjunction $A_0 \wedge \cdots \wedge A_n$ and $\bigvee\Gamma$ to abbreviate their disjunction $A_0 \vee \cdots \vee A_n$. We identify $\bigwedge\emptyset$ with $\mathbb{t}$ and $\bigvee\emptyset$ with $\mathbb{f}$.
We use $\forall x : Q.\ A$ to abbreviate $\forall x.\ Qx \rightarrow A$ and $\exists x : Q.\ A$ to abbreviate $\exists x.\ Qx \wedge A$.
If Λ is a finite set we use Λ in the logic to abbreviate the term $\{x : \bigvee_{\lambda\in\Lambda} x = \lambda\}$.

L.3 Semantics of the assertion language

We take the set of points to be Π, and static configurations to have sorts a subset of these. A point name α is thus associated with a corresponding point.

Semantics of strength expressions

Define $[\![\ ]\!]$ from closed strength expressions to subsets of pairs of static configurations and strengths by the following induction, with the understanding that static configurations σ have the form $\sigma = \langle S, V, I, \nrightarrow\rangle$.

$$
\begin{aligned}
&[\![s]\!] = \{(\sigma, s) \mid \sigma \in \mathrm{Sta}\}\\
&[\![S\alpha]\!] = \{(\sigma, s) \mid \alpha \in \mathrm{sort}(\sigma)\ \ \&\ \ S\alpha = s\}\\
&[\![e_0 \cdot e_1]\!] = \{(\sigma, s_0 \cdot s_1) \mid (\sigma, s_0) \in [\![e_0]\!]\ \ \&\ \ (\sigma, s_1) \in [\![e_1]\!]\}\\
&[\![e_0 + e_1]\!] = \{(\sigma, s_0 + s_1) \mid (\sigma, s_0) \in [\![e_0]\!]\ \ \&\ \ (\sigma, s_1) \in [\![e_1]\!]\}
\end{aligned}
$$

Semantics of circuit assertions

We firstly define the semantics of point relations, with the understanding that static configurations σ have the form $\sigma = \langle S, V, I, \nrightarrow\rangle$. Define $[\![\ ]\!]$ from closed point relations to the subsets of static configurations which satisfy them by

$$
\begin{aligned}
&[\![H\alpha]\!] = \{\sigma \mid \alpha \in \mathrm{sort}(\sigma)\ \ \&\ \ H \leq V\alpha\}\\
&[\![L\alpha]\!] = \{\sigma\ \ \alpha \in \mathrm{sort}(\sigma)\ \ \&\ \ L \leq V\alpha\}\\
&[\![h\alpha]\!] = \{\sigma \mid \alpha \in \mathrm{sort}(\sigma)\ \ \&\ \ H \leq I\alpha\}\\
&[\![l\alpha]\!] = \{\sigma \mid \alpha \in \mathrm{sort}(\sigma)\ \ \&\ \ L \leq I\alpha\}\\
&[\![\alpha = \beta]\!] = \{\sigma \mid \alpha = \beta \in \mathrm{sort}(\sigma)\}\\
&[\![\alpha \nrightarrow \beta]\!] = \{\sigma \mid \alpha, \beta \in \mathrm{sort}(\sigma)\ \ \&\ \ \alpha \nrightarrow \beta\}\\
&[\![E\alpha]\!] = \{\sigma \mid \alpha \in \mathrm{sort}(\sigma)\}.
\end{aligned}
$$

Now we define the semantics of strength relations:

$$
\begin{aligned}
&[\![Ee]\!] = \{\sigma \mid \exists s.\ (\sigma, s) \in [\![e]\!]\}\\
&[\![e_0 \leq e_1]\!] = \{\sigma \mid \exists s_0, s_1.\ (\sigma, s_0) \in [\![e_0]\!]\ \ \&\ \ (\sigma, s_1) \in [\![e_1]\!]\ \ \&\ \ s_0 \leq s_1\}\\
&[\![e_0 = e_1]\!] = \{\sigma \mid \exists s_0, s_1.\ (\sigma, s_0) \in [\![e_0]\!]\ \ \&\ \ (\sigma, s_1) \in [\![e_1]\!]\ \ \&\ \ s_0 = s_1\}
\end{aligned}
$$

