A Discrete Switch-Level Circuit Model that uses 4-valued node states

PROEFSCHRIFT

TER VERKRIJGING VAN DE GRAAD VAN DOCTOR AAN DE TECHNISCHE UNIVERSITEIT EINDHOVEN, OP GEZAG VAN DE RECTEUR MAGNIFICUS, PROF. DR. J.H. VAN LINT, VOOR EEN COMMISSIE AANGEWZEEN DOOR HET COLLEGE VAN DEKANEN IN HET OPENBAAR TE VERDEDIGEN OP DONDERDAG 9 DECEMBER 1993 TE 16.00 UUR

DOOR

Wilhelmus Hubertus Ferdinands Josephus Körver

GEBOREN TE HEERLEN
Dit proefschrift is goedgekeurd door de promotoren

prof.dr. M. Rem
en
prof.dr.ing. J.A.G. Jess

druk: wibro dissertatiedrukkerij, helmond.
CONTENTS

0 Introduction
  0.0 Global introduction 1
  0.1 Motivation 4
  0.2 Informal introduction to the model 5
  0.3 Approach and Outline 10
  0.4 Other discrete switch-level circuit models 12
  0.5 Notational remarks 14

1 Basic Model
  1.0 Representation of circuits 17
  1.1 Circuit states 18
  1.2 Circuit behaviour 19
  1.3 Properties of WM0 22
  1.4 Other correctness criteria 33
  1.5 Concluding remarks on chapter 1 41

2 Acyclic Circuits
  2.0 Acyclic circuits 45
    2.0.0 General acyclicity 45
    2.0.1 Acyclicity w.r.t. a source-connection 47
  2.1 Relations between A0, A1 and WM0 50
  2.2 Concluding remarks on chapter 2 55

3 Reaction-delays
  3.0 Modelling arbitrary reaction-delays 57
  3.1 Properties of WM1 and the relation to the basic model 60
  3.2 Modelling restricted reaction-delays 64
  3.3 Remarks on well-functioning 70
  3.4 Concluding remarks on chapter 3 71
4 Charge Storage
  4.0 Introduction to charge storage 74
  4.1 Properties of WM2 81
  4.2 Well-Functioning 92
  4.3 Further research 98
  4.4 Concluding remarks on chapter 4 102

5 Pass-dela ys and Wire-del ays 105
  5.0 Arbitrary pass-dela ys; initial behaviour 106
    5.0.0 Introduction 106
    5.0.1 Extension of the model 112
    5.0.2 Properties of WM3 113
    5.0.3 Additional correctness criteria 116
  5.1 Restricted pass-dela ys; dynamic behaviour 117
    5.1.0 Modelling restrictions on pass-dela ys 118
    5.1.1 Well Functioning 126
    5.1.2 Can pass-dela ys be modelled as part of reaction-del ays? 129
  5.2 Wire-del ays 131
  5.3 Concluding remarks on chapter 5 134

6 Imperfectness of Switches 137
  6.0 Introduction to Mutiation 138
  6.1 Formalisation of Mutiation Degree 142
    6.1.0 Mutiation Degree for Initial Behaviour 142
    6.1.1 Mutiation Degree for Dynamic Behaviour 149
  6.2 Correctness Criteria due to Imperfectness of Switches 151
  6.3 Remarks on the Formalisation 154
  6.4 Concluding remarks on chapter 6 157
7 The Relation between Model and Specifications
   7.0 Specifications of Circuits 159
   7.1 Correctness Criteria w.r.t. Specifications 161
   7.2 Concluding remarks on chapter 7 166

8 Concluding Remarks
   8.0 Remarks on the model 167
   8.1 Topics for further research 169

APPENDICES
A Overview Lattice Theory 171
B Proofs
   B0 Proof of theorem 1.28 176
   B1 Proof of theorem 3.8 184
   B2 Proof of theorem 5.10 189
C On the pessimism caused by assumption (1) on pass-delays 201

References 212
Notation Index 216
Subject Index 218
Samenvatting 220
Curriculum Vitae 222
As far as the properties of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.

A. Einstein
CHAPTER 0  INTRODUCTION

In this thesis a discrete switch-level model for digital CMOS circuits is developed. It describes the logic behaviour — static as well as dynamic — of transistor networks and captures logic faults due to conflicts, undefined gates, hazards, charge-sharing, imperfectness of transistors, and relative-timing problems. Although the theory is directed towards CMOS, it is applicable to every FET (field effect transistor) technology.

In section 0.0 we place discrete switch-level models into perspective with other aspects of CMOS circuit design. Section 0.1 gives a general motivation for switch-level modelling. In section 0.2 the goal of our model and an informal introduction to the model are given. Section 0.3 motivates our approach and gives an outline of the thesis. Section 0.4 gives an overview of other discrete switch-level circuit models and relates the model presented in this thesis to these models. Finally, in section 0.5 some notational conventions used throughout the thesis are given.

0.0  Global introduction

Since Complementary Metal Oxide Silicon (CMOS) has become the major technology for digital circuits, our model concentrates on describing the behaviour of CMOS transistor networks. In this section we explain the place of the model with respect to other circuit design aspects.

We believe that design activities should, like design methods, be based on a top-down hierarchical approach. The intended design trajectory of digital circuits is depicted in figure 0.0. Starting at the top level with a circuit specification, the first step consists of the design of a network of components that performs the specified behaviour. These components themselves also have the form of circuit specifications, and hence this step can be repeated until the derived network has a suitable form, for instance, until all components are from some set of basic components (which not necessarily need to be logic gates, cf. [E1] and [Ber2]). The next step is to construct, for each component, a CMOS transistor network implementing this component. The last step before fabrication leads to the geometric layout at the bottom level.
We will now consider some of these levels in more detail.

**Specifications**

The kind of specifications we consider regards a circuit as a black box with a number of external binary input and output terminals (see figure 0.1). The relation between input events and output events is called the communication behaviour and can be given by a - possibly CSP-like [Hoa2] - expression, as in [Ma1], [BeS], and [El], or by some state transition graph (where states are - completely or partly - defined by the values of the inputs and outputs), as in [HP]. Besides the communication behaviour, timing constraints must be specified. In case of *synchronous networks* (see below) a typical timing constraint gives an upper bound on the amount of time the circuit may use to reach a stable internal state after receiving an input change. In case of *asynchronous networks* (see below) two distinct timing constraints are traditionally distinguished, namely *fundamental mode* and *input-output mode* [BrzE]. Fundamental mode means that the environment of the circuit will not send the next input change until the circuit has internally reached a stable state. Input-output mode means that the next input change will be sent by the environment only after the expected output change is accomplished by the circuit, which may occur before the circuit has internally reached a stable state. Notice that both modes mainly restrict the environment of the circuit, and that for fundamental mode this restriction on the environment is much stronger than it is for input-output mode.

```
circuit specification
network of components ⊂
  network of transistors ⊂
layout
```

**Networks of components**

Networks of components can be synchronous or asynchronous. In synchronous networks the synchronisation of the computation (steps) is realised by a global clock. In asynchronous networks this synchronisation is realised by the communication between the components in the network. In order to verify whether a network of components behaves as specified, a formal model describing the behaviour of networks of components is needed, e.g. [BrzS1] or [Sc]. Models describing the behaviour of networks of components are called *gate-level models*. To derive such networks from a circuit specification a number of decomposition methods have been developed, e.g. [El], [Ma1], [BeS], and [Ber2].
Networks of transistors

While components can be usefully modelled as unidirectional and binary elements, transistors used in a more general way display behaviour that cannot be captured by such restricted models. Transistors are bidirectional devices and they are actually used as such in practical designs (e.g. in [Ber1]). The behaviour of transistors cannot be specified using binary values only; we return to this point in section 0.2. For these reasons gate-level models are not suitable to describe the behaviour of transistor networks. Models describing the behaviour of networks of transistors are called transistor models or switch-level models. They can describe transistor networks at various levels of abstraction. The most significant distinction is between continuous transistor models and discrete transistor models, the latter abstracting from the continuous physical behaviour of transistors, voltage changes, etc. A discrete switch-level model intends to function as an intermediate model – at transistor level – between the – discrete – higher level models on the one hand, and the – continuous – lower level models on the other hand (cf. section 0.1). The formal relation between the – discrete – specification and – discrete – gate-level models on the one hand and switch-level models on the other hand is very important (see below and section 0.1), and can more easily be given for discrete switch-level models. We therefore focus our attention on discrete switch-level models.

For a discrete switch-level model two relations are of major importance. The first is the relation to the underlying technology, which is necessary to validate the adequacy of the model. It describes the correspondence between physical behaviour and modelled behaviour. Which means that it relates physical correctness of a circuit, that is, absence of undesired behaviour in reality or in some continuous model, on the one hand, and correctness of that circuit within the discrete model, that is, no detection of undesired behaviour by the model, on the other hand. In section 0.2 we discuss the notion ‘undesired behaviour’ and describe the types of faults we consider in our model. Since a discrete model abstracts from the (continuous) physical behaviour, correctness within the discrete model need not correspond exactly with physical correctness. The model is called optimistic if physically correct behaviour of a circuit implies correctness of the modelled behaviour. In that case, only physical incorrectness of a circuit can be concluded (from incorrectness within the model) but not physical correctness. The model is called pessimistic (or conservative, cf. [SiB]) if incorrect physical behaviour implies incorrectness of the modelled behaviour. In this case, physical correctness can be concluded (from correctness within the model). A model that mixes optimism and pessimism is useless for formal verification purposes (although it may be useful in early stages of design exploration): no conclusions about the physical behaviour can be drawn from such a model. Since we wish to verify the physical correctness of circuits, we will accept a tendency to pessimism but none to optimism. Too much pessimism, however, leads to rejecting too many correct circuits. And hence, in order to describe realistically many correct circuits accurately, this tendency to
pessimism must be sufficiently small, i.e., the model must be sufficiently accurate in describing circuit behaviour.

The second relation that is of importance for switch-level circuit models is the relation to circuit specifications. We will consider only specifications of circuits (or components) placed in an asynchronous environment. This means that (i) we consider only the fundamental mode assumption and the input-output mode assumption, but we do not give any absolute delay estimates, and (ii) (static and dynamic) hazards and runt pulses on the outputs must be avoided. This does not mean that the model cannot be used for synchronous circuits since components placed in a synchronous environment can be regarded as components in an asynchronous environment with the fundamental mode assumption, where, depending on the synchronous protocol of the environment, hazards may sometimes be neglected. The relation to specifications is discussed in chapter 7.

Whether the model conveniently describes physical reality cannot be proven, since it is logically impossible to formalise the relation between modelled behaviour and real behaviour; the best one can do is to give the relation to a lower level model of the same reality. The relation to physical reality, or to some lower level continuous model, is given intuitively throughout the thesis. On the one hand, we concentrate on studying the physical aspects carefully and defining the necessary notions in an intuitively correct and accurate way. On the other hand, once the notions are defined, our model is purely mathematical, and we study the properties of the defined notions and the relations between them. Despite the restrictions of a discrete model we have the impression that our model correctly describes a sufficiently large class of circuits.

0.1 Motivation

This section gives a general motivation for studying discrete switch-level circuit models. This is done by giving a number of goals for which a formal switch-level model is required. The motivation for our model in particular — for our approach and with respect to other switch-level models — is given later (sections 0.2, 0.3, and 0.4).

As said above a discrete switch-level circuit model intends to build a bridge between the discrete higher levels (specification, gate-level) and the continuous lower levels (continuous transistor models, layout models). The first two goals below illustrate this ‘bridge function’.

The first goal is to lay a formal mathematical basis for proving correctness of higher level models, calculi, and design methods or strategies, both gate-level and switch-level. In order to prove the correctness of a circuit model (e.g. gate-level: [BrzS1], [Se], switch level: [Hoa2], [BeS], [Se], [BrzY], [BeK]) or of decomposition, transformation and calculation rules of a calculus (e.g. gate-level: [B1], [Ma1], [BeS], switch-level: [Ber1], [Ber2], [K1], [K2]), an
underlying mathematical model is required as a semantic domain.

The second goal is the verification or derivation of the exact conditions for correct behaviour of
a switch-level design, e.g. with respect to delay restrictions. A switch-level model enables the
verification of a designed circuit before implementing it in hardware, thus providing a valuable
check along the path to the (expensive) fabrication step. Moreover, a switch-level model can
guide the translation of a switch-level design to a lower level implementation; the conditions for
correctness of a lower level implementation (e.g. layout) of a logic design can be derived with
the switch-level model. These conditions are required for the translation of a design to this
lower level, e.g., relative-delay restrictions for transistors can be derived with a switch-level
model and they constrain relative sizes of the transistors (a layout-level concern).

Furthermore, a switch-level model is able to predict the behaviour of a design in case of certain
hardware errors such as stuck-at faults, which is important with respect to testing. For these
goals it may be attractive to build a switch-level simulator, for which an underlying formal
model evidently is required. That is, a discrete switch-level model is useful for fault analysis.

Finally, a discrete switch-level circuit model may help to understand the basic problems of
switch-level design and may thereby lay a (formal) basis for a switch-level design method.

0.2 Informal introduction to the model

Our model intends to describe the logical behaviour of switch-level circuits in such a way that
(A) this description is intuitively correct and accurate such that possible 'undesired behaviour',
or 'logic faults', can be detected, and (B) the description is formal as a result of which the
mathematical properties of the defined notions can be investigated, and the relation to the
circuit's specification can be formalised.

Undesired behaviour includes undefined switch gates, (static and transient) conflicts, and (static
and dynamic) hazards. They can, among other things, be caused by charge-sharing, by relative
timing of signals, or by imperfectness of switches. We return to these phenomena later on.

In order to give the relation to the specification it is necessary to investigate two types of
behaviour: (1) initial behaviour, i.e., the behaviour of a circuit with stable inputs when the
previous state of the circuit is unknown; and (2) dynamic behaviour, i.e., the behaviour of a
circuit starting in a – known – state when the input-values change. The main questions to be
answered are: 'What are the possible resulting states?', and: 'Are these resulting states correct,
i.e., free of undesired behaviour?'.

In case (1), all possible stable and oscillating states must be considered as possible resulting
states, whereas the knowledge about the previous state in (2) can restrict the states to be
considered as possible resulting states, viz. to those that are reachable from the starting state.
On the other hand, in case (2), also possible 'intermediate' states can be considered and must be checked for undesired behaviour. As a result of this, the two types of behaviour are essentially different. The two central notions in our model concerning correctness of behaviour are well-matchedness and well-functioning; the former is related to (1) above and the latter to (2) above.

For a thorough introduction to digital design using CMOS we refer to [WE], [MC], or [Braa]. We now proceed with a simplified explanation of switch-level circuits and address some of the aspects we are interested in. By means of simple examples we illustrate the subtlety of some types of 'undesired behaviour' and 'logic faults'.

A switch-level circuit can be regarded as a collection of sources and transistors connected by wires. In our model wires and switches are connected at nodes. Our model then studies the behaviour of arbitrary networks of switches and wires connected at nodes. All nodes connected to sources are treated as inputs. We discuss several aspects separately.

Sources
There are two types of sources: high-voltage sources (connected to POWER), which we call 'H-sources', and low-voltage sources (connected to GROUND), which we call 'L-sources'. Conventionally, a connection to an H-source represents the logical '1' (or true), and a connection to an L-source represents the logical '0' (or false).

Nodes
A node represents a point on a wire, or, if the wire is assumed to cause no delay, the complete wire. A node is called high if it is connected, via a path of zero or more conducting switches, to an H-source but not connected to an L-source; it is called low if it is connected to an L-source but not connected to an H-source; if it is not connected to any source it is called floating; and if it is connected to both kinds of sources it is called conflicting (or fighting). Since conflicts can damage the circuit, they must be avoided (see 'states' below).

Switches
Two types of switches are considered: the n-switch and the p-switch (modelling the n-mos enhancement mode transistor and p-mos enhancement mode transistor). They are depicted as in figure 0.2 below. A switch is connected to three nodes: one gate node (labelled g in figure 0.2) and two pass nodes.

figure 0.2:  n-switch:  \[\text{\_\_\_\_}\] ;  p-switch:  \[\text{\_\_\_\_}\]
The simplified behaviour of switches is as follows. The voltage value on the gate controls the switching state of the transistor. If the switching state is conducting (or ON) the pass nodes are connected, and if the switching state is nonconducting (or OFF) the pass nodes are disconnected. An n-switch is in the conducting state if the gate is high, and in the nonconducting state if the gate is low. A p-switch behaves in a converse fashion; it is conducting if the gate is low, and nonconducting if the gate is high. Transistors are, however, not perfect switches; n-transistors are able to pass low voltages well, but, due to threshold effects, they "mutate" high voltages. Similarly, p-transistors are able to pass high voltages well, but they "mutate" low voltages (see chapter 6 for a more detailed explanation). Switches are bidirectional with respect to the connection of the pass nodes, and unidirectional with respect to the influence of the voltage at the gate node on the conductance state.

We distinguish two types of inconsistency with respect to switches: type 0, if the conductance state of a switch does not correspond to the gate value; and type 1, if the switch is conducting, and therefore the pass nodes are connected, but the pass nodes have different states.

**States**

A state of a circuit with a given source-connection is determined by the states of the nodes and the switches in the circuit. A node can have state $\emptyset$, $\{L\}$, $\{H\}$, or $\{L, H\}$ (corresponding with floating, low, high, and conflicting). A switch can have states 1 or 0 (corresponding with conducting and nonconducting). In chapter 1, this choice is explained in more detail.