Now we have defined the semantics of point and strength relations, we extend the semantics to all closed assertions by the following induction on the length of assertions:

$$
\begin{aligned}
&[\![\mathrm{tt}]\!] = \mathrm{Sta}\\
&[\![\mathrm{ff}]\!] = \emptyset\\
&[\![A \wedge B]\!] = [\![A]\!] \cap [\![B]\!]\\
&[\![A \vee B]\!] = [\![A]\!] \cup [\![B]\!]\\
&[\![A \to B]\!] = (\mathrm{Sta} \setminus [\![A]\!]) \cup [\![B]\!]\\
&[\![\exists x.A]\!] = \{\sigma \mid \exists \alpha \in \mathrm{sort}(\sigma).\ \sigma \in [\![A[\alpha/x]]\!]\}\\
&[\![\forall x.A]\!] = \{\sigma \mid \forall \alpha \in \mathrm{sort}(\sigma).\ \sigma \in [\![A[\alpha/x]]\!]\}\\
&[\![\{x : A\}\alpha]\!] = [\![E\alpha \ \&\ A[\alpha/x]]\!]\\
&[\![\exists P.A]\!] = \{\sigma \mid \exists \Lambda \subseteq \mathrm{sort}(\sigma).\ \sigma \in [\![A[\Lambda/P]]\!]\}\\
&[\![\forall P.A]\!] = \{\sigma \mid \forall \Lambda \subseteq \mathrm{sort}(\sigma).\ \sigma \in [\![A[\Lambda/P]]\!]\},\\
&[\![\ ^{\wedge}A]\!] = \{\sigma \mid \sigma\lceil \Lambda \downarrow \Rightarrow \sigma\lceil \Lambda \in [\![A]\!]\}.
\end{aligned}
$$

We have already seen how to define the denotations of circuit terms in section 4.

L.4 Satisfaction, validity and entailment

For a closed circuit assertion A we write

$$\sigma \models A \text{ iff } \sigma \in [\![A]\!],$$

and say σ *satisfies* A.

Let A be an assertion with free variables $x_0, \ldots, P_0, \ldots$. Define

$$\models A \quad \text{iff} \quad \sigma \models A[\alpha_0/x_0, \ldots, \Lambda_0/P_0, \ldots] \text{ for all } \sigma \in \mathrm{Sta}$$

for any substitution with $\alpha_0, \ldots \in \Pi$ and $\Lambda_0, \ldots \subseteq \Pi$. When $\models A$ we say A is *valid*.

More generally, letting Γ be a set of assertions and A an assertion, define $\Gamma \models A$ iff for every substitution ϑ of the free variables in Γ and A every static configuration which satisfies each assertion $B[\vartheta]$, for B in Γ, also satisfies $A[\vartheta]$.

L.5 Proof rules for circuit assertions

The proof rules follow a natural deduction style [Pr], with extra axioms special to static configurations and circuits.

Structural rules

(refl) $\Gamma \vdash A$ if $A \in \Gamma$ (tran) $\dfrac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B}$

Propositional logic

($\wedge$I) $\dfrac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$ ($\wedge$E) $\dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \quad \dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}$

($\vee$I) $\dfrac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \quad \dfrac{\Gamma \vdash B}{\Gamma \vdash A \vee B}$ ($\vee$E) $\dfrac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}$

($\rightarrow$I) $\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}$ ($\rightarrow$E) $\dfrac{\Gamma \vdash A \rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}$

(tt) $\Gamma \vdash \mathrm{tt}$ (ff) $\dfrac{\Gamma, \neg A \vdash \mathrm{ff}}{\Gamma \vdash A}$

First-order rules

(sub) $\dfrac{\Gamma \vdash A}{\Gamma \vdash A[\pi/x]}$ $(x \notin \mathrm{FV}(\Gamma))$

($\forall$I) $\dfrac{\Gamma \vdash Ex \rightarrow A}{\Gamma \vdash \forall x.\ A}$ $(x \notin \mathrm{FV}(\Gamma))$ ($\forall$E) $\dfrac{\Gamma \vdash \forall x.\ A}{\Gamma \vdash Et \rightarrow A[t/x]}$

($\exists$I) $\dfrac{\Gamma \vdash Et \wedge A[t/x]}{\Gamma \vdash \exists x.\ A}$ ($\exists$E) $\dfrac{\Gamma \vdash \exists x.\ A \quad \Gamma, Ex \wedge A \vdash B}{\Gamma \vdash B}$ $(x \notin \mathrm{FV}(\Gamma, B))$

Rules for second-order quantifiers and abstraction

(sub^2) $\dfrac{\Gamma \vdash A}{\Gamma \vdash A[T/P]}$ $(P \notin \mathrm{FV}(\Gamma))$

($\forall^2$I) $\dfrac{\Gamma \vdash A}{\Gamma \vdash \forall P.\ A}$ $(P \notin \mathrm{FV}(\Gamma))$ ($\forall^2$E) $\dfrac{\Gamma \vdash \forall P.\ A}{\Gamma \vdash A[T/P]}$

($\exists^2$I) $\dfrac{\Gamma \vdash A[T/P]}{\Gamma \vdash \exists P.\ A}$ ($\exists^2$E) $\dfrac{\Gamma \vdash \exists P.\ A \quad \Gamma, A \vdash B}{\Gamma \vdash B}$ $(P \notin \mathrm{FV}(\Gamma, B))$

(ab I) $\dfrac{\Gamma \vdash Ey \wedge A[y/x]}{\Gamma \vdash \{x : A\}y}$ (ab E) $\dfrac{\Gamma \vdash \{x : A\}y}{\Gamma \vdash Ey \wedge A[y/x]}$

(strict) $\Gamma \vdash Px \rightarrow Ex$

Remark. A word on the second-order logic: The idea is that in a particular static configuration σ, $\{x : A\}$ denotes the set of points in sort(σ) which satisfy A. So unlike first-order terms for points or strengths, set abstractions are always defined, and the second-order rules do not need to invoke an existence predicate. Although set abstractions only appear within assertions in the form $\{x : A\}\pi$, their use enable a simpler account of the rules for second-order quantifiers—*e.g.* try writing the $(\exists^2 I)$ rule without!

Proof rules for point relations

We assume that point names are in 1–1 correspondence with points, so we have the following basic axioms.

$\vdash \alpha = \beta$ iff $\models \alpha = \beta$ (*i.e.* α and β are the same point name).

$\vdash \neg\alpha = \beta$ iff $\models \neg\alpha = \beta$ (*i.e.* α and β are different point names).

The following axioms are essentially a reformulation of the axioms for static configurations which we have seen earlier.

(refl) $\quad \vdash Ex \leftrightarrow x = x$

(eq) $\quad \vdash x = y \wedge A[x/z] \rightarrow A[y/z]$

(str)
$$\vdash Hx \rightarrow Ex$$
$$\vdash Lx \rightarrow Ex$$
$$\vdash hx \rightarrow Ex$$
$$\vdash lx \rightarrow Ex$$
$$\vdash x \nrightarrow y \rightarrow Ex \wedge Ey$$

$$\vdash Ex \rightarrow x \nrightarrow x$$
$$\vdash x \nrightarrow y \wedge y \nrightarrow z \rightarrow x \nrightarrow z$$
$$\vdash x \nrightarrow y \rightarrow Sx \geq Sy$$
$$\vdash x \nrightarrow y \wedge Sx = Sy \rightarrow y \nrightarrow x$$
$$\vdash x \nrightarrow y \wedge \mathbb{W}_{k\in K} Sy = k \rightarrow Sx = Sy$$
$$\vdash Sx = 0 \leftrightarrow \neg Hx \wedge \neg Lx$$
$$\vdash hx \rightarrow Hx \wedge lx \rightarrow Lx$$
$$\vdash Hx \wedge x \nrightarrow y \rightarrow Hy$$
$$\vdash Lx \wedge x \nrightarrow y \rightarrow Ly$$
$$\vdash hx \wedge x \nrightarrow y \rightarrow hy$$
$$\vdash lx \wedge x \nrightarrow y \rightarrow ly$$