The states of nodes will be subdivided later on in order to increase the accuracy of the model, e.g. with respect to charge storage or imperfection of switches. The notion of state will be extended even further in order to model various types of delay (see 'delays' below). We now give some elementary examples.

First consider the circuit depicted in figure 0.3a, where $x_0$ and $x_1$ are input nodes and $z$ is an output node. If $x_0$ and $x_1$ are not directly connected to sources, the state of both of them clearly is $\emptyset$ in the resulting global state, but the state of the switches and therefore the state of $z$ in the resulting global state is unclear. Floating or conflicting gate nodes are called undefined and should not occur in stable states of properly designed circuits. The second type of undesired behaviour occurs if $x_0$ is low and $x_1$ is high. Since both switches are, due to their gate value, conducting in the resulting state, this source-connection (i.e. input combination) leads to a (static) conflict on $z$.

![Figure 0.3a](image1)

![Figure 0.3b](image2)
Now consider the circuit depicted in figure 0.3b; it is the usual CMOS implementation of an inverter. Node $x$ is the input and $y$ is the output. Let $x$ be connected to an H-source, and let the circuit be in the (resulting, stable) state where the p-switch is nonconducting, the n-switch is conducting, and node $y$ is low. If now $x$ is changed to low, this will cause both switches to change state, viz. the p-switch from 0 to 1 and the n-switch from 1 to 0. Depending on the relative timing of these actions, node $y$ will change directly from low to high, or via an intermediate — and transient — floating or conflicting state. Such a transient conflict can also damage the circuit (if it continues long enough), and is therefore also considered to be undesired behaviour. In this example, it can be avoided by restricting the delays (see 'delays' below). In a lower level modelling the switching from 1 to 0 can be regarded as increasing the internal resistance of the switch from "practically 0" to "practically $\infty$", and the switching from 0 to 1 as decreasing the internal resistance. Avoiding the transient conflict now means that the internal resistances are not allowed to be close enough to "practically 0" simultaneously and long enough to be able to cause a 'damaging' conflict on $y$. Notice also that the switching period in the lower level model is abstracted to a 'switching point action'; this will lead to a correctness criterion on state transitions (section 1.4).

**Delays**

In order to be able to capture 'relative-timing' variations and investigate the effects of delays, we distinguish three logic types of delay: wire-delay, pass-delay, and reaction-delay. Wire-delay models the time it takes a voltage transition to pass from one end of a wire to the other. Pass-delay is comparable to wire-delay and models the time it takes a voltage transition to pass from one pass node of a conducting switch to the other. Reaction-delay models the time it takes for a switch to adapt its conductance state to the state of the gate node. We will study relative, abstract delays which are assumed to be nonnegative and finite. Restrictions on delays are often necessary to guarantee correct functioning of a circuit (e.g. in the inverter example above). These delay restrictions restrict the lower level implementation of the circuit, e.g., with respect to transistor sizes. It is possible that these delay restrictions cannot be met by any physical implementation of the circuit, which, of course, makes the translation to the layout level (see section 0.0) impossible. Although it is important to notice this, it does not bother us in this study of abstract circuits; it is an aspect of the technology and not one of the logic behaviour of abstract circuits. The study of the consequences of delay restrictions for a lower level implementation, and of the sort of delay restrictions that are feasible in current technology are entirely different topics and fall outside the scope of this thesis.
Charge Storage
The capability of nodes and wires to store charge is modelled by means of a notion of capacitance. This notion is used only to model charge storage. The delays caused by capacitances can be modelled as pass-delays or wire-delays. The study of the relation between capacitances and delays falls outside the scope of this thesis. Capacitance values retain the previous states of nodes, and can therefore also be high, low, floating, or conflicting. Capacitances are assumed to be large enough to maintain — temporarily — a controlled switch in its conductance state if this state is consistent (type 0, see 'switches' above) with the capacitance value of the gate node. Of course, capacitance values are assumed to be so small that they can be overruled by source values. Although conflicts at capacitance level (mostly due to charge-sharing) are considered to be harmless, they can cause undefinedness of gates (cf. chapter 4). In order for the model to be accurate enough to "explain" sequential behaviour of circuits the modelling of charge storage is necessary.

 Resistances
The resistances we consider (briefly) in our model (section 4.3) are assumed to be so small that a passing signal is capable of charging a node or a wire (and controlling a switch), but so large that a signal that did not pass such a resistance overrules a conflicting signal that passed the resistance. For the circuit depicted in figure 0.4a, an n-mos inverter, this means that if input node \( x \) is low, and, consequently, switch \( s \) is nonconducting, output node \( y \) will be charged and become high. If node \( x \) is high, and, consequently, switch \( s \) is conducting, the low signal from the L-source will overrule the high signal that passed the resistance and \( y \) will become low. This kind of usage of resistances (pull-up, pull-down) is typical for n-mos and p-mos.

Since CMOS hardly uses resistances (with some exceptions, e.g. in [Ma2]) we will — almost completely (except briefly in section 4.3) — ignore resistances (cf. section 0.3). Abstracting from resistances, however, means that we need to be careful with conclusions about the resulting state. We will explain this by means of the following example.

In the circuit depicted in figure 0.4b it may seem obvious that the resulting state, with \( x, y, \) and \( z \) connected to L, H, and L respectively, is the state where both switches are nonconducting and \( x, y, \) and \( z \) are low, high and low respectively.

figure 0.4a

figure 0.4b
However, the circuit depicted in figure 0.4b could be an abstraction from the circuit depicted in figure 0.4c. For the circuit depicted in figure 0.4c it is possible that both switches in the resulting state are conducting; nodes \( x, y_0, \) and \( z_1 \) are low; and nodes \( y_1 \) and \( z_0 \) are high (in \( y_0 \) the connection to the source in \( x \) overrules the connection to the source in \( y_1 \)). So besides the obvious one mentioned above, another resulting state is possible. In the abstract circuit (fig. 0.4b) this will be modeled as a possible resulting state with both switches conducting and all three nodes conflicting. By the way, notice that the circuit depicted in figure 0.4c consists of an n-mos inverter and a p-mos inverter connected in a cycle, as depicted in figure 0.4d.

\[ \text{figure 0.4c} \]

\[ \text{figure 0.4d} \]

### 0.3 Approach and Outline

Since the behavior of switch-level circuits is complex, we start by making a number of assumptions and abstractions. Once we have modeled this "simplified" reality, we refine the model and weaken the assumptions. This approach enables us to study the various aspects of switch-level circuits in isolation and keep our grip on the theory. Furthermore, it enables the investigation of the consequences of each refinement. Convenient assumptions and abstractions are:

(a) perfect switches: switches are assumed to be perfect, which means that the pass nodes are disconnected if the switch is nonconducting, and, if the switch is conducting, the pass nodes are connected and, hence, have exactly the same voltage value;

(b) pass-delays and wire-delays are negligible;

(c) reaction-delays are uniform (, positive, and finite): all inconsistent (type 0; cf. 'switches' in section 0.2) switches change their conductance states simultaneously;

(d) no charge storage: the capability of nodes (wires) to store charge is not considered;

(e) no resistances: resistances are not considered.

It is important to understand the difference between (a), (b) and (c) on the one hand, and (d) and (e) on the other hand. Restrictions (a), (b), and (c) make assumptions about aspects of reality, e.g., assume a specific delay-value, and the model should therefore be interpreted in a (restricted) reality that fulfills these assumptions. Restrictions (d) and (e) abstract from aspects
of reality, which means that the model does not take certain aspect of reality into account (but does not restrict the 'reality' that is described). The consequence for the model-reality relation (section 0.0) is that as far as (a), (b), and (c) are concerned, the model is allowed to be optimistic, viz. if these assumptions are unrealistic, and as far as (d) and (e) are concerned, the model must be pessimistic, and will indeed be pessimistic in those cases where the correct functioning of a circuit depends on charge storage (e.g. a latch) or resistances (e.g. a pull-up).

Outline of the thesis
In chapter 1 all assumptions and abstractions mentioned above are used and their convenience is explained. A basic model is developed which serves as a starting point for the extended models given in later chapters. The basic ingredients of the model are defined (relative to the assumptions). Since charge storage is not modelled in the basic model ((d) above), and therefore sequential behaviour cannot be modelled, we concentrate on the analysis of initial behaviour, i.e., on well-matchedness (see section 0.2). It turns out that well-matchedness cannot, in general, be calculated efficiently (even in this restricted model).
Chapter 2 defines several classes of acyclic circuits, for which well-matchedness can be calculated efficiently. Well-matchedness for these circuit classes is investigated and characterised by a surprisingly simple formula.
In chapters 3, 4, and 5 the assumptions (c), (d), and (b) are relaxed. The effects of delays and charge storage are studied and the notions defined in the basic model are redefined in order to capture delays and charge storage. The properties of the new notions and their relation to the old notions are investigated. Besides well-matchedness also well-functioning (see section 0.2) is defined. Restrictions on delays (see section 0.2) are formalised. In chapter 4 the effects of the omission of restriction (e) above are also regarded (but not developed in detail).
In chapter 6 the imperfectness of switches is explained and formalised (restriction (a)). The main notion in this chapter is called mutilation degree. The mutilation degree of signals is a new notion that enables a concise and elegant description of the effects of the imperfectness of switches. It can be calculated separate from the notions defined in the previous chapters. Consequently, this extension does not affect the underlying model. Additional correctness criteria, depending on the design rules regarding imperfectness of switches, are formulated.
In chapter 7 the relation between modelled behaviour of circuits and specified behaviour of circuits is studied. The main goal is to analyse what kind of correctness criteria are required to link modelled behaviour to specified behaviour, and to demonstrate that these criteria can be expressed in our model. The fundamental mode assumption as well as the input-output mode assumption (section 0.0) are considered.
Each chapter ends with a summary and discussion of the results obtained.
Finally, in chapter 8 concluding remarks and suggestions for further research are made.
0.4 Other discrete switch-level circuit models

In this section some of the features of other discrete switch-level circuit models are discussed. There is no pretension (or intention) of giving a complete historical overview of the development of discrete switch-level modelling. At some places, a comparison to our approach is given. At the end of the section some concluding remarks are made.

In [Bry1], Bryant presents a model that uses ternary node states (0 and 1 for low and high voltage values, and X for an unknown or intermediate value) and ternary switch states (0 and 1 for off and on respectively, and X for unknown). The values at nodes are extended with strengths to denote charge storage (non-driven 'retained' signals) and resistances (driven signals that passed a resistance). The disadvantage of using ternary values is that the unknown value X is used for floating, for improper charge-storage, as well as for short circuits (conflicts), and, consequently, leaves no possibility of distinguishing between the possibly damaging conflicts and the —in first instance— harmless other two phenomena. The model uses a unit (reaction-) delay, that is, all transistors switch with the same delay after their gate values change. The effects of imperfectness of switches are not taken into account. Bryant's model, and also the models discussed in the next paragraph, have no intentional pessimism or optimism (see section 0.0), with the result that they mix optimism and pessimism.

The models presented in [BrzS1], [BrzS2], [Se], [SBrz], and [SBry], are based on Bryant's model. In these models the fundamental mode assumption (section 0.0) is made. Delays are extended to arbitrary delays, with the assumption that delays are inertial, which means that if the state is unstable for only a short period of time, the switch state does not have to change. In our model such unstable periods are classified as incorrect because of the danger of runt pulses (cf. section 1.4, criterion c29). The arbitrary delay model leads to so-called transient cycles, which are cycles of oscillating states with an infinite delay at one of the gates. These transient cycles are excluded from circuit behaviour, using the reasonable assumption of finite delays. We will demonstrate that the transient cycles can be avoided by modelling delays with the help of countdown functions (cf. section 3.0).

In [BrzY] a similar model is used to explain the effects of imperfectness of switches. The modelling of this imperfectness is given only for combinational circuits. Different models are required for the different design rules described in chapter 6 of this thesis. The ideas presented in [BrzY] are used in [BrzS1] and [Se] to give a similar modelling of the effects of the imperfectness of switches, also for sequential circuits (still under the fundamental mode assumption). For this purpose, different models are presented, some of which use the assumption "good paths override bad paths" (cf. [BrzS1] and [Se,appendix], models 2 and 4). This
assumption leads to a far too optimistic modelling of behaviour. For instance, in the circuit
depicted in figure 0.5, these models predict a 0-output at node y when the input x is 1 (and,
hence, both transistors are conducting), totally neglecting the certain (damaging) conflict.

![Figure 0.5](image)

The model presented in [Wi1] is also based on Bryant’s model, but uses a four-valued node state
(extendable to (3N+1)-valued, see next paragraph). The main goal of the presented model and
logic is to achieve composability. The circuits that are considered are combinational circuits in a
static environment (fundamental model!). Imperfectness of switches is not taken into account.
The logic presented, though elegant, is complicated even for these static circuits.

In [Ha1], Hayes proposes the basic ideas of a four-valued node state, comparable to the ones
used in this thesis. Delays, however, are modelled in a completely different way. The
imperfectness of switches is handled in a traditional way, namely, by avoiding problems due to
imperfectness by using traditional complementary circuitry (with, e.g. a block of n-transistors
for the connection to 0). In [Ha2] it is explained how the four-valued node state can be extended
to a (3N+1)-valued node state using a kind of strength for three of the four original values. In
this model, a translation step is required from a transistor network to a so-called CSA network,
consisting of connectors, switches, and attenuators. An additional element, called a ‘well’, is
used to model the rise and fall delays of transistors due to charge storage effects. The theory
presented is directed towards design, and gives no complete description of the modelled
behavioural aspects.

In [ZH], Hoare and Zhou give a convincing motivation for discrete switch-level modelling. The
model presented in [ZH] and [Ho2] uses a four-valued node state that is equal (except for
notation) to $\mathcal{A}(L,H)$ used here. The switch behaviour is not defined in case the gate is
undefined (floating or conflicting), thereby possibly excluding the possibility to detect undesired
behaviour. This danger seems to be the most imminent in the initialisation phase, where possible
incorrect stable states may remain undetected. The approach is directed towards simulation, and
describes a number of phases of circuit behaviour. Certain types of undesired transient
behaviour, like transient conflicts and transient charge-sharing at gates, cannot be detected.
Most of the models discussed above are less detailed than the model presented in this thesis, in particular with respect to node states (recall our objections against the X-state) and w.r.t. the consequences of delays. Consequently, some of them (like [ZHI]) can be considered as less accurate higher-level discrete switch-level circuit models, and would, in order to verify or correct the way they model circuit behaviour, require a more detailed model, like the one presented in this thesis, as a semantic domain. The models are often strongly directed towards the efficiency of simulation based on the model. Without doubt a noble goal, but it should not lead to an unnoticed loss of accuracy. In our opinion, modelling of reality should have accuracy as the primary goal. The second most important goal, efficiency, is to be achieved by mathematical manipulation of the derived model (and, hence, keeping track of possible exchanges of accuracy for efficiency).

What we missed in the models we have discussed above is a well-defined relation between the modelled behaviour and specified behaviour (although often suggested intuitively). Some of the models only deal with combinational circuits. All of them use, implicitly or explicitly, the fundamental mode assumption (section 0.0). The effects of imperfectness of switches are usually also neglected.

A number of the features of these models have influenced the way circuit behaviour is modelled in this thesis. As said in [Wi2], a formal comparison between models has two main advantages. The first one is already mentioned above, namely, the validation of simpler models using more accurate models as semantic domain. Another advantage is that results that can be proved easily in some models, can possibly, on account of this comparison, be used in others.

### 0.5 Notational remarks

The set of functions from A to B is denoted as $A \rightarrow B$. Consequently, $F = A \rightarrow B$ denotes that $F$ is the set of functions from A to B, and $f \in A \rightarrow B$ denotes that $f$ is a function from A to B. Operator $\rightarrow$ is right binding, that is, $A \rightarrow B \rightarrow C$ must be read as $A \rightarrow (B \rightarrow C)$.

Function application is denoted by the infix operator $\cdot$ ('dot'; [Hool]). Operator $\cdot$ is left binding, that is, $f \cdot x \cdot y$ must be read as $(f \cdot x) \cdot y$.

Operators $\cdot$ and $\rightarrow$ bind stronger than all other operators.

The set of natural numbers is denoted by $\mathbb{N}$ (and includes zero: $0 \in \mathbb{N}$). The set of Boolean values is denoted by $\mathbb{B}$. We use $\mathbb{B} = \{0, 1\}$, where 0 and 1 correspond with $\text{false}$ and $\text{true}$ respectively.