Proof rules for strength relations

$\vdash s \leq s'$ if $\models s \leq s'$ $\qquad$ $\vdash \neg s \leq s'$ if $\models \neg s \leq s'$

$\vdash s = s'$ if $\models s = s'$ $\qquad$ $\vdash \neg s = s'$ if $\models \neg s = s'$

$\vdash s_0 \cdot s_1 = s$ if $\models s_0 \cdot s_1 = s$ $\qquad$ $\vdash \neg s_0 \cdot s_1 = s$ if $\models \neg s_0 \cdot s_1 = s$

$\vdash s_0 + s_1 = s$ if $\models s_0 + s_1 = s$ $\qquad$ $\vdash \neg s_0 + s_1 = s$ if $\models \neg s_0 + s_1 = s$

$$\vdash Ee \leftrightarrow e = e$$
$$\vdash e = e' \rightarrow e' = e$$

$$\vdash e = e' \wedge e' = e'' \rightarrow e = e''$$

$$\vdash Ex \rightarrow \bigvee_{s \in S} Sx = s$$
$$\vdash E(Sx) \rightarrow Ex$$
$$\vdash E(e_0 \cdot e_1) \leftrightarrow Ee_0 \wedge Ee_1$$
$$\vdash E(e_0 + e_1) \leftrightarrow Ee_0 \wedge Ee_1$$
$$\vdash e_0 \leq e_1 \rightarrow Ee_0 \wedge Ee_1$$

$$\vdash e_0 = e_0' \wedge e_1 = e_1' \rightarrow e_0 \cdot e_1 = e_0' \cdot e_1'$$
$$\vdash e_0 = e_0' \wedge e_1 = e_1' \rightarrow e_0 + e_1 = e_0' + e_1'$$
$$\vdash e_0 = e_0' \wedge e_1 = e_1' \wedge e_0 \leq e_1 \rightarrow e_0' \leq e_1'$$

L.6 Rules for restriction preconditions

Let

$$G \equiv \forall x.\ \neg \Lambda x \rightarrow [\mathrm{H}x \rightarrow hx \vee (\exists y : \Lambda.\ \mathrm{H}y \wedge y \rightarrowtail x)] \wedge [\mathrm{L}x \rightarrow lx \vee (\exists y : \Lambda.\ \mathrm{L}y \wedge y \rightarrowtail x)]$$

Introduction rules for $^{\Lambda}(\ )$:

$$\frac{\vdash A}{\vdash {}^{\Lambda}A} \qquad \neg G \vdash {}^{\Lambda}A$$

Distribution rules for $^{\Lambda}(\ )$:

$$G \vdash {}^{\Lambda}(Sx = s) \leftrightarrow (Sx = s \wedge \Lambda x)$$
$$G \vdash {}^{\Lambda}(Hx) \leftrightarrow (Hx \wedge \Lambda x)$$
$$G \vdash {}^{\Lambda}(hx) \leftrightarrow (hx \wedge \Lambda x)$$
$$G \vdash {}^{\Lambda}(x \rightarrowtail y) \leftrightarrow (x \rightarrowtail y \wedge \Lambda x \wedge \Lambda y)$$
$$G \vdash {}^{\Lambda}(Ex) \leftrightarrow \Lambda x$$
$$G \vdash {}^{\Lambda}\mathrm{tt} \leftrightarrow \mathrm{tt}$$
$$G \vdash {}^{\Lambda}(A \wedge B) \leftrightarrow ({}^{\Lambda}A \wedge {}^{\Lambda}B)$$
$$G \vdash {}^{\Lambda}(A \rightarrow B) \leftrightarrow ({}^{\Lambda}A \rightarrow {}^{\Lambda}B)$$
$$G \vdash {}^{\Lambda}(\forall x.A) \leftrightarrow (\forall x : \Lambda.\ {}^{\Lambda}A)$$
$$G \vdash {}^{\Lambda}(\{x : A\}y) \leftrightarrow (\{x : {}^{\Lambda}A\}y \wedge \Lambda y)$$
$$G \vdash {}^{\Lambda}(\forall P.A) \leftrightarrow \forall P.\ {}^{\Lambda}A$$

$$G \vdash {}^{\Lambda}(Lx) \leftrightarrow (Lx \wedge \Lambda x)$$
$$G \vdash {}^{\Lambda}(lx) \leftrightarrow (lx \wedge \Lambda x)$$
$$G \vdash {}^{\Lambda}(x = y) \leftrightarrow (x = y \wedge \Lambda x \wedge \Lambda y)$$
$$G \vdash {}^{\Lambda}\mathrm{ff} \leftrightarrow \mathrm{ff}$$
$$G \vdash {}^{\Lambda}(A \vee B) \leftrightarrow ({}^{\Lambda}A \vee {}^{\Lambda}B)$$
$$G \vdash {}^{\Lambda}(\exists x.A) \leftrightarrow (\exists x : \Lambda.\ {}^{\Lambda}A)$$
$$G \vdash {}^{\Lambda}(Px) \leftrightarrow (Px \wedge \Lambda x)$$
$$G \vdash {}^{\Lambda}(\exists P.A) \leftrightarrow \exists P.\ {}^{\Lambda}A$$

L.7 Product circuit–assertions

Product circuit–assertions are defined inductively to have the form

$$A \times B \mid {}^{\bullet}A \mid E\pi \mid \pi_0 = \pi_1$$
$$\mathtt{tt} \mid \mathtt{ff} \mid D_0 \wedge D_1 \mid D_0 \vee D_1 \mid D_0 \to D_1 \mid \exists x.D \mid \forall x.D \mid$$
$$\{x : D\}\pi \mid \exists P.D \mid \forall P.D$$

where A, B are circuit–assertions and D, D_0, D_1 are product assertions.

L.8 Semantics of product assertions:

$$[\![A \times B]\!] = [\![A]\!] \times [\![B]\!]$$
$$[\![{}^{\bullet}A]\!] = \{(\sigma,\rho) \mid \sigma \bullet \rho \downarrow \Rightarrow \sigma \bullet \rho \in [\![A]\!]\}$$
$$[\![E\alpha]\!] = \{(\sigma,\rho) \mid \alpha \in \mathrm{sort}(\sigma) \cup \mathrm{sort}(\rho)\}$$
$$[\![\alpha = \beta]\!] = \{(\sigma,\rho) \mid \alpha = \beta \in \mathrm{sort}(\sigma) \cup \mathrm{sort}(\rho)\}$$
$$[\![\{x : D\}\alpha]\!] = [\![E\alpha \wedge D[\alpha/x]]\!]$$
$$[\![\exists P.D]\!] = \{(\sigma,\rho) \mid \exists \Lambda \subseteq \mathrm{sort}(\sigma) \cup \mathrm{sort}(\rho).\ (\sigma,\rho) \in [\![D[\Lambda/P]]\!]\}$$
$$[\![\forall P.D]\!] = \{(\sigma,\rho) \mid \forall \Lambda \subseteq \mathrm{sort}(\sigma) \cup \rho.\ (\sigma,\rho) \in [\![D[\Lambda/P]]\!]\}$$

The semantics for the remaining clauses follow those for circuit assertions. The proof rules for product assertions include those for for second order logic, with the understanding that terms T substituted in the second order quantifier rules are first–order set abstractions, *i.e.* of the form $\{x : A\}$ with A first–order. We include, in addition, the following.