Predicates on a set are not formally distinguished from subsets of that set (cf. [Hool]). So we can write $U \cdot x$ instead of $x \in U$, and the powerset $\mathcal{P}(X)$ of a set $X$ satisfies $\mathcal{P}(X) = X \rightarrow \mathbb{B}$. 
The priority of operators is defined as follows, in order of decreasing binding power:

\[ \cdot, \rightarrow \quad \text{with highest binding power} \]
\[ \land, \lor, \land \]
\[ \in, \notin, \subseteq, \subset, \subseteq, \supset \]
\[ =, \neq \]
\[ \land, \lor \]
\[ \Rightarrow, \Leftarrow, \equiv \quad \text{with lowest binding power} \]

Instead of operator \( = \) and \( \equiv \) we often use \( = . \)

The set of infinite lists over a set \( A \) and the set of non-empty finite lists over a set \( A \) are denoted as \( \mathcal{L}(A) \) and \( \mathcal{L}^+(A) \) respectively.

\[ \mathcal{L}(A) = \mathbb{N} \to A \]
\[ \mathcal{L}^+(A) = (\mathbb{N} : n \in \mathbb{N} : \{ i : 0 \leq i < m : i \to A \}) \]

The length of a list \( L \in \mathcal{L}(A) \cup \mathcal{L}^+(A) \) is denoted as \( \#_L \), with the obvious convention that \( \#_L = 0 \) if \( L \in \mathcal{L}(A) \). Elements of a list \( L \) are called \( L_i \), for \( 0 \leq i < \#_L \).

The concatenation of lists \( L_0 \) and \( L_1 \) is denoted as \( \text{cat}(L_0, L_1) \), and the repetition of a finite list \( L \) is denoted as \( *_L \).

The notation for quantifications is adopted from [DF]. We will introduce it informally here.

Universal quantification, i.e., generalized conjunction, is denoted by \( (A \, l : R : E) \), where \( A \) is the quantifier, \( l \) is a list of bound variables, \( R \) is a predicate that delineates the range of the bound variables, and \( E \) is the quantified expression. In general, both \( R \) and \( E \) will contain variables from \( l \). Likewise, we denote existential quantification, union and intersection using quantifiers \( \exists \, \cup \), and \( \cap \) respectively.

Besides the usual notation, we also use quantification for sets. So \( \{ 3 \} \) can also be denoted by \( \{ i \mid i = 3 \} \) or by \( \{ i : i = 3 \} \). Variables that range over numbers, range over natural numbers.

So \( \{ 3 \} \) can also be denoted by \( \{ i : 3 \leq i < 4 \} \).

The following denotation of proofs is also adopted from [DF]. For expressions \( E \) and \( G \), an expression of the form \( E \Rightarrow G \) will often be proved by introduction of intermediate expressions.

For instance, if we can prove \( E \Rightarrow G \) by proving \( E = F \) and \( F \Rightarrow G \), we record this proof as follows:

\[
E \\
= \{ \text{hint why } E = F \} \\
F \\
\Rightarrow \{ \text{hint why } F \Rightarrow G \} \\
G
\]
This notation prevents us from writing down intermediate expressions like \( F \) twice. The major advantage of this denotation is that verification of proofs is extremely simple (viz. stepwise using the hints). Throughout the thesis, the presentation of proofs is given the attention it deserves (cf. [Gas], where, besides the presentation, also some heuristics for proof design are discussed).

We will use a number of notions and results from Lattice Theory, of which an overview of theory and terminology is given in appendix A.
CHAPTER 1 BASIC MODEL

The model presented in this chapter is a basic model; it serves as a framework for the extended models given in later chapters. The restrictions mentioned in section 0.3 are used, which are:

(a) switches are perfect;
(b) wire-delays and pass-delays are zero;
(c) reaction-delay is uniform, positive, and finite;
(d) no capability for charge storage exists;
(e) resistances are neglected.

Restrictions (a,d,e) are convenient, since they allow a simple notion of state. Restrictions (b,c) are convenient, since they lead to a simple next-state function. The simplifications resulting from these restrictions, and thereby also the convenience of these assumptions, are explained further in sections 1.1 and 1.2.

Since charge storage is not modelled in this basic model ((d) above), sequential behaviour of circuits cannot be modelled. Although we are interested in dynamic as well as in initial behaviour of circuits (cf. section 0.2), we therefore concentrate in this chapter on initial behaviour. In later chapters dynamic behaviour will also be modelled.

The chapter is organized as follows.
In the first section a formal representation of circuits is presented. In the second section circuit states are defined. The behaviour of circuits is analysed in the third section, where a number of key notions in the model are defined, including the main correctness criterion for initial behaviour. This criterion is expressed in the notion well-matchedness. This notion is investigated further in the fourth section, where a number of its properties are derived. In the fifth section two additional correctness criteria for initial behaviour are defined.
In the final section the major results derived in chapter 1 are summarised and discussed.
1.0 Representation of circuits

In this section, a formal representation of circuits is given. The kind of circuits we want to study can be regarded as arbitrary collections of sources and switches connected by wires. All connections to sources, permanent connections as well as inputs, are treated in the same way; they are left out of the formal circuit representation and are taken care of by suitably chosen source connections (see below). Since we assume that wire-delay is negligible, thus assuming that all points on a wire have exactly the same voltage value, a wire can be represented by one node. Consequently, a circuit can be viewed as a set of nodes and a set of switches. A switch can be characterized by its type, its gate node, and its pass nodes.

1.0 Definition

A circuit is a quintuple $\langle N, SW, t, g, pn \rangle$, where:

- $N$ is a finite set (set of nodes)
- $SW$ is a finite set (set of switches)
- $N \cap SW = \emptyset$
- $t \in SW \rightarrow \{L, H\}$ (switch type)
- $g \in SW \rightarrow N$ (switch gate)
- $pn \in SW \rightarrow B2N$ (switch pass node pair)

where $B2N$ is the set of two-element-bags over $N$.

The set of circuits is called CIR.

Typical names used for elements of CIR are $C$ and $C_i$; for elements of $N$: $x$, $y$, and $z$; and for elements of $SW$: $s$ and $s_i$.

For a switch $s$, $t: s = \{L\}$ denotes that $s$ is a p-switch, and $t: s = \{H\}$ denotes that $s$ is an n-switch. The reason for choosing \{L\} and \{H\} as switch types, and not for instance more descriptive names like $pst$ and $nst$, has to do with the elegance of definitions and will become clear later on.

Elements of $B2N$ are two-element bags. A bag with elements $x$ and $y$ is denoted by $\{x, y\}$; evidently $\{x, y\} = \{y, x\}$.

1.1 Example

Circuit $C_0 = \langle \{x, y, z\}, \{x, \{s, \{H\}\}\}, \{\{s, \{y\}\}, \{\{s, \{x, z\}\}\}\rangle \rangle$ consists of one n-switch. The familiar drawing is:

```
  y
 ____________
x         z
```

Source connections

Two types of sources are considered: low-voltage sources (connected to GROUND), called L-sources, and high-voltage sources (connected to POWER), called H-sources.

A circuit node can be directly connected to an H-source, to an L-source, to neither or to both of them. Such a direct connection can be to a permanent source or through an input of the circuit.

For a circuit \( C = (N, SW, t, \phi, pn) \), the direct connection to sources (for all circuit nodes) is represented by a function \( \gamma : \gamma \in N \rightarrow \mathcal{P}(\{L,H\}) \), called the source-connection of \( C \), with the obvious convention that:

\[ H \in \gamma x \iff \text{"node } x \text{ is directly connected to an H-source"} \]

\[ L \in \gamma x \iff \text{"node } x \text{ is directly connected to an L-source"} \]

where 'directly' means 'without switches in between'.

1.2 Example

Consider circuit \( C_0 \) (ex. 1.1) with source-connection \( \gamma_0 = \{(x,\{L\}),(y,\{H\}),(z,\emptyset)\} \). This situation is drawn as:

![Diagram](image)

1.1 Circuit states

We are interested in the possible behaviour of a circuit in combination with a source-connection, say \( C = (N, SW, t, \phi, pn) \) with \( \gamma \). This means that we want to analyse the states of \( C \) that may occur as results of \( \gamma \). In order to do so, we first define the notion 'state', then analyze which states can be 'stable' or 'oscillating', and finally consider correctness criteria for these states. We define a circuit state as a combination of a node-state and a switch-state; these states are introduced in this section. The circuit behaviour is investigated in the next section.

Node states

The state of a node describes what types of sources the node is (indirectly) connected to. Since two types of sources are considered, the state of a node can have four values, viz. no connection to a source, a connection to an L-source but no connection to an H-source or vice versa, and connections to both types of sources. We want to be able to distinguish all four possibilities.

First of all, the connections to a single source-type must be distinguished from the other possibilities, because (A) these single-source connections represent the logical values (cf. section 0.2) and therefore are necessary to relate circuit behaviour to specified behaviour (which often is in terms of Boolean values), and (B) they determine (correct) switch behaviour, as explained in sect. 1.2. Finally, we want to distinguish 'no connection' from 'both connections', since we allow the former to occur in a correct circuit state but not the latter (since it can damage the circuit).
We therefore choose the elements of $\mathcal{P}(\{L,H\})$ to represent the state of a node, with the obvious interpretation. A global state of all the nodes, called a node state of the circuit, can then be represented by an element of $N = \mathcal{P}(\{L,H\})$.

Since charge storage and resistances are not considered and switches are assumed to be perfect (restrictions (d),(e),(a)), the four values $\emptyset$, $\{L\}$, $\{H\}$, and $\{L,H\}$ are sufficient to describe node-states in the basic model. They will be subdivided later on when these restrictions are weakened (chapters 4 and 6).

Notice that multiple connections to the same source type are not distinguished from single connections to that source type.

1.3 Definition

The set of node-states $\text{NST}$ and the (partial) order $\preceq$ on $\text{NST}$ are defined by:

$\text{NST} = N \to \mathcal{P}(\{L,H\})$

$\Delta \preceq \Gamma = (A x : N \times : \Delta x \subseteq \Gamma x) \quad \text{for} \quad [\Delta, \Gamma] \subseteq \text{NST}$

Typical elements of $\text{NST}$ will be called $\Gamma$, $\Gamma_0$, or $\Delta$, and, in case they specifically denote source-connections: $\gamma$, $\gamma_0$, or $\delta$.

The top and bottom element of $\text{NST}$, denoted by $\top$ and $\bot$, then satisfy:

$(A x : N \times : (\top x = \{L,H\}) \land (\bot x = \emptyset))$

Let $\Gamma$ be a node-state and $x$ a circuit node. $\Gamma x$ can equal $\emptyset$, $\{L\}$, $\{H\}$, or $\{L,H\}$, in which cases we say that $x$ is floating, low, high, or conflicting respectively.

Node-states in a circuit $C$ with source-connection $\gamma$ are intuitively interpreted as follows:

1.4 Node-state $\Gamma$ is called a resulting state of $C$ and $\gamma$ if, for all $x \in N$:

$H \in \Gamma x \iff$ "as a result of $\gamma$ node $x$ is (via some conducting path) connected to an H-source"

$L \in \Gamma x \iff$ "as a result of $\gamma$ node $x$ is (via some conducting path) connected to an L-source"

1.5 Example

Consider again circuit $C_0$ with source-connection $\gamma_0$ from example 1.2. Since the gate of the n-switch is high, the switch will be conducting. As a result of this, the pass nodes $x$ and $z$ will be connected and therefore have the same state. Since $x$ and $z$ are connected only to an L-source, their state is $\{L\}$. The resulting node-state is: $\{(x,\{L\}), (y,\{H\}), (z,\{L\})\}$.  

Before we can analyse resulting states (informally described in 1.4 above) any further, we need to consider the behaviour of switches more closely. As a start, we define switch-states.

**Switch-states**

The state of a switch can be either conducting or nonconducting. Elements of \( SW \rightarrow B \) represent switch-states of circuits, where the range value '1' denotes that the switch is conducting, and '0' denotes that the switch is nonconducting. Typical names used for switch-states are \( Q \) and \( Q_1 \).

1.6 **Definition**

The set of switch-states \( SST \) and the (partial) order \( \leq \) on \( SST \) are defined by:

\[
SST = SW \rightarrow B
\]

\[
Q_0 \leq Q_1 = (A \ s : SW \cdot s : Q_0 \cdot s \Rightarrow Q_1 \cdot s), \text{ for } (Q_0, Q_1) \in SST
\]

The top and bottom element of \( SST \) are denoted by \( T \) and \( F \). They satisfy:

\[
(A \ s : SW \cdot s : T \cdot s \land \neg F \cdot s)
\]

**Remark**

A third value for switches can be considered, viz., in the case that the switch gate is floating or conflicting, and hence, the conductance state of the switch is not well-defined. A value denoting an 'unknown' state of the switch may seem appropriate, like, for instance, the 'X' in some of the models discussed in section 0.4 (like [Bry1], [Bry2], [BryS], [BrzS1], [BrzS2], and [Se]).

This, however, would lead – in our model – to a proliferation of states, both node-states and switch-states. This can be understood as follows. If a switch is in an 'unknown' state, the output of the switch (i.e. the non-driven pass node) will be in an 'unknown' state also (instead of \( \emptyset \), \( \{L\} \), \( \{H\} \), or \( \{L,H\} \)). But now one can distinguish different 'unknown' states for nodes, viz. depending on the state of the driving pass node. Making this distinction would lead to a further differentiation of switch-states.

In our model, we do not need such an 'unknown' state for switches. Instead, we will check switches on 'undefinedness' by considering their gate values (see 'gate definedness' in the next section) in all resulting states. To make sure we detect all possible malbehaviours (i.e. all possible incorrect resulting states) we allow both states of a switch if the gate is floating or conflicting.

\( \Box \) (end remark)
Circuit states

The node-state and the switch-state of a circuit affect each other. We therefore choose objects from $\text{SSTxNST}$ to represent circuit states. Henceforth, we will often call circuit states simply states. Typical names we use for states are $\Pi$ and $\Pi_1$.

1.7 Definition

The set of circuit states $\text{STO}$ and the (componentwise) order $\subseteq$ on $\text{STO}$ are defined by:

$$\text{STO} = \text{SSTxNST}$$

$$(Q_0, \Gamma_0) \subseteq (Q_1, \Gamma_1) = (Q_0 \subseteq Q_1 \land \Gamma_0 \subseteq \Gamma_1)$$

for $\{(Q_0, \Gamma_0), (Q_1, \Gamma_1)\} \subseteq \text{STO}$

1.8 Properties

a. $(\mathcal{P}(\{L, H\}), \subseteq)$ and $(\mathcal{B}, \leq)$ are complete lattices

b. $(\text{NST}, \subseteq)$ and $(\text{SST}, \subseteq)$ are complete lattices

c. $(\text{STO}, \subseteq)$ is a complete lattice

Property 1.8a is trivial, 1.8b follows from 1.8a, and 1.8c follows from 1.8b.

Remark

Most of the notions defined in this chapter are defined w.r.t. a specific circuit, say $C$. For instance, node-states and switch states are defined w.r.t. a specific circuit (cf. def. 1.3 and 1.6). The dependency of the notions on $C$ is not mentioned, except when there is a danger of confusion otherwise (in that case we add a subscript $C$).

1.2 Circuit behaviour

The state of a switch is well-defined if its gate node is either high or low. Namely, an $n$-switch is conducting if the gate is high, and nonconducting if the gate is low. A $p$-switch behaves in a converse fashion; it is conducting if the gate is low, and nonconducting if the gate is high. If, however, the gate is either floating or conflicting, then it is not clear whether the state of the switch is conducting or nonconducting. We shall show that, as a result of this, there are combinations $(C, \xi)$ for which we cannot adhere to the intuitive interpretation of resulting state as given above (remark 1.4). In example 1.9 below such a case is given.

If all switches are well-defined in all resulting states (which therefore adhere to 1.4), the combination $(C, \xi)$ is called well-matched. In the sequel we will define the notion well matched, and the resulting state(s) in case of well-matchedness.
We first investigate in what way a node-state affects a switch-state and vice versa. In order to do so, we will consider the case that the node-state is (possibly by magic) fixed, and investigate the possible resulting switch-state(s) (regardless of the effect such switch-state would have on the node-state). Similarly, we will consider the case that the switch-state is (possibly by magic) fixed, and investigate the possible resulting node-state(s). Later on, we will incorporate the results on these influences in circuit behaviour, and define stability of states, a notion of state transition, and oscillation of states. Here, the essence of restrictions (b) and (c), i.e., zero wire and pass-delay, and unit reaction-delay, will become clear. Finally, these notions are used to define well-matchedness and the resulting states.

1.9 Example

Circuit C_0 (ex. 1.1) with source-connection \( \gamma_1 = \{(x,(L),(y,0),(z,0))\} \) (figure 1.0a) is a case for which it is unclear what the resulting state is. Since the state of a switch with floating gate is not well-defined, the switch can be either nonconducting or conducting. As a result, the resulting node-state can be either \( \gamma_1 \) or \( \{(x,(L),(y,0),(z,(L)))\} \). Since the gate of the switch is floating, the switch will be called not gate-defined-0. Resulting states should have only gate-defined-0 switches.

\[
\begin{array}{ccc}
  x & \rightarrow & y \\
  \downarrow & \hookrightarrow & \downarrow \\
  L & \rightarrow & z
\end{array}
\]

\[
\begin{array}{ccc}
  x & \rightarrow & y \\
  \downarrow & \hookrightarrow & \downarrow \\
  L & \rightarrow & z
\end{array}
\]

figure 1.0a: ex. 1.9
figure 1.0b: ex. 1.10

Remark

The numbering of the notion gate-defined (with -0) results from the fact that this notion will be refined in later chapters. The same remark holds for the notions stable0, feasible0, and WM0 defined later on.

The following example informally introduces two consistency notions for states, and a stability notion for states.

1.10 Example

Consider circuit C_0 with source-connection \( \gamma_0 \) from example 1.2; see figure 1.0b.