Proof rules for product assertions:

$(\times \vdash)$ $\quad \dfrac{A \vdash A' \quad B \vdash B'}{A \times B \vdash A' \times B'}$

$(\times E)$ $\quad \vdash Ex \leftrightarrow ([Ex \times \mathtt{tt}] \vee [\mathtt{tt} \times Ex])$

$(\times =)$ $\quad \vdash x = y \leftrightarrow ([x = y \times \mathtt{tt}] \vee [\mathtt{tt} \times x = y])$

(eq) $\quad \vdash x = y \wedge C[x/z] \to C[y/z]$

$(\times \mathtt{tt})$ $\quad \vdash \mathtt{tt} \leftrightarrow [\mathtt{tt} \times \mathtt{tt}]$

$(\times \mathtt{ff})$ $\quad \vdash \mathtt{ff} \leftrightarrow ([\mathtt{ff} \times \mathtt{tt}] \vee [\mathtt{tt} \times \mathtt{ff}])$

$(\times \wedge)$ $\quad \vdash ([A \times B] \wedge [A' \times B']) \leftrightarrow ([A \wedge A'] \times [B \wedge B'])$

$(\times \vee)$ $\quad \vdash [(A \vee A') \times B] \leftrightarrow ([A \times B] \vee [A' \times B])$

$\quad \vdash [A \times (B \vee B')] \leftrightarrow ([A \times B] \vee [A \times B'])$

$(\times \neg)$ $\quad \vdash \neg[A \times B] \leftrightarrow ([\neg A \times \mathtt{tt}] \vee [\mathtt{tt} \times \neg B])$

L.9. Rules for composition precondition

Let

$$D \equiv \forall x.\ (\lfloor Ex \times \mathrm{tt} \rfloor \wedge \lfloor \mathrm{tt} \times Ex \rfloor) \rightarrow (\lfloor Hx \times \mathrm{tt} \rfloor \leftrightarrow \lfloor \mathrm{tt} \times Hx \rfloor \wedge \lfloor Lx \times \mathrm{tt} \rfloor \leftrightarrow \lfloor \mathrm{tt} \times Lx \rfloor \wedge \mathbb{W}_{s \in S} \lfloor Sx = s \times Sx = s \rfloor)$$

Introduction rules for $^\bullet(\ )$:

$$(\ ^\bullet I) \qquad \frac{\vdash A}{\vdash {}^\bullet A} \qquad \frac{}{\neg D \vdash {}^\bullet A}$$

Distribution rules for $^\bullet(\ )$:

$(d\ ^\bullet S)$ $D \vdash {}^\bullet(Sx = s) \leftrightarrow (\lfloor Sx = s \times \mathrm{tt} \rfloor \vee \lfloor \mathrm{tt} \times Sx = s \rfloor)$

$(d\ ^\bullet H)$ $D \vdash {}^\bullet(Hx) \leftrightarrow (\lfloor Hx \times \mathrm{tt} \rfloor \vee \lfloor \mathrm{tt} \times Hx \rfloor)$

$(d\ ^\bullet L)$ $D \vdash {}^\bullet(Lx) \leftrightarrow (\lfloor Lx \times \mathrm{tt} \rfloor \vee \lfloor \mathrm{tt} \times Lx \rfloor)$

$(d\ ^\bullet h)$ $D \vdash {}^\bullet(hx) \leftrightarrow (\exists y.\ (\lfloor hy \times \mathrm{tt} \rfloor \vee \lfloor \mathrm{tt} \times hy \rfloor) \wedge {}^\bullet y \nrightarrow x)$

$(d\ ^\bullet l)$ $D \vdash {}^\bullet(lx) \leftrightarrow (\exists y.\ (\lfloor ly \times \mathrm{tt} \rfloor \vee \lfloor \mathrm{tt} \times ly \rfloor) \wedge {}^\bullet y \nrightarrow x)$