First consider a state \((Q_0) (\Gamma_0)\) with \(-Q_0 \wedge (\Gamma_0 \neq \{H\})\), that is, a nonconducting n-switch with high gate. Since an n-switch with high gate is supposed to be (come) conducting, such \( Q_0 \) and \( \Gamma_0 \) are called inconsistent. Now consider a state \((Q_1) (\Gamma_1)\) with \(Q_1 \wedge (\Gamma_1 \neq \{L\})\), that is, a conducting switch with pass nodes that have different states. Since a conducting switch is supposed to connect its pass nodes, and hence, they should have the same state, such \( Q_1 \) and \( \Gamma_1 \) are also called inconsistent.
The two examples of inconsistency are, however, essentially different. In the first case (Q_0 and \Gamma_0) the inconsistency is caused by the influence of the node state on the switch-state, i.e., the node state demands a change in the switch-state. In the second case the inconsistency is caused by the influence of the switch-state on the node state, i.e., the switch-state demands a change in the node-state. These types of inconsistency are called inconsistent-0 and inconsistent-1 respectively. If the switch-state and the node-state are consistent (both types simultaneously) and the node-state corresponds to the given source-connection, the state is called stable. As explained in example 1.5 the current example has only one stable state, viz. \(((s,1),(s,(L)),(y,(H)),(z,(L)))\)

\(\square\) (end example 1.10)

**Fixed node-state**

We will analyse, for a fixed node-state \(\Gamma\), which switch-states can result from \(\Gamma\). The state of a switch depends only on its type and on the state of its gate node, and not, for instance, on the states of its pass nodes. As mentioned above the state of a switch, say \(s\), is well-defined if its gate is low or high, viz. by \(Q\cdot s = (t\cdot s = \Gamma\cdot(g\cdot s))\). A switch with low or high gate is called gate defined-0. A circuit state with all switches gate defined-0 is called completely gate defined-0.

If the state of a switch corresponds in the above described way with the state of its gate node, it is called consistent-0. That is, if the gate of a switch \(s\) is high or low, then \(s\) is consistent-0 only if \(Q\cdot s = (t\cdot s = \Gamma\cdot(g\cdot s))\). But if the gate of a switch \(s\) is floating or conflicting, then \(s\) is consistent-0 regardless of the value of \(Q\cdot s\). If all switches are consistent 0, the switch state is called completely consistent-0.

1.11 Definition

\[ g_{d0} \in \text{NST} \rightarrow \text{SW} \rightarrow \mathbb{B} \quad \text{and} \quad c_{g_{d0}} \in \text{ST0} \rightarrow \mathbb{B} \quad \text{are defined by:} \]

\[ g_{d0}\cdot \Gamma\cdot \cdot s = (s\cdot (g\cdot s) \in \{L\}, \{H\}) \]

\[ c_{g_{d0}} \cdot (Q, \Gamma) = (A \cdot s : SW\cdot s : g_{d0}\cdot \Gamma\cdot \cdot s) \]

\((c)g_{d0}\) stands for (completely) gate defined 0.

1.12 Definition

\[ \text{consistent-0} \in \text{NST} \rightarrow \text{SST} \rightarrow \text{SW} \rightarrow \mathbb{B} \quad \text{and} \quad \text{coco-0} \in \text{NST} \rightarrow \text{SST} \rightarrow \mathbb{B} \quad \text{are defined by:} \]

\[ \\text{consistent-0} \cdot \Gamma\cdot Q\cdot s = (g_{d0}\cdot \Gamma\cdot \cdot s \cdot (Q\cdot s = (t\cdot s = \Gamma\cdot(g\cdot s))) \]

\[ \text{coco-0} \cdot \Gamma\cdot Q = (A \cdot s : SW\cdot s : \text{consistent-0} \cdot \Gamma\cdot Q\cdot s) \]

\text{consistent-0} stands for consistent-0, and coco-0 for completely consistent-0.

The completely consistent-0 switch-states can be regarded as results of the fixed node-state \(\Gamma\).

The reason for constraining switch behaviour only if the switch is gate defined 0, is that we want to detect all possible resulting states.
In 1.13 below some properties of the above defined notions are given. Notice that if each gate is either high or low in a specific node-state, only one switch-state corresponds to this node-state (1.13c).

1.13 Properties
a. \( \neg g_{do}\Gamma; s \circ (A Q : SST\cdot Q : consistent^0\cdot \Gamma\cdot Q; s) \) , for NST-\( \Gamma \wedge SW\cdot s \)
b. \( \#(coco0\cdot \Gamma) = 2\#(SW\setminus g do\cdot \Gamma) \) , for NST-\( \Gamma \)
c. \( (g do\cdot \Gamma = SW) = (\#(coco0\cdot \Gamma) = 1) \) , for NST-\( \Gamma \)

Since property 1.13a is a direct consequence of definition 1.12, and property 1.13c follows immediately from 1.13b, we only prove 1.13b.

Proof of 1.13b

Let \( \Gamma \) be a node-state. Then:

\[ \#(coco0\cdot \Gamma) \]

= (definition 1.12)

\[ \#(Q | SST\cdot Q \wedge (A s : SW\cdot s : consistent^0\cdot \Gamma\cdot Q\cdot s)) \]

= (definition 1.12)

\[ \#(Q | SST\cdot Q \wedge (A s : SW\cdot s : g do\cdot \Gamma\cdot s \wedge (Q\cdot s = (t\cdot s = \Gamma\cdot (g\cdot s)))) ) \]

= (calculus, using the definition of SST (1.6))

\[ \#(Q | SST\cdot Q \wedge (A s : g do\cdot \Gamma\cdot s : Q\cdot s = (t\cdot s = \Gamma\cdot (g\cdot s)) \wedge (A s : \neg g do\cdot \Gamma\cdot s : Q\cdot s \in \emptyset)) \]

= (calculus, \#B = 2, and \( \#(g do\cdot \Gamma) = \#(SW\setminus g do\cdot \Gamma) \})

\[ 1\#(g do\cdot \Gamma) \times 2\#(SW\setminus g do\cdot \Gamma) \]

\[ = (calculus) \]

\[ 2\#(SW\setminus g do\cdot \Gamma) \]

\[ \square \]

Resulting switch-state

As argued above, the switch-states that correspond to a node-state \( \Gamma \) are those that are completely consistent 0 with \( \Gamma \). In other words, a switch-state \( Q \) corresponds to a node-state \( \Gamma \) if \( coco0\cdot \Gamma\cdot Q \)

\[ (0) \]

Fixed switch-state

Let \( C = (N,SW,t,g,p) \) be a circuit and \( \gamma \) a source-connection. We will analyse, for a fixed switch-state \( Q \) , which node-states can be resulting from \( Q \). Let \( \Gamma \) be such a resulting node-state of \( C \), \( \gamma \), and \( Q \). Recall that \( \Gamma \) must adhere to the intuitive interpretation given in 1.4. To be able to formalise 1.4 we define when nodes are connected via a conducting path. Each switch, when conducting, establishes a basic conducting path between its pass nodes. The relation conducting path is the reflexive and transitive closure (defined in appendix A) of the
1.14 Definition

\[ \text{bcp} \text{ and } \text{cp}, \text{ both of type } \text{SST} \rightarrow \text{B2N} \rightarrow \text{B} \], are defined by:

\[ \text{bcp-Q} : b = (E : s : SW_s \land Q : s : b = pn_s ) \]

\[ \text{cp-Q} \text{ is the reflexive and transitive closure of } \text{bcp-Q} \]

(b)cp stands for (basic) conducting path.

Notice that nodes \( x \) and \( y \) are, as a result of \( Q \), connected via a conducting path if and only if \( \text{cp-Q} : [x,y] \).

With the help of 1.4 we will analyse the properties that \( \Gamma \) must satisfy in order to be a resulting state of \( C, \gamma \), and \( Q \). Firstly, as a result of \( \gamma \), all nodes \( x \) with \( H \in \gamma x \) are connected to an H-source, so \( H \in \Gamma x \) must hold (similar for L). This is concisely expressed as \( \gamma \subseteq \Gamma \); condition (1) below. Furthermore, if a switch is conducting, its pass nodes are connected and will therefore have the same state (condition (2) below; recall that switches are assumed to be perfect). Condition (2) is called consistent-1, and is defined similarly to consistent-0 in definition 1.15 below.

\[ \gamma \subseteq \Gamma \]

\[ (A_{x,y} : \text{bcp-Q} : [x,y] : \Gamma x = \Gamma y ) \]

(1)

(2)

1.15 Definition

\text{consistent-1} \in \text{NST} \rightarrow \text{SST} \rightarrow \text{SW} \rightarrow \text{B} \text{ and } \text{cocol} \in \text{NST} \rightarrow \text{SST} \rightarrow \text{B} \text{ are defined by:}

\[ \text{consistent-1} : \Gamma Q : s = (Q : s = (A_{x,y} : pn_s = [x,y] : \Gamma x = \Gamma y ) ) \]

\[ \text{cocol} : \Gamma Q = (A : s : SW_s : \text{consistent-1} : \Gamma Q : s ) \]

\[ \text{consistent-1} \text{ stands for consistent-1, and cocol for completely consistent-1.} \]

Notice that \( \text{cocol} : \Gamma Q = (A_{x,y} : \text{bcp-Q} : [x,y] : \Gamma x = \Gamma y ) \), that is, \( \text{cocol} : \Gamma Q \) equals (2).

The following example shows that conditions (1) and (2) are not sufficient to characterise resulting states.

1.16 Example

Consider circuit \( \langle \{x\} : \emptyset, \emptyset, \emptyset, \emptyset \rangle \) consisting of one node only, with source-connection \( \gamma \) defined by \( \gamma x = \emptyset \). Now all node-states of this circuit satisfy conditions (1) and (2), that is, \( x \) can have state \( \emptyset, \{L\}, \{H\}, \text{ or } \{L,H\} \). Notice that only the first state (with \( \Gamma x = \emptyset \)) adheres to the intuitive interpretation given in 1.4. The other states suggest a miraculous connection of \( x \) to absent sources.
Similar examples can be given for circuits that do contain switches, e.g., for \( C_0 \) with \( \gamma_0 \) (ex. 1.2) and \( Q \) such that \( Q \cdot \gamma \), four node-states exist that satisfy (1) and (2), but only one of them adheres to 1.4.

Clearly, a node \( x \) is connected to an \( H \)-source if and only if a conducting path between \( x \) and an \( H \)-source (i.e., a node \( y \) with \( H \in \gamma y \)) exists (similar for \( L \)). In other words: the state of a node equals the union of its reachable sources, that is, for resulting state \( \Gamma \):

\[
(\text{A} \cdot x : \text{N} \cdot x : \Gamma : x = (u \cdot y : \text{cp-Q} \cdot \{x, y\} : \gamma y ))
\]

(3)

Notice that, on account of (3), the node-state resulting from the fixed switch-state \( Q \) is unique. In fact, this resulting state is the smallest one satisfying (1) and (2). Notice also that, on account of reflexivity and transitivity of \( \text{cp-Q} \) (def. 1.14), condition (3) implies conditions (1) and (2).

The resulting node-state for a fixed switch-state \( Q \) can be nicely expressed with the following function.

1.17 Definition

The response function \( R \in \text{NST} \rightarrow \text{SST} \rightarrow \text{NST} \) is defined as:

\[
R \cdot \gamma \cdot Q \cdot x = (u \cdot y : \text{cp-Q} \cdot \{x, y\} : \gamma y )
\]

Resulting node-state

Conditions (1), (2), and (3) can now be reformulated as: For a fixed switch-state \( Q \) and a source-connection \( \gamma \), the resulting node-state is \( R \cdot \gamma \cdot Q \).

Our earlier observation "(3) \( \Rightarrow (1) \wedge (2) \)" is formalised in property 1.18a below.

1.18 Properties

\( (\gamma \in R \cdot \gamma \cdot Q) \wedge \text{cocon-}(R \cdot \gamma \cdot Q) \cdot Q \)

b \( \text{bcp and cp are monotonic with respect to the order} \leq \text{on the domain (SST)} \) and the order \( \subseteq \text{on their codomain (AB2N)} \)

c \( R \cdot \gamma \text{ is monotonic with respect to the order} \leq \text{on the domain (SST)} \) and the order \( \subseteq \text{on its codomain (NST)} \).

Proof of 1.18b,c

Let \( Q_0 \) and \( Q_1 \) be given such that \( \{Q_0, Q_1\} \subseteq \text{SST} \) and \( Q_0 \leq Q_1 \). It is easily verified (using def. 1.14) that \( \text{bcp-Q}_0 \subseteq \text{bcp-Q}_1 \). As a result: \( \text{cp-Q}_0 \subseteq \text{cp-Q}_1 \). Using the definition of \( R \cdot \gamma \) (1.17) it is easily seen that \( R \cdot \gamma \cdot Q_0 \subseteq R \cdot \gamma \cdot Q_1 \).

\( \Box \)
**Resulting states**

We will now combine the results from the (magical) cases with fixed node-state and fixed switch-state respectively. For a circuit with a specific source-connection we want to determine the possible resulting states. Resulting states are those states the circuit can either remain in, or can return to infinitely often. The states the circuit can remain in are called stable. The states the circuit can return to infinitely often are called oscillating. In order to define oscillations, some notion of state transition is required. For the latter notion the restrictions (b) and (c) on delays will be used.

Stability of states can be defined easily; a state is stable if the switch-state and the node-state match according to (0) and (3) above.

**1.19 Definition**

\[ \text{stable} \in \text{NST} \cap \text{STO} \cap \text{E} \] is defined by:

\[ \text{stable} \in \gamma(Q, \Gamma) = (c_0 \in \Gamma \cap Q \land \Gamma = R \cdot \gamma(Q)) \]

Notice that this is a weak notion of stability in the sense that, although the node-state and the switch-state match (stability is possibly), there may be more than one switch state that matches the node-state (stability is not certain), viz. if the node-state is not completely gate defined-0. Notice also that, on account of property 1.13a, if the node-state is completely gate defined-0, then only one switch-state matches the node-state (stability is certain), and, hence, the notion of stability is strong for “correct” states.

**1.20 Example**

a) Consider again circuit \( C_0 \) with source-connection \( \gamma_0 \) (ex. 1.2 and 1.5). Regardless of the conductance state of switch \( s \), the two states where \( x, y, z \) have values \{L\}, \{H\}, \emptyset respectively, are not stable-0. Namely, if the state of \( s \) is 0 then \( \sim \text{consistent}0 \cap Q \cdot s \); and if the state of \( s \) is 1 then \( \sim \text{consistent}1 \cap Q \cdot s \), and hence (property 1.18a) \( \Gamma = R \cdot \gamma \cdot Q \). The only stable state of \( C_0 \) w.r.t. \( \gamma_0 \) is the state where \( x, y, z \) have states \{L\}, \{H\}, and \{L\} respectively.

b) It is possible for a circuit to have more than one stable state w.r.t. a specific source connection. Consider, for instance, circuit \( C_1 = (\{x, y\}, \{s, \{H\}\}, \{(s, \{H\})\}, \{(x, y)\}, \{(s, [y])\}) \) with source-connection \( \gamma = (\{x, \{H\}\}, \{y, \emptyset\}) \), as depicted in figure 1.1a. Now two states are stable: \( (\{s, 0\}, \gamma) \) and \( (\{s, 1\}, \{(x, \{H\}), (y, \{H\})\}) \). Notice that only the latter is completely gate defined-0.
A circuit can have more than one completely gate defined-0 stable state. Consider circuit $C_2$ depicted in figure 1.1b with source connection $y_2 = \{(x, \{H\}),(y, \emptyset),(z, \{L\})\}$. Circuit $C_2$ has four stable states w.r.t. $y_2$, viz. the states where $s_0, s_1, x, y, z$ have the values:

$1, 0, \{H\}, \{H\}, \{L\}$; $0, 1, \{H\}, \{L\}, \{L\}$; $0, 0, \{H\}, \emptyset, \{L\}$; and $1, 1, \{L,H\}, \{L,H\}$, respectively. Notice that only the first two are completely gate defined-0.

Consider circuit $C_2$ with source-connection $y_3 = \{(x, \{L\}),(y, \emptyset),(z, \{H\})\}$, as depicted in figure 1.1c. It is an inverter with the output fed back to the input. Now consider the states $\Pi_0$ and $\Pi_1$ defined as (with the same notation as above) $0, 1, \{L\}, \{H\}, \{H\}$ and $1, 0, \{L\}, \{L\}, \{H\}$ respectively. Both states satisfy (3) but do not satisfy (0). According to (0) the high value of $y$ in state $\Pi_0$ would lead to values 1 and 0 for $s_0$ and $s_1$ respectively. Now using (3) we observe that state $\Pi_0$ leads to state $\Pi_1$. Similarly, state $\Pi_1$ leads to $\Pi_0$. Such a cycle of states is called oscillating (where a similar remark as for stability, on weakness and strongness, is applicable).

Besides stable states we are interested in oscillating states (see example 1.20d above), since they are also possible resulting states. In order to define oscillations, we need to consider possible state transitions. Now assuming that wire and pass-delays are zero (restriction (b)), all states ($Q, \Gamma$) will satisfy $\Gamma = R - \gamma - Q$. We therefore restrict ourselves to those elements of STO that satisfy (3). Assuming a unit, positive, and finite reaction-delay (restriction (c)) all possible next states ($Q_n, \Gamma_n$) of a state ($Q_0, \Gamma_0$) will satisfy $\text{ coco0} - \Gamma_0 - Q_1$. Notice that restrictions (b) and (c) lead to a nice next-state function, called $\text{ next0} - \gamma$ (def. 1.21 below).

For a circuit state $\Pi$, $\text{ next0} - \gamma - \Pi$ is the set of possible next states.

### 1.21 Definition

$\text{next0} \in \text{NST} \rightarrow \text{STO} \rightarrow \mathcal{K}(\text{STO})$ is defined by:

$\text{next0} - \gamma (Q, \Gamma) = \{ Q_1 : \text{ coco0} - \Gamma - Q_1 : (Q_1, R - \gamma - Q_1) \}$

In following chapters variation of delays will lead to a revision of the next-state function.

In properties 1.22a,b,c alternative definitions for $\text{next0}$ and $\text{stable0}$ are given. Property 1.22e expresses that for completely gate defined-0 states the next state is unique.
1.22 Properties

a $\text{next}_{\cdot \gamma} \cdot (Q_0, \Gamma_0) \cdot (Q_2, \Gamma_2) = (\text{coco}_{\cdot \gamma} \cdot Q_1 \land (\Gamma_1 = R \cdot \gamma \cdot Q_1))$

b $\text{stable}_{\cdot \gamma} \cdot \Pi = \text{next}_{\cdot \gamma} \cdot \Pi \cdot \Pi$

c $\text{coco}_{\cdot \gamma} \cdot (R \cdot \gamma \cdot Q) \cdot \Pi = \text{stable}_{\cdot \gamma} \cdot \gamma (Q, R \cdot \gamma \cdot Q)$

d $\# (\text{next}_{\cdot \gamma} \cdot (Q, \Gamma)) = 2^\# (\Sigma \cup \delta_{\cdot \gamma})$

e $\text{cog}_{\cdot \gamma} \cdot \Pi = (\# (\text{next}_{\cdot \gamma} \cdot \gamma \cdot \Pi) = 1)$

Properties 1.22a,b follow directly from definitions 1.21 and 1.19. Property c follows from a and b. Properties 1.22d,e follow directly from definition 1.21 and properties 1.13b and 1.13c.

The transitive closure $(\text{next}_{\cdot \gamma})^+ \cdot \gamma$ is defined in the usual way, viz. $(\text{next}_{\cdot \gamma})^+ \cdot \Pi_0$ is the smallest set $M$ satisfying $(\text{next}_{\cdot \gamma} \cdot \Pi_0 \subseteq M) \land (A \cdot \Pi : \Pi \in M : \text{next}_{\cdot \gamma} \cdot \Pi \subseteq M)$.

The set of all stable and all oscillating states is called $\text{feasible}_{\cdot \gamma}$.

1.23 Definition

$\text{feasible}_{\cdot \gamma} \in \text{NST} - \text{ST0} - \text{B}$ is defined by:

$\text{feasible}_{\cdot \gamma} \cdot \Pi = (\text{next}_{\cdot \gamma})^+ \cdot \Pi \cdot \Pi$

1.24 Property

For all $\gamma : \text{stable}_{\cdot \gamma} \cdot \Pi \subseteq \text{feasible}_{\cdot \gamma}$

Proof

$\text{stable}_{\cdot \gamma} \cdot \Pi$

$\Rightarrow (\text{next}_{\cdot \gamma} \cdot \Pi \cdot \Pi)$

$(\text{next}_{\cdot \gamma})^+ \cdot \Pi \cdot \Pi$

$\Rightarrow (\text{def. } (\text{next}_{\cdot \gamma})^+)$

$\Rightarrow (\text{def. 1.23})$

$\text{feasible}_{\cdot \gamma} \cdot \Pi$

$\square$

Well-matchedness

At the start of this section, we observed that for some combinations $(C, \gamma)$ it is impossible to define the resulting states in such a way that they adhere to the intuitive interpretation given in remark 1.4 (recall example 1.9). This is a result of the possible occurrence of floating or conflicting gates in resulting states, and the undefinedness of switches this causes. We have overcome part of this problem by allowing all possible behaviours for gate-undefined switches. In this way we are able to observe all possible resulting states within the model, also those that
denote possible malbehaviour of the circuit. A consequence of this is, however, that some of the resulting states within the model, i.e., some of the feasible states, need not correspond to physical behaviour (as resulting states) at all, namely, if they contain gate-defined switches.

Our first correctness criterion, expressed in the notion well-matched defined below, therefore requires that all feasible states contain only gate-defined switches.

This requirement may be too severe, namely, for some combinations (C,G) it is clear what the resulting node-state is, although it is not clear, due to gate-undefined switches, what the resulting switch-state is (see ex. 1.26a below). We feel, however, that correctly designed circuits should not contain undefined switches in any possible resulting state. This explains the – possibly too severe – correctness criterion mentioned above, and the pessimism of the model resulting from it.

1.25 Definition

\[ WM0 \in \text{CSR} \rightarrow \text{NST} \rightarrow \mathbb{B} \] is defined by:

\[ WM0 \cdot C \cdot G = ( \text{feasible} \cdot G \subseteq \text{cgd0} ) \]

\[ WM0 \] stands for well-matched-0.

Well-matchedness expresses the possibility to predict (within the model) the resulting state(s) for a circuit with a specific source-connection. In case of well-matchedness, the set of possible resulting states is feasible-\( G \).

1.26 Example

a) Consider circuit \( C_0 \) from example 1.1 with source-connection \( \gamma_4 = \{ (x,\emptyset), (y,\emptyset), (z,\emptyset) \} \).

Now feasible-\( G \) contains two states, viz. \( \{ (x,0), \gamma_4 \} \) and \( \{ (x,1), \gamma_4 \} \). Since these states are not completely gate-defined, \( C_0 \) and \( \gamma_4 \) are not well-matched.

b) Consider again the combinations circuit – source-connection given in example 1.20a–d. For \( C_0, \gamma_0 \) (1.20a) only one feasible state exists, viz. the stable state mentioned in 1.20a. Since this state is completely gate defined-0, \( WM0 \cdot C_0 \cdot \gamma_0 \) holds.

Both \( C_1, \gamma \) and \( C_2, \gamma_2 \) (1.20a,b) have stable states that are not completely gate defined-0 (see 1.20a,b). As a result (use prop. 1.24) \( \neg WM0 \cdot C_1 \cdot \gamma \land \neg WM0 \cdot C_2 \cdot \gamma_2 \).

The fourth combination \( (C_2, \gamma_2) \text{ from 1.20d)} \) is also not well-matched. Namely, state \( (T,T) \) is stable (and hence feasible), but not completely gate defined-0.

Two additional correctness criteria will be formulated later on (section 1.4).

Except for these additional correctness criteria the modelling is now complete. The remainder of this chapter contains (mathematical) consequences of the definitions.
The evaluation of $WM0$ by its definition is rather complex; the set $feasible\cdot\gamma$ cannot be calculated efficiently. Namely, the number of switch states is $z^{\delta}\cdot SW$, for each state $(Q,R,\gamma;Q)$ the successors (i.e. $next\bar{\gamma}(Q,R,\gamma;Q)$) must be calculated, and, finally, the feasible states must be selected. In order to find equivalent expressions for $WM0$ that are more convenient, we will investigate $WM0$ more closely.

On account of property 1.24 it is easily seen that:

1.27 Property

$$(A, \gamma; CIR\cdot C \land NST\cdot \gamma; WM0\cdot C \cdot \gamma \Rightarrow (stable\cdot \gamma \subseteq cgdo))$$

Proof

Let $C$ and $\gamma$ be such that $CIR\cdot C \land NST\cdot \gamma$. Then:

$$WM0\cdot C \cdot \gamma$$

$$= (\text{def. 1.25})$$

$$feasible\cdot \gamma \subseteq cgdo$$

$$\Rightarrow (\text{prop. 1.24})$$

$$stable\cdot \gamma \subseteq cgdo$$

$\Box$

Good mathematical practice now leads to the question whether or not the converse of the above property holds. Since it is difficult to get a grip on the notions $feasible\cdot \gamma$, $stable\cdot \gamma$, and $cgdo$, it is not intuitively obvious whether or not this converse property holds.

The converse of property 1.27 turns out to be true (theorem 1.28 below). Theorem 1.28 requires a lengthy proof, for which reason we have banished it to appendix B0.

1.28 Theorem

$$(A, \gamma; CIR\cdot C \land NST\cdot \gamma; WM0\cdot C \cdot \gamma = (stable\cdot \gamma \subseteq cgdo))$$

The complete proof of 1.28 is given in appendix B0. In an attempt to give some intuition for theorem 1.28, we give a global outline of the proof here.

The proof given in appendix B0 is a constructive one: For a cycle of $m$, with $1 \leq m$, feasible states $(Q_i; i)$, $0 \leq i < m$, satisfying:

$$(A, i; 0 \leq i < m : next\bar{\gamma}(Q_i; i) \Rightarrow (Q_{i+1}; i'))$$

with $(Q_m; i_m) = (Q_0; i_0)$,

a stable state $(\Delta; Q; I)$ is constructed that satisfies, for all $s \in SW$:

$$(B, i; 0 \leq i < m : \neg gd\cdot \gamma; i; x) \Rightarrow \neg gd\cdot \Delta; s$$

This means that a switch is gate defined-0 in the constructed stable state only if it is gate defined-0 in all feasible states of the cycle. Since for all feasible states such a cycle exists (use def. 1.23), the property follows from this construction and the def. of $WM0$. 

Notice that 1.28 is a powerful theorem. The evaluation of WM0 by this theorem is much easier than by its definition (see above), since calculating the set \textit{\textit{stable}o\textit{\gamma}} is much easier than calculating \textit{\textit{feasible}o\textit{\gamma}}.

1.3 Properties of WM0

Consider a state transition from \((Q_0, \Gamma_0)\) to \((Q_1, \Gamma_1)\), that is, \textit{nexto}\textit{\gamma}(Q_0, \Gamma_0) = (Q_1, \Gamma_1) holds. Notice that \(Q_i\) depends only on \(\Gamma_0\) and \(\Gamma_j\) depends only on \(Q_i\) (cf. def. \textit{nexto}\textit{\gamma}). State transitions can therefore be characterized by elements of the set of switch functions \(\textit{SF} \subseteq \textit{NST} \rightarrow \textit{SST}\), defined by:

\[
\textit{SF} = \{ G \mid (G \in \textit{SST}) \land (\Lambda \Gamma : \textit{NST} \rightarrow \textit{SST} : \textit{coco}o\textit{\gamma}(G \rightarrow \Gamma)) \}.
\]

State transitions can be characterized by: \textit{nexto}\textit{\gamma}(Q, \Gamma) = \{ G : \textit{SF} \rightarrow \textit{SST} : (G \rightarrow \Gamma, R \rightarrow \gamma(G \rightarrow \Gamma)) \}.

Since we are mainly interested in the resulting node-state(s), we could eliminate the switch-states from our definitions, viz., by using switch functions and consider the "excitation functions" \(R \rightarrow \gamma G\) (which are functions from \(\textit{NST} \rightarrow \textit{NST}\)). The separation of switch-states and node-states is, however, necessary to formalise delays, in particular reaction-delays (chapter 3).

The switch functions can, however, be helpful in deriving properties of the notions we want to calculate. Two of them, called \textit{G0} and \textit{G1}, are particularly interesting, since they are monotonic and anti-monotonically respectively. In this section we will investigate these two functions.

The monotonic switch function \textit{G0}

Let \(G \in \textit{SF}, \Gamma \in \textit{NST}\), and \(s \in \textit{SW}\). Since \(G\) is a switch function, \(\Gamma\) and \(G \rightarrow \Gamma\) are consistent: 0 in \(s\), that is, \(\textit{gd}o\textit{\gamma}(\Gamma \rightarrow s) \circ (G \rightarrow \Gamma \rightarrow s) = (t \in s = \textit{gd} \rightarrow \Gamma \rightarrow s)\) holds. Now consider node-states \(\Gamma_0, \Gamma_1, \Gamma_2, \text{ and } \Gamma_3\) that are equal to \(\Gamma\) in all nodes \(y\) with \(y \neq g \rightarrow s\), and have values \(\emptyset, \{ L \}, \{ H \}\), and \(\{ L, H \}\) in \(g \rightarrow s\) respectively. Notice that \(G \rightarrow \Gamma_1 \neq G \rightarrow \Gamma_2 \rightarrow s\). Since \((\Gamma_1 \leq \Gamma_2) \land (\Gamma_2 \leq \Gamma_3)\), \(G\) must, in order to be monotonic, satisfy: \(G \rightarrow \Gamma_1 \leq G \rightarrow \Gamma_2 \land (G \rightarrow \Gamma_2 \leq G \rightarrow \Gamma_3)\), and hence, since either \(G \rightarrow \Gamma_1\) or \(G \rightarrow \Gamma_2\) is 1 in \(s\), \(G \rightarrow \Gamma_2 \rightarrow s = 1\). Similarly, in order to be monotonic, \(G\) must satisfy: \(G \rightarrow \Gamma_0 \rightarrow s = 0\).

Since \(\Gamma \in \{ \Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3\}\), this means that \(G\) must satisfy, in order to be monotonic:

\[\textit{gd}o\textit{\gamma}(\Gamma \rightarrow s) \circ (G \rightarrow \Gamma \rightarrow s) = (t \in s \subseteq \Gamma \rightarrow (g \rightarrow s))\].

Notice that this means that only one switch function is monotonic, and that it can be defined as follows:

1.29 Definition

\(G \in \textit{NST} \rightarrow \textit{SST}\) is defined by:

\(G \rightarrow \Gamma \rightarrow s = (t \in s \subseteq \Gamma \rightarrow (g \rightarrow s))\)
1.30 Properties

a  \((A \Gamma : \text{NST-}\Gamma : \text{cooco-}\Gamma : (G0-\Gamma ))\)

b  \(G0\) is monotonic w.r.t. order \(\leq\) on its domain and order \(\leq\) on its codomain.

c  For all \(\gamma \), \(\text{NST-}\gamma \): \(R-\gamma;G0\) is monotonic w.r.t. \(\leq\) on its domain and codomain.

Property 1.30a follows from definitions 1.29 (G0) and 1.12 (cooco). Property 1.30b follows from definitions 1.3 (\(\leq\)), 1.6 (\(\leq\)), and 1.29 (G0). Property 1.30c follows immediately from properties 1.18c (monotonicity of \(R-\gamma\)) and 1.30b.

1.31 Theorem

\(R-\gamma;G0\) has a least and a greatest fixpoint, which can be calculated by successive approximation from below and by successive approximation from above respectively.

Proof

The proof consists of the following steps:

i) \((\text{NST,}\leq)\) is a complete lattice (property 1.8b)

ii) \(R-\gamma;G0\) is monotonic (property 1.30c)

iii) The existence of a least and a greatest fixpoint follows from the theorem of Knaster-Tarski (appendix A, th. A2) using i) and ii)

iv) \(R-\gamma;G0\) is continuous (NST is finite and ii))

v) With the fixpoint theorem for continuous functions (appendix A; th. A3) it follows that the least and the greatest fixpoint can be calculated with successive approximation from below and above respectively.

Since NST is finite, the least and the greatest fixpoint of \(R-\gamma;G0\) can be calculated in a finite number of steps. The circuit states formed by these fixpoints are called \(\gamma^*\) and \(\gamma^*\) (def. 1.32). Since they are formed with fixpoints of \(R-\gamma;G0\), they are stable (property 1.34a).

1.32 Definition

Circuit states \(\gamma \in \text{ST0}\) and \(\gamma^* \in \text{ST0}\) are defined by:

\[\gamma = (G0-(\text{lf}(R-\gamma;G0)))\] \[\gamma^* = (G0-(\text{gfp}(R-\gamma;G0)))\]

where \(\text{lf}\) is the least-fixpoint function, and \(\text{gfp}\) is the greatest-fixpoint function.

We can now define the next states and the feasible states w.r.t. G0.
1.33 Definition

next\textsubscript{0}\gamma \in \text{NST} \setminus \text{STO} \rightarrow \gamma(\text{STO}) and feasible\textsubscript{0}\gamma \in \text{NST} \setminus \text{STO} \rightarrow \mathcal{B} are defined by:

\begin{align*}
\text{next}\textsubscript{0}\gamma(Q,\Gamma) &= \{(\text{G0-}\Gamma, R - \gamma(\text{G0-}\Gamma))\} \\
\text{feasible}\textsubscript{0}\gamma(\gamma) &= \text{next}\textsubscript{0}\gamma(\gamma) + \Pi \cdot \Gamma
\end{align*}

1.34 Properties

\begin{enumerate}
\item \quad \text{stable}\gamma(\gamma) \wedge \text{stable}\gamma(\gamma*)
\item \quad (\text{next}\textsubscript{0}\gamma(\gamma) = \{Q\}) \wedge (\text{next}\textsubscript{0}\gamma(\gamma*) = \{Q\})
\item \quad \text{feasible}\textsubscript{0}\gamma(\gamma) \wedge \text{feasible}\textsubscript{0}\gamma(\gamma*)
\item \quad \text{feasible}\textsubscript{0}\gamma \subseteq \text{feasible}\gamma
\end{enumerate}

Properties 1.34a,b,c follow immediately from definitions 1.32 and 1.33. Property 1.34d follows immediately from property 1.30a and definition 1.33.