$(d\ ^\bullet \nrightarrow)$ $D \vdash {}^\bullet(x \nrightarrow y) \leftrightarrow \lfloor \forall P.\ {}^\bullet Px \wedge (\forall v, w.\ {}^\bullet Pv \wedge (\lfloor v \nrightarrow w \times \mathrm{tt} \rfloor \vee \lfloor \mathrm{tt} \times v \nrightarrow w \rfloor) \rightarrow {}^\bullet Pw) \rightarrow {}^\bullet Py \rfloor$

$(d\ ^\bullet =)$ $D \vdash {}^\bullet(x = y) \leftrightarrow (\lfloor x = y \times \mathrm{tt} \rfloor \vee \lfloor \mathrm{tt} \times x = y \rfloor)$

$(d\ ^\bullet E)$ $D \vdash {}^\bullet(Ex) \leftrightarrow Ex$

$(d\ ^\bullet \mathrm{tt})$ $D \vdash {}^\bullet \mathrm{tt} \leftrightarrow \mathrm{tt}$

$(d\ ^\bullet \mathrm{ff})$ $D \vdash {}^\bullet \mathrm{ff} \leftrightarrow \mathrm{ff}$

$(d\ ^\bullet \wedge)$ $D \vdash {}^\bullet(A \wedge B) \leftrightarrow ({}^\bullet A \wedge {}^\bullet B)$

$(d\ ^\bullet \vee)$ $D \vdash {}^\bullet(A \vee B) \leftrightarrow ({}^\bullet A \vee {}^\bullet B)$

$(d\ ^\bullet \rightarrow)$ $D \vdash {}^\bullet(A \rightarrow B) \leftrightarrow ({}^\bullet A \rightarrow {}^\bullet B)$

$(d\ ^\bullet \forall)$ $D \vdash {}^\bullet(\forall x.A) \leftrightarrow \forall x.\ {}^\bullet A$

$(d\ ^\bullet \exists)$ $D \vdash {}^\bullet(\exists x.A) \leftrightarrow \exists x.\ {}^\bullet A$

$(d\ ^\bullet ab)$ $D \vdash {}^\bullet(\{x : A\}y) \leftrightarrow \{x : {}^\bullet A\}y$

$(d\ ^\bullet \forall^2)$ $D \vdash {}^\bullet(\forall P.A) \leftrightarrow \forall P.\ {}^\bullet A$

$(d\ ^\bullet \exists^2)$ $D \vdash {}^\bullet(\exists P.A) \leftrightarrow \exists P.\ {}^\bullet A$

L.10. Elimination rules for circuit terms

$$\frac{c \vdash {}^{\wedge}A}{c\lceil \Lambda \vdash A}$$

$$\frac{c_0 \times c_1 \vdash {}^{\bullet}A}{c_0 \bullet c_1 \vdash A}$$

The elimination rules for the basic components obtained directly from their semantics in 4.2, with equations such as $I\alpha = H$ being understood as abbreviating assertions in the logic, in this case $hx \wedge \neg lx$. For example, the elimination rule for a resistance $res_g(\alpha, \beta)$ takes the form:

$$\begin{aligned} res_g(\alpha,\beta) \vdash & E\alpha \wedge E\beta \wedge (\forall x.\ x = \alpha \vee x = \beta) \wedge \\ & \forall x.\ (\neg hx \wedge \neg lx) \wedge \\ & S\alpha \cdot g \leq S\beta \wedge S\beta \cdot g \leq S\alpha \wedge \\ & (S\alpha \cdot g = S\beta \hookrightarrow \alpha \twoheadrightarrow \beta) \wedge (S\beta \cdot g = S\alpha \hookrightarrow \beta \twoheadrightarrow \alpha)\} \end{aligned}$$

References

[A] Abramsky, S., An intuitionistic logic of computable functions. Copy of slides 1984.

[B1] Bryant, R.E., A switch-level model of MOS circuits. In VLSI '81, Ed. J. Gray, Academic Press 1981.

[B] Bryant, R.E., A switch-level model and simulator for MOS digital systems. IEEE Transactions on Computers C-33 (2) pp. 160-177, February 1984.