Since \gamma and \gamma* can be calculated efficiently, we would like to use them to calculate WM\textsubscript{0}-C-\gamma. In order to find out whether this is possible, we investigate \gamma and \gamma* further.

1.35 Lemma

\( (\gamma = \gamma^*) \iff (\text{feasible}\textsubscript{0}\gamma = \{Q\}) \)

Proof

From property 1.34c follows: \( (\text{feasible}\textsubscript{0}\gamma = \{Q\}) \Rightarrow (\gamma = \gamma^*) \). So we only need to prove the implication to the right.

Assume that \( \gamma = \gamma^* \) and \( \text{feasible}\textsubscript{0}\gamma(\text{Q,}\Gamma) \).

Let \( f_0\gamma = R - \gamma - \text{G0} \), \( \gamma = (Q_0, \Gamma_0) \), and \( \gamma^* = (Q_0, \Gamma_0) \). Choose \( n, 1 \leq n \), such that:

\( (\text{next}\textsubscript{0}\gamma)^n(Q, \Gamma) = (Q, \Gamma) \wedge ((f_0\gamma)^n - \Gamma = \Gamma_0) \wedge ((f_0\gamma)^n - \Gamma = \Gamma_0) \).

Such \( n \) exists on account of finiteness of NST and def. 1.32 and 1.33. Then:

\text{true}

\[ (\text{definition of } L \text{ and } T) \]

\[ (L \subseteq \Gamma) \wedge (\Gamma \subseteq T) \]

\[ (\text{monotonicity of } f_0\gamma (\text{prop. 1.30c})) \]

\[ ((f_0\gamma)^n - L \subseteq (f_0\gamma)^n - \Gamma) \wedge ((f_0\gamma)^n - \Gamma \subseteq (f_0\gamma)^n - T) \]

\[ (\text{specification of } n, \text{ def. next}\textsubscript{0}\gamma (1.33)) \]

\[ (\Gamma_0 \subseteq \Gamma) \wedge (\Gamma \subseteq \Gamma_0) \]

\[ (\text{from } \gamma = \gamma^*, \Gamma_0 = \Gamma_0 \text{)} \]

\[ \Gamma_0 = \Gamma \]

\[ (\text{specification of } n, \gamma \text{ is a fixpoint, and def. next}\textsubscript{0}\gamma (1.33)) \]

\[ \gamma = (Q, \Gamma) \]

\[ \square \]
1.36 Lemma

\((A, C, \gamma: CIR-C \land NST-\gamma: WMO0-C-\gamma \Rightarrow \gamma_s = \gamma^*)\)

Proof

Let \(CIR-C, NST-\gamma, \gamma_s = (Q_{\text{ho}}, \Gamma_{10})\), and \(\gamma^* = (Q_{\text{ho}}, \Gamma_{\gamma0})\). Then:

\[ WMO0-C-\gamma \]

= [def. \(WMO0\) (1.25)]

\[ \text{feasible}6-\gamma \subseteq cgd0 \]

= [properties 1.34c and 1.34d]

\[ cgd0-\gamma_s \land cgd0-\gamma^* \]

= [def. \(cgd0\) (1.11)]

\[ (A s: SW-s: gdh-\Gamma_{10}, s \land gdh-\Gamma_{\gamma0}, s) \]

\[ \Rightarrow \text{[def. }gdh\text{ (1.11), and (from their definition): } \Gamma_{10} \subseteq \Gamma_{\gamma0}\text{]} \]

\[ (A s: SW-s: \Gamma_{ho}(g,s) = \Gamma_{\gamma0}(g,s)) \]

\[ \Rightarrow \text{[def. }G0\text{ (1.29)]} \]

\[ G0-\Gamma_{10} = G0-\Gamma_{\gamma0} \]

\[ \Rightarrow \text{[def. }\gamma_s, \gamma^*\text{ (1.32)]} \]

\[ \gamma_s = \gamma^* \]

\[ \square \]

From lemma 1.36, prop. 1.34a, and th. 1.28 follows that: \(WMO0-C-\gamma \Rightarrow \gamma_s = \gamma^* \land cgd0-\gamma_s\), for all \(C\) and \(\gamma\). Again, the question arises whether the converse of this property holds. Intuitively, \(\gamma_s\) and \(\gamma^*\) may seem to be the least and the greatest stable state (due to def. 1.29 (G0) and th. 1.31), and the converse may therefore seem true. Although states \(\gamma_s\) and \(\gamma^*\) are the least and the greatest state that are stable w.r.t. G0 (since they are formed with the least and the greatest fixpoint of \(R-\gamma0G0\), they need not be the least and greatest stable state (see ex. 1.38). In fact, the set of stable states need not contain a least and a greatest element at all. Thus, the converse of the property mentioned above does not hold for all \(C\) and \(\gamma\):

1.37 Property

\[ \neg(A, C, \gamma: CIR-C \land NST-\gamma: WMO0-C-\gamma = (\gamma_s = \gamma^* \land cgd0-\gamma_s)) \]

A counter example is given below (1.38).

1.38 Example

Consider circuit \(C\) with source-connection \(\delta = \{(x_0, \{L\}), (x_1, \{H\}), (x_2, \emptyset), (x_3, \{H\})\}\) as depicted below. Since \(\delta_s = \delta^*\) (left to the reader), there is only one feasible state w.r.t. G0, namely: \(\{(x_0, \{0\}), (x_1, \{L\}), (x_1, \{H\}), (x_2, \{H\}), (x_3, \{H\})\}\).

Notice that this state is completely gate defined-0, and hence: \(\delta_s = \delta^* \land cgd0-\delta_s\).
The following state is, however, also stable, and hence feasible, but not completely gate defined-0: \( \{(x_0,1), (x_1,0), (x_2,L,1), (x_2,L,1), (x_2,L,1), (x_3,L,1)\} \).

Consequently, \( \text{WM0-}\text{C-}\delta \) does not hold.

In the following theorem we prove that there is only one feasible state in case of well-matchedness. Although this is a nice result, it does not simplify the calculation of \( \text{WM0} \), nor does the corollary following this theorem.

1.39 Theorem

\[ (A, C, \gamma: \text{CIR-C } \land \text{NST-}\gamma: \text{WM0-}\text{C-}\gamma = (\text{feasible0}\cdot\gamma = \{y_e\} \land cgd0\cdot\gamma_e)) \]

Proof

\[ \text{WM0-}\text{C-}\gamma \]

\[ = \{\text{def. WM0 (1.25) and lemma 1.36}\} \]

\[ (\text{(*)} \quad (\text{feasible0}\cdot\gamma \subseteq cgd0) \land (\gamma_e = \gamma^*) \]

\[ = \{\text{note 0}\} \]

\[ (\text{(**) \quad (feasible0}\cdot\gamma \subseteq cgd0) \land (\text{feasible0}\cdot\gamma = \text{feasible0}\cdot\gamma) \land (\gamma_e = \gamma^*) \]

\[ = \{\text{lemma 1.35 (for a), and properties 1.34c and 1.34d for a)}\]

\[ (\text{feasible0}\cdot\gamma \subseteq cgd0) \land (\text{feasible0}\cdot\gamma = \{y_e\}) \]

\[ = \{\text{calculus}\} \]

\[ (\text{feasible0}\cdot\gamma = \{y_e\}) \land cgd0\cdot\gamma_e \]

Note 0

Since the implication (***) \( \Rightarrow \) (**) is trivial, we only prove the implication (**) \( \Rightarrow \) (**).

Assume \( \text{feasible0}\cdot\gamma \subseteq cgd0 \). We prove that \( \text{feasible0}\cdot\gamma \subseteq \text{feasible0}\cdot\gamma \). Property 1.34d then completes the proof.

Let \( \text{feasible0}\cdot\gamma \subseteq cgd0 \), and let \( \text{m}\) be such that \( (\text{next0} \cdot \gamma)^m \cdot (Q_0, \Gamma_0) = (Q_0, \Gamma_0) \).

Choose, for all \( i, 1 \leq i \leq m\), states \( (Q_i, \Gamma_i) \) such that:

\[ (A; i: 0 \leq i < m; \text{next0} \cdot \gamma; (Q_i, \Gamma_i); (Q_{i+1}, \Gamma_{i+1}) ) \land (Q_{m}, \Gamma_{m}) = (Q_0, \Gamma_0) . \]

Notice that, for all \( i: 0 \leq i < m\), \( \text{cgdo0} \cdot (Q_i, \Gamma_i) \). On account of def. 1.21 (next0), prop. 1.22e, and prop. 1.30a we can therefore conclude that, for all \( i: 0 \leq i < m\), \( (Q_{i+1}, \Gamma_{i+1}) \land (T_i = R \cdot \gamma \cdot Q_i) \). From def. 1.33 then follows that \( (\text{next00}\cdot\gamma)^m \cdot (Q_0, \Gamma_0) = (Q_0, \Gamma_0) \), and hence (using def. 1.33) \( \text{feasible00}\cdot\gamma \cdot (Q_0, \Gamma_0) \).
1.40 Corollaries

a \((A_{C;\gamma}: CIR-C \land NST-\gamma: WMO-C-\gamma = (stable_{0-\gamma} = \{\gamma_s\} \land cg_{0-\gamma} \gamma_s))\)
b \((A_{C;\gamma}: CIR-C \land NST-\gamma: WMO-C-\gamma \Rightarrow (feasible_{0-\gamma} = stable_{0-\gamma}))\)

The implication to the left in corollary 1.40a follows directly from theorem 1.28, and the implication to the right follows from theorem 1.39 and properties 1.24 and 1.34a. Property 1.40b follows directly from theorem 1.39 and corollary 1.40a.

The converse of property 1.40b (c) does not hold. This can be easily seen from the circuit \(\langle (x),(x),(s,[H]),(s,[x]),(s,[x,[x]]) \rangle\) with source-connection \(\gamma = \emptyset\). The circuit has two feasible states, which are both stable (viz. \(\langle (s,0),((s,0)) \rangle\) and \(\langle ((s,1),((s,1)) \rangle\)), but it is not well-matched.

The investigation of the monotonic switch function \(G0\) led to some nice results, viz., th. 1.39 and cor. 1.40. It did not lead to an equivalent expression for \(WMO\) that can be evaluated efficiently.

As said in the introduction of this section, we will also investigate the anti-monotonic switch function. Since the investigation of the anti-monotonic switch function is similar to the investigation of \(G0\) and does not lead to any major results for \(WMO\), it is treated in a compact way.

The anti-monotonic switch function \(G1\)

Similar arguments as used for \(G0\) (for the uniqueness of a monotonic switch function) lead to the uniqueness of the anti-monotonic switch function. This anti-monotonic switch function, called \(G1\) (def. 1.41 and prop. 1.42a,b), is interesting, since, as a result of this anti-monotonicity, \(R^{-\gamma}G1\) is anti-monotonic, and, hence, \((R^{-\gamma}G1)^2\) is monotonic (prop. 1.42c).

Similar as for \(R^{-\gamma}G0\), the least and greatest fixpoint of \((R^{-\gamma}G1)^2\) can be calculated efficiently (compare with th. 1.31). The circuit states formed with these fixpoints, called \(\gamma_s\) and \(\gamma^s\) (def. 1.43a), need not be stable (see ex. 1.45a and prop. 1.44a,b). In example 1.38 (counter example' for 1.37) the stable state different from \(\delta_s\) is stable w.r.t. \(G1\). This leads to the question whether \(WMO\) can be characterised by a similar formula as tried in 1.37, but now using, besides \(\gamma_s\) and \(\gamma^s\), also \(\gamma_{s+1}\) and \(\gamma^{s+1}\) (i.e., does the converse of prop. 1.44c hold?). This turns out not to be the case (see prop. 1.44d and ex. 1.45b).

1.41 Definition

\(G1 \in NST - SST\) is defined by:

\(G1 - \Gamma - s = \langle t - s \in \Gamma - (g - s) \rangle\)
1.42 Properties
a) $G_1 \in SF$

b) $G_1$ is anti-monotonic w.r.t. order $\preceq$ on its domain and order $\leq$ on its codomain.

c) $R \cdot \gamma \cdot G_1$ is anti-monotonic, and $(R \cdot \gamma \cdot G_1)^2$ is monotonic, both w.r.t. order $\preceq$ on both domain and codomain.

These properties follow by arguments similar to those used for properties 1.30.

1.43 Definition
a) Circuit states $\gamma_{s1}$ and $\gamma^{s1}$ are defined as:

\[
\gamma_{s1} = (G_1 \cdot (lfp \cdot ((R \cdot \gamma \cdot G_1)^2), (lfp \cdot ((R \cdot \gamma \cdot G_1)^2)))
\]

\[
\gamma^{s1} = (G_1 \cdot (gfp \cdot ((R \cdot \gamma \cdot G_1)^2), (gfp \cdot ((R \cdot \gamma \cdot G_1)^2)))
\]

b) $next_{01} \in NST \rightarrow ST0 \rightarrow \gamma(ST0)$ and feasible$01 \in NST \rightarrow ST0 \rightarrow B$ are defined by:

\[
next_{01} \cdot \gamma(Q, \Gamma) = \{ (G_1 \cdot \Gamma, R \cdot \gamma \cdot (G_1 \cdot \Gamma)) \}
\]

\[
\text{feasible}_{01} \cdot \gamma(\Pi) = (next_{01} \cdot \gamma)^s \cdot \Pi \cdot \Pi
\]

1.44 Properties
a) $(next^0 \cdot \gamma)^2 \cdot \gamma_{s1} \cdot \gamma_{s1} \land (next^0 \cdot \gamma)^2 \cdot \gamma^{s1} \cdot \gamma^{s1}$

b) $(\gamma_{s1} = \gamma^{s1}) = (stable \cdot \gamma \cdot \gamma_{s1} \land \text{feasible}_{01} \cdot \gamma \cdot \gamma_{s1} = \{ \gamma_{s1} \})$

c) $(\forall C, \gamma : CIR \cdot C \land NST \cdot \gamma : WMO \cdot C \cdot \gamma \neq \{ \gamma_{s1}, \gamma^{s1}, \gamma^{s1} \} = \{ \gamma_s \} \land cgd^0 \cdot \gamma_s \})$

d) $\neg (\forall C, \gamma : CIR \cdot C \land NST \cdot \gamma : WMO \cdot C \cdot \gamma \neq \{ \gamma_{s1}, \gamma^{s1}, \gamma^{s1} \} = \{ \gamma_s \} \land cgd^0 \cdot \gamma_s \})$

Property 1.44a follows directly from def. 1.43 and prop. 1.42a. Property 1.44b can be proven similarly to lemma 1.35. Property 1.44c follows from th. 1.39 using $\{ \gamma_{s1}, \gamma^{s1} \} \subseteq \text{feasible} \cdot \gamma$ (which follows from 1.44a). For property 1.44d: see example 1.45b.

1.45 Example
a) Consider circuit $C_1$ from example 1.20b with source-connection $\gamma = \{ (x, [L]), (y, \emptyset) \}$ as depicted below. For this situation $\gamma_s$, $\gamma^{s1}$, and $\gamma_{s1}$ equal $\{ (x, 0), (y, \emptyset) \}$, and $\gamma^{s1} = \{ (x, 1), (y, [L]), (y, [L]) \}$. Since $\text{feasible} \cdot \gamma_{s1} = \{ \gamma^{s1} \}$ and $\text{feasible} \cdot \gamma^{s1} = \{ \gamma_{s1} \}$, both $\gamma_{s1}$ and $\gamma^{s1}$ are not stable w.r.t. $G_1$. State $\gamma_{s1}$ is stable (w.r.t. $G_0$), but $\gamma^{s1}$ is not stable.

Circuit $C_1$ with source-connection $\gamma$:

```
  x  
  \  
  \  
  3    y
```
Consider circuit $C$ and source-connection $\delta = \{(x_0, \{H\}), (x_1, \{L\}), (x_2, \{L\}), (x_3, \{H\}), (x_4, \{L\}), (x_5, \{\emptyset\})\}$ as depicted below. In this case, states $\delta_a$, $\delta^a$, $\delta_{a1}$, and $\delta^{a1}$ are all equal to state (2) in the table below. To verify this, consider the approximations (starting in L or in $\top$, which are (0) and (5) in the table below, using $G0$ and $G1$): for $\delta_a : (0) \to (1) \to (2)$; for $\delta^a : (5) \to (4) \to (3) \to (2)$; for $\delta_{a1} : (0) \to (4) \to (1) \to (6) \to (2)$ (notice that only (0), (1), and (2) are approximations of $\delta_{a1}$, and (4) and (6) are not; this is because we apply $R^{-}\delta\circ G1$, and $\delta_{a1}$ is a fixpoint of $(R^{-}\delta\circ G1)^\infty$; for $\delta^{a1} : (5) \to (1) \to (6) \to (2)$. Since state (2) is completely gate defined-0, it is stable w.r.t. both $G0$ and $G1$ (use property 1.22c), and, moreover, $\{\delta_a, \delta^a, \delta_{a1}, \delta^{a1}\} = \delta_a \land cgdo \land \delta_a$ holds.

There is, however, another stable state, viz. (7), and, hence, $WM0\cdot C\cdot \delta$ does not hold (use cor. 1.40a or observe that this state is not cgdo and use the definition (1.25)). Consequently, the converse of property 1.44c does not hold.

Circuit $C$ with source-connection $\delta$:

Table of example 1.45b (note: $x_1, x_3, x_5$ are the gates of $s_0, s_1, s_2$ respectively):

<table>
<thead>
<tr>
<th></th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0)</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(2)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(3)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(4)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(5)</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(6)</td>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(7)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

$\blacksquare$ (end of example 1.45)
1.4 Other correctness criteria

Besides $cqd0$ two other correctness criteria are formulated. In the next section these correctness criteria are combined.

**Conflict-free**
First of all, as said in section 1.1, resulting states should be conflict-free. Conflict-freeness is required because conflicts can damage the circuit. It can be expressed as follows:

1.46 Definition
$CF \subseteq \mathcal{ST} \to \mathcal{B}$ is defined by:

$CF^-(Q, \Gamma) = (x : N \land x \lor \Gamma \land x \in \{L, H\})$

$CF$ stands for conflict-free.

**Correct state transition**
A third correctness criterion is a result of the abstraction we made from a 'switching period' to a 'switching point action' (see 'states' in section 0.2). Due to this abstraction some occurrences of spikes (or runt pulses) are not visible within our model. We will explain this phenomenon and formulate a correctness criterion on state transitions. In the next section, we will show that this correctness criterion is not needed in the basic model.

Consider the circuit depicted in figure 1.2a below; the usual CMOS inverter. Let node $x$ be connected to an L-source, and let the circuit be in the resulting stable state, i.e., let $s_0, s_1, x, y_0, y_1,$ and $z$ have values 0, 1, {L}, {L}, {H}, and {H} respectively. Let us consider the continuous voltage changes that occur at node $z$ if the state of node $x$ is changed. If the state of node $x$ is not changed, the voltage value follows scenario (a) depicted in figure 1.2b below. If the state of $x$ is changed into {H} the voltage value of $z$ follows scenario (b). If the state of $x$ is first changed into {H} and shortly after that changed into {L}, the voltage value of node $z$ can follow scenarios (a), (c), or (d). According to our basic model the state of $z$ will change, in the latter case, from {H} to {L} and back again, and, hence simulates scenario (c). Scenario (a) can be simulated in our model once we have weakened the delay assumptions. The possible

![Figure 1.2a: inverter](image1)

![Figure 1.2b: scenarios](image2)
occurrence of a spike in scenario (d) cannot be observed within our model. Such a spike can cause malbehaviour in the circuit, e.g., if node $z$ is a gate node. We therefore classify the latter case as incorrect, that is, the case that the state of the gate node changes immediately after the switch-state has changed. This is expressed in the following correctness criterion on state transitions, which requires stability of gates for inconsistent-0 switches (cf. def. 1.47).

In order for a state transition $(Q_0, \Gamma_0) \rightarrow (Q_1, \Gamma_1)$ to be correct, the states must satisfy

$$(A \ s : Q_0 \cdot s \neq Q_1 \cdot s : \Gamma_0 \cdot (g \cdot s) = \Gamma_1 \cdot (g \cdot s))$$

This will be expressed in a notion $cst\emptyset$ ('correctness state transition type 0'). It is a requirement on all state transitions between resulting (i.e. feasible) states. Since $cgd\emptyset$ is required for all resulting states (by $WM0$), $cst\emptyset$ can be defined as follows (explained by property 1.48b).

1.47 Definition

$cst\emptyset \in ST\emptyset \rightarrow ST\emptyset \rightarrow B$ is defined by:

$cst\emptyset-Q_0, \Gamma_0; (Q_1, \Gamma_1) = (A \ s : \neg consistent\emptyset-\Gamma_0 \cdot Q_0 \cdot s : \Gamma_0 \cdot (g \cdot s) = \Gamma_1 \cdot (g \cdot s))$

1.48 Properties

a) $(A \ \Pi : ST\emptyset \cdot \Pi : cst\emptyset \cdot \Pi)$

b) For all $((Q_0, \Gamma_0), (Q_1, \Gamma_1)) \in ST\emptyset$:

$cgd\emptyset-Q_0, \Gamma_0 \land next\emptyset-\gamma(Q_0, \Gamma_0)-(Q_1, \Gamma_1)$

$\Rightarrow (A \ s : SW \cdot s : (Q_0 \cdot s \neq Q_1 \cdot s) = \neg consistent\emptyset-\Gamma_0 \cdot Q_0 \cdot s)$

Property 1.48a follows directly from the definition of $cst\emptyset$ (1.47). We therefore only prove property 1.48b.

Proof of property 1.48b

Assume $cgd\emptyset-Q_0, \Gamma_0 \land next\emptyset-\gamma(Q_0, \Gamma_0)-(Q_1, \Gamma_1)$. Let $s \in SW$. Then:

$Q_0 \cdot s \neq Q_1 \cdot s$

= (note 0 using the assumption)

$gd\emptyset-\Gamma_0 \cdot s \land (Q_0 \cdot s \neq (t \cdot s = \Gamma \cdot (g \cdot s)))$

= (def. 1.12 (consistent\emptyset))

$\neg consistent\emptyset-\Gamma_0 \cdot Q_0 \cdot s$

Note 0

$cgd\emptyset-Q_0, \Gamma_0 \land next\emptyset-\gamma(Q_0, \Gamma_0)-(Q_1, \Gamma_1)$

$\Rightarrow$ (def. 1.11 (cgd\emptyset), def. 1.21 (next\emptyset))

$gd\emptyset-\Gamma_0 \cdot s \land coco\emptyset-\Gamma_0 \cdot Q_1$

$\Rightarrow$ (def. 1.12 (coco\emptyset))

$gd\emptyset-\Gamma_0 \cdot s \land (Q_1 \cdot s = (t \cdot s = \Gamma \cdot (g \cdot s)))$
1.5 Concluding remarks on chapter 1

In this chapter we have studied the initial behaviour of circuit – source-connection combinations using assumptions (a) to (e) mentioned in the introduction of this chapter. The main question we have asked is: does such a combination lead to correct resulting states, and, if it does, what are these resulting states? For correctness of the resulting states of a combination C, γ we have formulated the following three correctness criteria.

Well-matchedness
The first correctness criterion is well-matchedness (WM0, def. 1.25), i.e., feasible0 · γ ⊑ cgdo. Since the evaluation of WM0 by its definition is complex, we have searched for equivalent expressions for WM0 that can be evaluated more efficiently. The most important results of this search are theorem 1.28 and corollary 1.40 (which are, for convenience, listed below).

As a result of th. 1.28, WM0 can be calculated using the set of stable states instead of the set of feasible states. From cor. 1.40 follows that there is only one resulting state in case of well-matchedness. This resulting state can be calculated efficiently using switch function G0 (def. 1.29, 1.32). Notice that WM0 itself cannot be evaluated efficiently using these results, since for the evaluation of WM0 all circuit states need to be checked for stability.

Another important result is that WM0 cannot, in general, be calculated using the least and greatest fixpoints of R·γ·G0 and (R·γ·G1)2 (cf. prop. 1.37 with ex. 1.38, and prop. 1.44d with ex. 1.45b), which can be calculated efficiently by successive approximation. In the next chapter we define several classes of circuits for which WM0 can be calculated efficiently using these fixpoints.

Theorem 1.28 and corollary 1.40:
1.28 (A C, γ: CIR-C ∧ NST-γ: WM0·C·γ = (stable0 · γ ⊑ cgdo))
1.40a (A C, γ: CIR-C ∧ NST-γ: WM0·C·γ = (stable0 · γ = {γs} ∧ cgdo · γs))
1.40b (A C, γ: CIR-C ∧ NST-γ: WM0·C·γ = (feasible0 · γ = stable0 · γ))

Conflict-freeness
The second correctness criterion requires that all resulting states be conflict-free. On account of theorem 1.39 (or cor. 1.40) the set of resulting states is, in case of well-matchedness: {γs}.

The additional requirement for conflict-freeness can, therefore, be expressed as: CF·γs.

Correctness of state transitions
Notice that the third requirement, that is, correctness of all state transitions between resulting states, does not lead to an additional correctness criterion for resulting states, since (see prop. 1.48a) cex0 · γ · γs holds, and (in case WM0·C·γ holds, and hence also cgdo · γs) the only transition possible from γs is to itself (use prop. 1.22a).
Correctness criterion for initial behaviour

For correctness of initial behaviour of a circuit C with a source connection γ we therefore require, apart from correspondence between the specified values on the outputs and the resulting values on the outputs (which is handled in chapter 7):

\[ WM0 \cdot C \cdot γ \land CF \cdot γ^* \]
CHAPTER 2 ACYCLIC CIRCUITS

In chapter 1 we concluded that $WMO\cdot\xi$ cannot, in general, be evaluated efficiently using the least and greatest fixpoints of $R\cdot\eta\cdot G_0$ and $(R\cdot\eta\cdot G_1)$. In this chapter several classes of circuits are defined, called the classes of acyclic circuits, for which $WMO\cdot\xi$ can be evaluated efficiently using these fixpoints.

In the first section these classes are defined. In the second section we show that $WMO\cdot\xi$ can, for these classes, be characterised by a simple formula. In the last section some concluding remarks are made.

2.0 Acyclic circuits

A circuit is acyclic if it contains no cycles. The notion of cycle we use is the following: a cycle is a list of switches $s_0, s_1, ... s_n$ where $0 \leq n$, $s_0 = s_n$ and, for all $i : 0 \leq i < n$, a path exists from a pass node of $s_i$ to the gate node of $s_{i+1}$. A path from $s_0$ to $x_m$ is a list of nodes $s_0, x_1, ... x_n$ where $0 \leq m$ and, for all $j : 0 \leq j < m$, $[s_j, x_{j+1}]$ is the bag of pass nodes of a switch.

For example:

- acyclic: 

- cyclic:

We distinguish general acyclicity (2.0.0) and acyclicity w.r.t. a source-connection (2.0.1).

2.0.0 General acyclicity

Let $C = (N, SW, t, g, pn)$ be a circuit.

First we introduce the notion node-influence-relation. Basically, a node $x$ influences a node $y$ if $x$ is the gate node and $y$ a pass node of a switch, or if $x$ and $y$ are the pass nodes of a switch. Furthermore, this influence relation is reflexive and transitive (def. 2.0).
2.0 Definition

\begin{align*}
\text{bni0, ni0 } & \in N \rightarrow N \rightarrow B \text{ are defined by:} \\
\text{bni0-}\ x \cdot y & = (B s : SW \cdot s : (x = g \cdot s \land y \in pn \cdot s) \lor (x, y) = pn \cdot s) \\
\text{ni0} & \text{ is the reflexive and transitive closure of bni0} \\
\text{(b)ni0 stands for (basic-)node-influence-relation-0.}
\end{align*}

Now the second influence relation, switch-influence-relation, is easily defined: switch \ s_0 \ influences \ switch \ s_1 \ if \ a \ pass \ node \ of \ s_0 \ influences \ the \ gate \ node \ of \ s_1 \ (\text{def. 2.1}).\n
The switch-influence-relation thus defined is transitive (lemma 2.2).

2.1 Definition

\begin{align*}
\text{si0 } & \in SW \rightarrow SW \rightarrow B \text{ is defined by} \\
\text{si0-}\ s_0 \cdot s_1 & = (B x : x \in pn \cdot s_0 : \text{ni0-}\ x \cdot (g \cdot s_1)) \\
\text{si0 stands for switch-influence-relation-0.}
\end{align*}

2.2 Lemma

\text{si0 is transitive.}

Proof

\begin{align*}
\text{Let } s_0, s_1 \text{ and } s_2 \text{ be elements of } SW. \text{ Then:} \\
\text{si0-}\ s_0 \cdot s_1 & \land \text{si0-}\ s_1 \cdot s_2 \\
= & \quad (\text{def. 2.1 (si0)}) \\
& \quad (B x, y : x \in pn \cdot s_0 \land y \in pn \cdot s_1 : \text{ni0-}\ x \cdot (g \cdot s_1) \land \text{ni0-}\ y \cdot (g \cdot s_2)) \\
= & \quad (\text{def. 2.0 (ni0)}) \\
& \quad (B x, y : x \in pn \cdot s_0 \land y \in pn \cdot s_1 : \text{ni0-}\ x \cdot (g \cdot s_1) \land \text{ni0-}\ (g \cdot s_2) \cdot y \land \text{ni0-}\ y \cdot (g \cdot s_2)) \\
\Rightarrow & \quad \text{transitivity of ni0 (from def. 2.0)} \\
& \quad (B x : x \in pn \cdot s_0 : \text{ni0-}\ x \cdot (g \cdot s_2)) \\
= & \quad (\text{def. 2.1 (si0)}) \\
& \quad \text{si0-}\ s_0 \cdot s_2
\end{align*}

\( \square \)

A circuit contains a cycle if a switch influences itself. Acyclicity can therefore be defined as follows:

2.3 Definition

A0 \in CIR \rightarrow B \text{ is defined by: } A0 \cdot C = (A s : SW \cdot s : \neg\text{ni0-}\ s \cdot s)

If A0 \cdot C holds we say C is acyclic.
2.0.1 Acyclicness w.r.t. a source-connection

Let \( C = (N, SW, t, g, pn) \) be a circuit. In this section we regard \( C \) with a specific source-connection, say \( \gamma \). Acyclicness of \( C \) w.r.t. \( \gamma \) can be defined as:

1. \( C \) is acyclic w.r.t. \( \gamma \) if for each feasible state each cycle contains a switch that is nonconducting under that state.

For well-matched combinations \( C, \gamma \) only one feasible state exists (th. 1.39), and, hence, this formulation of acyclicness of \( C \) w.r.t. \( \gamma \) is equivalent to:

2. \( C \) is acyclic w.r.t. \( \gamma \) if each cycle in \( C \) contains a switch that is certainly nonconducting, i.e. nonconducting for each feasible state.

Since we are interested in well-matched combinations only, we may freely choose between the alternative formulations. We choose the second one, (2), because it is easier to formalise, and start with defining the complement of the notion 'certainly nonconducting'.

2.4 Definition

\[ PC \in NST \rightarrow SW \rightarrow \mathbb{B} \text{ is defined by:} \]

\[ PC \gamma s = (E Q, \Gamma : \text{feasible} \gamma (Q, \Gamma) : Q \cdot s) \]

\( PC \) stands for possibly conducting.

The remainder of this subsection is similar to subsection 2.0.0. A node-influence-relation and a switch-influence-relation are defined w.r.t. a specific source-connection, and, with their help, acyclicness w.r.t. a specific source-connection.

2.5 Definition

\( bnl \), \( nil \) \( \in NST \rightarrow N \rightarrow N \rightarrow \mathbb{B} \) are defined by:

\[ bnl \gamma x y = (E s : SW \cdot s \land PC \gamma s : (x \equiv g \cdot s \land y \in pn \cdot s) \lor ((x, y) = pn \cdot s)) \]

\( nil \gamma \) is the reflexive and transitive closure of \( bnl \gamma \)

(b)\( nil \) stands for (basic-) node-influence-relation-1.

2.6 Definition

\( sil \in NST \rightarrow SW \rightarrow SW \rightarrow \mathbb{B} \) is defined by

\[ sil \gamma x y = (PC \gamma s_0 \land (E x : x \in pn \cdot s_0 : nil \gamma x \cdot (g \cdot s_0))) \]

\( sil \) stands for switch-influence-relation-1.

2.7 Lemma

For all \( \gamma \in NST : sil \gamma \) is transitive.
Proof

Let \( \text{NST} \cdot \gamma \) and let \( s_0, s_1 \) and \( s_2 \) be elements of \( \text{SW} \). Then:

\[
\text{sil} \cdot \gamma \cdot s_0 \cdot s_1 \land \text{sil} \cdot \gamma \cdot s_1 \cdot s_2 \\
= \{ \text{def. 2.6 (sil)} \}
\]

\[
\text{PC} \cdot \gamma \cdot s_0 \land \text{PC} \cdot \gamma \cdot s_1 \\
\implies \{ \text{def. 2.5 (nil)} \}
\]

\[
\text{PC} \cdot \gamma \cdot s_0 \land (B \cdot x, y : x \in \text{pn} \cdot s_0 \land y \in \text{pn} \cdot s_1 : \text{nil} \cdot \gamma \cdot x \cdot (g, s_1) \land \text{nil} \cdot \gamma \cdot y \cdot (g, s_2))
\]

\[
= \{ \text{transitivity of nil} \cdot \gamma \text{ (from def. 2.5)} \}
\]

\[
\text{PC} \cdot \gamma \cdot s_0 \land (B \cdot x : x \in \text{pn} \cdot s_0 : \text{nil} \cdot \gamma \cdot x \cdot (g, s_2))
\]

\[
= \{ \text{def. 2.6 (sil)} \}
\]

\[
\text{sil} \cdot \gamma \cdot s_0 \cdot s_2
\]

\[\square\]

2.8 Definition

\( A1 \in \text{CIR} \rightarrow \text{NST} \rightarrow \mathbb{E} \) is defined by:

\( A1 \cdot \gamma = (A \cdot s : \text{SW} \cdot s : \neg \text{sil} \cdot \gamma \cdot s \cdot s) \)

If \( A1 \cdot \gamma \) holds we say \( C \) is acyclic w.r.t. source-connection \( \gamma \).

2.9 Example

Consider circuit \( C \) with source-connection \( \delta \) from example 1.38. It is depicted below.