[C] Cardelli, L., An algebraic approach to hardware description and verification. Ph.D. thesis. Comp.Sc.Dept., University of Edinburgh (1982).

[CGM] Camilleri, A., Gordon, M., and Melham, T., Hardware verification using higher order logic. To appear in the proceedings of the IFIP International working conference, Grenoble, France, September 1986. Also available as a report 91 of the Computer Laboratory, University of Cambridge (1986).

[D] Dijkstra, E.W., A discipline of programming. Prentice-Hall (1976).

[F] Fourman, M.P., Verification using higher-order specifications and transformations. Department of Electrical Engineering, Brunel University (1986).

[FS] Fourman, M.P.; and Scott, D.S., Sheaves and logic. In proc. of Durham conference on Applications of Sheaves 1977, Lecture notes in Math., Springer-Verlag 1979.

[Gor1] Gordon, M.J.C., LCF-LSM. Report no. 41 of the Computer Laboratory, University of Cambridge (1983).

[Gor2] Gordon, M.J.C., How to specify and verify hardware using higher order logic. Lecture notes, Computer Laboratory, University of Cambridge (1984).

[Gor3] Gordon, M.J.C., Why higher order logic is a good formalism for specifying and verifying hardware. Report no.77 of the Computer Laboratory, Cambridge University 1985.

[GH] Gordon, M.J.C., and Herbert, J., A formal methodology and its approach to a network interface chip. Report no.84 of the Computer Laboratory, Cambridge University 1985.

[Gra] Grayson, R., Heyting-valued semantics. Proc. logic colloquium '82, North-Holland (1984).

[Ha] Harel, D., First-order dynamic logic. Springer Verlag Lecture Notes in Comp.Sc., vol. 68 (1979).

[HD] Hanna, F.K., and Daeche, N., Specification and verification using higher-order logic. Proc. IFIP WG 10.2, 7th. international conference on computer hardware

description languages and their applications, Tokyo, Japan 1985, Koomen & Moto-oka (eds.), North–Holland (1985)

[Hay] Hayes, J., A unified switching theory with applications to VLSI design. Proc. IEEE 70 (10) pp. 1140–1155 Oct. 1982.

[KB] Bergstra, J.A., and Klop, J.W., A proof rule for restoring logic circuits. Integration 1 pp.161–178 (1983).

[MC] Mead, C., and Conway, L., Introduction to VLSI systems. Addison–Wesley (1980).

[MD] Mavor, J., and Denyer, P.B., Introduction to MOS design. Addison Wesley 1983.

[Me] Melham, T., Abstraction in hardware verification. Progress report and thesis proposal, Computer Laboratory, University of Cambridge (1985).

[Mi] Milne, G., CIRCAL, TOPLAS April 1985.

[Mos] Moszkowski, B., Executing temporal logic programs. Report no.71 of the Computer Laboratory, Cambridge University 1985.

[OH] Olderog, E., and Hoare, C.A.R., Specification–oriented semantics for communicating processes. ICALP 83, Springer Lecture Notes in Comp. Sc. vol. 154 (1983).

[P] Plotkin, G.D., Types and partial functions. Lecture notes, Computer Science Dept., University of Edinburgh 1985.

[Pr] Prawitz, D., Natural deduction. Almqvist and Wiksell 1985.

[deR] de Roever, W.P., The quest for compositionality. In the proceedings of the IFIP working conference, January 1985, North–Holland (1985).

[S] Scott, D.S., Identity and existence in intuitionistic logic. In proc. of Durham conference on Applications of Sheaves 1977, Lecture notes in Math., Springer–Verlag 1979.

[She] Sheeran, M., μFP an algebraic VLSI description language. D.Phil. thesis, Oxford 1983.

[St] Stirling, C., Modal logics for communicating systems. Report CSR–193–85 of the Computer Science Dept., Univ. of Edinburgh (1985).

[W] Winskel, G., A complete proof system for SCCS with modal assertions. In the proceedings of Foundations of Software Technology, Springer lecture notes in Comp.Sc. (1985).