![Circuit Diagram]

Recall, from example 1.38, that the following states are elements of \( \text{stable} \cdot \delta \), and, hence, elements of \( \text{feasible} \cdot \delta \) (the first one is \( \delta_0 \)):

\[
(Q_0, \Gamma_0) = (\{(s_0,0),(s_1,1)\}, \{(x_0,\{L\},x_1,\{H\}), (x_2,\{H\})\})
\]

\[
(Q_1, \Gamma_1) = (\{(s_0,1),(s_1,0)\}, \{(x_0,\{L\},x_1,\{L\},x_2,\{H\})\})
\]

Now observe that:

\[\cdot C \text{ is not acyclic:}\]

true

\[
= \{ \text{def. 2.0 (bni)} , \text{using } g \cdot s_0 = x_2 \text{ and } x_1 \in \text{pn} \cdot s_0 \}
\]

\[
\text{bni}0 \cdot x_2 \cdot x_1
\]

\[
\Rightarrow \{ \text{def. 2.0 (nil)} \}
\]

\[
\text{nil}0 \cdot x_2 \cdot x_1
\]
\[ \text{\{def. 2.1 (si0), using } x_2 \in \text{ pn-}s_1 \text{ and } g \cdot s_1 = x_1 \}\]
\[ \text{si0} \cdot s_1 \cdot s_1 \]
\[ \Rightarrow \text{\{def. 2.3 (A0)\}} \]
\[ \neg \text{A0-C} \]

- **C** is not acyclic w.r.t \( \delta \):
  
  true
  
  \[ \Rightarrow \text{\{def. Q_0 \text{ and } Q_1\}} \]
  
  \[ Q_1 \cdot s_0 \land Q_0 \cdot s_1 \]
  
  \[ \Rightarrow \text{\{def. 2.4 (PC)\}} \]
  
  \[ PC \cdot \delta \cdot s_0 \land PC \cdot \delta \cdot s_1 \]
  
  \[ \Rightarrow \text{\{def. 2.5 (ni1), using } g \cdot s_0 = x_2 \text{ and } x_1 \in \text{ pn-}s_0 \}\]
  
  \[ ni_1 \cdot \delta \cdot x_2 \cdot x_1 \land PC \cdot \delta \cdot s_1 \]
  
  \[ \Rightarrow \text{\{def. 2.6 (ni1), using } x_2 \in \text{ pn-}s_1 \text{ and } g \cdot s_1 = x_1 \}\]
  
  \[ ni_1 \cdot \delta \cdot s_1 \cdot s_1 \]
  
  \[ \Rightarrow \text{\{def. 2.8 (A1)\}} \]
  
  \[ \neg \text{A1-C}) \cdot \delta \]

- **C** can be acyclic w.r.t a source-connection,
  
  e.g. **C** is acyclic w.r.t \( \delta' = ((x_0, \emptyset), (x_1, \{-\}), (x_2, \{H\}), (x_3, \emptyset)) \):
  
  true
  
  \[ \Rightarrow \text{\{def. } \delta' \text{ and def. 1.23 (feasible0)\}} \]
  
  \[ (A, Q, \Gamma : \text{ feasible0-} \delta' \cdot (Q, \Gamma) : \Gamma \cdot x_1 = \{L\} \land \Gamma \cdot x_2 = \{H\}) \]
  
  \[ \Rightarrow \text{\{structure of } C_2 \} \]
  
  \[ (A, Q, \Gamma, s : \text{ feasible0-} \delta' \cdot (Q, \Gamma) \land SW \cdot s : \Gamma \cdot (g \cdot s) \neq t \cdot s) \]
  
  \[ \Rightarrow \text{\{def. 1.23 (feasible0) and def. 1.12 (coco0)\}} \]
  
  \[ (A, Q, \Gamma, s : \text{ feasible0-} \delta' \cdot (Q, \Gamma) \land SW \cdot s : \neg Q \cdot s) \]
  
  \[ \Rightarrow \text{\{def. 2.4 (PC)\}} \]
  
  \[ (A, s : SW \cdot s : \neg PC \cdot \delta' \cdot s) \]
  
  \[ \Rightarrow \text{\{def. 2.6 (ni1)\}} \]
  
  \[ (A, s : SW \cdot s : \neg ni_1 \cdot \delta' \cdot s) \]
  
  \[ \Rightarrow \text{\{def. 2.8 (A1)\}} \]
  
  \[ A1 \cdot C \cdot \delta' \]

\( \square (\text{end example 2.9}) \)
2.1 Relations between A0, A1 and WM0

In this section the main theorems of this chapter are given. The first theorem (2.11) gives the relation between A0 and A1. The following two theorems give an equivalent expression for WM0 that can be evaluated efficiently; the former (2.13) for circuits that are acyclic w.r.t. a source-connection, and the latter (2.14) for general acyclic circuits.

2.10 Lemma

a Let s be a switch and let x and y be circuit nodes. Then
\[ \text{ni0} \cdot x \cdot y = (E \, \gamma : \text{NST} \cdot \gamma : \text{PC} \cdot \gamma \cdot s \land \text{ni1} \cdot \gamma \cdot x \cdot y) \]

b Let s₀ and s₁ be switches. Then
\[ \text{ni0} \cdot s₀ \cdot s₁ = (E \, \gamma : \text{NST} \cdot \gamma : \text{si1} \cdot \gamma \cdot s₀ \cdot s₁) \]

Proof of lemma 2.10a

Let \( s \in \text{SW} \) and \( x, y \in \mathbb{N} \). Then
\[ (E \, \gamma : \text{NST} \cdot \gamma : \text{PC} \cdot \gamma \cdot s \land \text{ni1} \cdot \gamma \cdot x \cdot y) \]

\[ \Rightarrow \{ \text{instantiation} \} \]
\[ \text{PC} \cdot \gamma \cdot s \land \text{ni1} \cdot \gamma \cdot x \cdot y \]

\[ = \{ \text{note 1} \} \]
\[ \text{ni1} \cdot \gamma \cdot x \cdot y \]

\[ = \{ \text{note 2} \} \]
\[ \text{ni0} \cdot x \cdot y \]

Furthermore
\[ (E \, \gamma : \text{NST} \cdot \gamma : \text{PC} \cdot \gamma \cdot s \land \text{ni1} \cdot \gamma \cdot x \cdot y) \]

\[ \Rightarrow \{ \text{calculus} \} \]
\[ (E \, \gamma : \text{NST} \cdot \gamma : \text{ni1} \cdot \gamma \cdot x \cdot y) \]

\[ \Rightarrow \{ \text{induction, using note 3, def. 2.5 (ni1), and def. 2.0 (ni0)} \}
\[ \text{ni0} \cdot x \cdot y \]

Notice that the two implications above complete the proof. The notes are given below.

Note 1

\[ \Rightarrow \{ \text{def. } \top \} \]
\[ (A \, s : \text{SW} \cdot s : \top \cdot s = \{ \text{L,H} \} ) \]

\[ \Rightarrow \{ \text{def. 1.12 (coco0)} \} \]
\[ \text{coco0} \cdot \top \cdot \top \]

\[ \Rightarrow \{ \text{for all } \gamma, Q : \gamma \notin R \cdot \gamma \cdot Q \ (\text{prop. 1.18a), def. } \top \} \]
\[ \text{coco0} \cdot \top \cdot \top \land (\top = R \cdot \top \cdot \top) \]

\[ \Rightarrow \{ \text{def. 1.19 (stable0)} \} \]
\[ \text{stable} \cdot \top \cdot (\top, \top) \]
⇒ \{\text{prop. 1.24}\}

\text{feasible}(\text{T} \cdot (\text{T}, \text{T}))

⇒ \{\text{def. 2.4 (PC), def. T}\}

(A s : SW s : PC \cdot T \cdot s)

\text{Note 2}
true

⇒ \{\text{note 1}\}

(A s : SW s : PC \cdot T \cdot s)

⇒ \{\text{def. 2.0 (bn10) and 2.5 (bn11)}\}

bn1\cdot 1 = bn10

⇒ \{\text{def. 2.0 (ni0) and 2.5 (ni1 \cdot T)}\}

ni1 \cdot T = ni0

\text{Note 3}
Let \( u, v \in \mathbb{N} \). For local use in this note we define, for \( SW s : \)
\( I s = ( ( ( u = g \cdot s ) \wedge ( v \in \text{pn-s} ) ) \vee ( [ u, v ] = \text{pn-s} ) ) \). Then:

\[(E \gamma : \text{NST} \cdot \gamma : \text{bn1} \cdot \gamma \cdot u \cdot v)\]

= \{\text{def. 2.5 (bn11)}\}

\[(E \gamma : \text{NST} \cdot \gamma : (E s : SW s \wedge PC \cdot \gamma \cdot s : I s ))\]

⇒ \{\text{calculus}\}

\[(E \gamma, s : \text{NST} \cdot \gamma \wedge SW s \wedge PC \cdot \gamma \cdot s : I s )\]

⇒ \{\text{calculus}\}

\[(E s : SW s : I s )\]

= \{\text{def. 2.0 (bn10)}\}

bn10 \cdot u \cdot v

\(\square\) \text{ (end proof lemma 2.10a)}

Proof of lemma 2.10b

\[(E \gamma : \text{NST} \cdot \gamma : \text{si1} \cdot \gamma \cdot s0 \cdot s1 )\]

= \{\text{def. 2.6 (si1)}\}

\[(E \gamma : \text{NST} \cdot \gamma : PC \cdot \gamma \cdot s0 \wedge (E x : x \in \text{pn-s0} : \text{ni1} \cdot \gamma \cdot x \cdot (g \cdot s1) ))\]

⇒ \{\text{calculus}\}

\[(E \gamma : \text{NST} \cdot \gamma : (E x : x \in \text{pn-s0} : PC \cdot \gamma \cdot s0 \wedge \text{ni1} \cdot \gamma \cdot x \cdot (g \cdot s1) ))\]

⇒ \{\text{calculus}\}

\[(E x : x \in \text{pn-s0} : (E \gamma : \text{NST} \cdot \gamma : PC \cdot \gamma \cdot s0 \wedge \text{ni1} \cdot \gamma \cdot x \cdot (g \cdot s1) ))\]

= \{\text{lemma 2.10a}\}

\[(E x : x \in \text{pn-s0} : \text{si0} \cdot x \cdot (g \cdot s1) ))\]

= \{\text{def. 2.6 (si0)}\}

si0 \cdot s0 \cdot s1

\(\square\) \text{ (end proof lemma 2.10b)}
2.11 Theorem

For $C \in \text{CIR}$, $A0\cdot C = (A\gamma : \text{NST} \cdot \gamma : A1 \cdot C \cdot \gamma )$

Proof

\[
(A\gamma : \text{NST} \cdot \gamma : A1 \cdot C \cdot \gamma )
\]
\[
\{ \text{def. 2.8 (A1)} \}
\]
\[
(A\gamma : \text{NST} \cdot \gamma : (A s : \text{SW} \cdot s : \neg \text{nil} \cdot \gamma \cdot s \cdot s ))
\]
\[
\{ \text{calculus} \}
\]
\[
(A s : \text{SW} \cdot s : (A\gamma : \text{NST} \cdot \gamma : \neg \text{nil} \cdot \gamma \cdot s \cdot s ))
\]
\[
\{ \text{lemma 2.10} \}
\]
\[
(A s : \text{SW} \cdot s : \neg \text{nil} \cdot s \cdot s )
\]
\[
\{ \text{def. 2.3 (A0)} \}
\]
A0\cdot C

\[\square\]

2.12 Lemma

Let $(\gamma , \Gamma , \Gamma \_1) \subseteq \text{NST}$, $(Q_0 , Q_1) \subseteq \text{SST}$ be such that $\text{feasible0} \cdot \gamma ^\cdot (Q_0 , \Gamma ) \land \text{feasible0} \cdot \gamma ^\cdot (Q_1 , \Gamma )$. Let $s_0 \in \text{SW}$. Then:
\[
(A s : \text{nil} \cdot \gamma \cdot s \cdot s_0 : Q_0 \cdot s = Q_1 \cdot s) \Rightarrow (A x : N : cp \cdot Q_0 \cdot [x,g \cdot s_0] = cp \cdot Q_1 \cdot [x,g \cdot s_0])
\]

Proof

Assume $(A s : \text{nil} \cdot \gamma \cdot s \cdot s_0 : Q_0 \cdot s = Q_1 \cdot s)$. Then:

true
\[
\Rightarrow \{ \text{note 0, using feasible0} \cdot \gamma ^\cdot (Q_0 , \Gamma ) \}
\]
\[
(A x,y : \text{bcp} \cdot Q_0 \cdot [x,y] : \text{bnil} \cdot \gamma \cdot x \cdot y )
\]
\[
\Rightarrow \{ \text{induction} \}
\]
\[
(A x,y : cp \cdot Q_0 \cdot [x,y] : \text{nil} \cdot \gamma \cdot x \cdot y )
\]
\[
\Rightarrow \{ \text{instantiation} \}
\]
\[
(A x : cp \cdot Q_0 \cdot [x,g \cdot s_0] : \text{nil} \cdot \gamma \cdot x \cdot (g \cdot s_0))
\]
\[
\Rightarrow \{ \text{theorem A0 from appendix A, using note 1} \}
\]
\[
(A x : cp \cdot Q_0 \cdot [x,g \cdot s_0] : \text{cp} \cdot Q_1 \cdot [x,g \cdot s_0])
\]
The symmetry completes the proof.

The notes 0 and 1 are given below.

\textbf{note 0}

Let $\text{feasible0} \cdot \gamma ^\cdot (Q, \Gamma )$ and $(x,y) \subseteq N$. Then:
\[
\text{bcp} \cdot Q_0 \cdot [x,y]
\]
\[
\{ \text{def. 1.14 (bcp)} \}
\]
\[
(E s : \text{SW} \cdot s \land Q \cdot s : \text{pn} \cdot s = [x,y])
\]
\[
\Rightarrow \{ \text{def. 2.4 (PC), using feasible0} \cdot \gamma ^\cdot (Q, 1) \}
\]
\[
(E s : \text{SW} \cdot s \land \text{PC} \cdot \gamma \cdot s : \text{pn} \cdot s = [x,y])
\]
\[ \text{def. 2.5(bnil)} \]
\[ \text{bnil} \cdot \gamma \cdot x \cdot y \]

**Note 1**
\[ \text{nil} \cdot \gamma \cdot x \cdot (g \cdot s_0) \wedge \text{bcp-Q}_{\theta}[x,y] \]
\[ = \text{def. 1.14 (bcp)} \]
\[ \text{nil} \cdot \gamma \cdot x \cdot (g \cdot s_0) \wedge (\text{E } s : \text{SW} \cdot s \wedge Q_{\theta} \cdot s ; \text{pn} \cdot s = [x,y] ) \]
\[ \text{def. 2.4 (PC), using feasible} \gamma \cdot (Q, \Gamma) \]
\[ \text{nil} \cdot \gamma \cdot x \cdot (g \cdot s_0) \wedge (\text{E } s : \text{SW} \cdot s \wedge Q_{\theta} \cdot s \wedge \text{PC} \cdot \gamma \cdot s : \text{pn} \cdot s = [x,y] ) \]
\[ \text{def. 2.6 (si1)} \]
\[ \text{bcp-Q}_{\Gamma}[x,y] \]
\[ \text{assumption} \]
\[ \text{E } s : \text{SW} \cdot s \wedge Q_{\theta} \cdot s : \text{pn} \cdot s = [x,y] \]
\[ \text{def. 1.14 (bcp)} \]
\[ \text{bcp-Q}_{\Gamma}[x,y] \]
\[ \square \text{(end proof lemma 2.12)} \]

In the following theorem an equivalent expression for \( WM^0 \) that can be evaluated efficiently is given for circuits that are acyclic w.r.t. a specific source-connection.

### 2.13 Theorem
\( (A \cdot C \cdot \gamma : \text{CIR} \cdot C \wedge \text{NST} \cdot \gamma : \text{A1} \cdot C \cdot \gamma \) \( \Rightarrow (WM^0 \cdot C \cdot \gamma = cgdo \cdot \gamma_s) \) \)

**Proof**

Let \( C, \gamma \) be such that \( A1 \cdot C \cdot \gamma \). Then:
\[ WM^0 \cdot C \cdot \gamma \]
\[ = \text{cor. 1.40a} \]
\[ \text{stable} \cdot \gamma = \{ \gamma_s \} \wedge cgdo \cdot \gamma_s \]
\[ = \text{note 0} \]
\[ cgdo \cdot \gamma_s \]

**Note 0**
Assume \( cgdo \cdot \gamma_s \), let \( \gamma_s = (Q_{\theta}, \Gamma_{\theta}) \), and \( \text{stable} \cdot \gamma \cdot (Q, \Gamma) \). Then:
\[ (Q, \Gamma) \neq \gamma_s \]
\[ = \text{from assumptions: } \Gamma = R \cdot \gamma \cdot Q \wedge \Gamma_{\theta} = R \cdot \gamma \cdot Q_{\theta} \]
\[ Q \neq Q_{\theta} \]
\[ = \text{calculus} \]
\[ \text{E } s : \text{SW} \cdot s : Q \cdot s \neq Q_{\theta} \cdot s \]
\[ \text{note 0.0} \]
\[ \text{E } s \cdot \text{Q}_s \cdot \text{Q}_{\theta} \cdot \text{Q}_s : (A \cdot s : Q \cdot s \neq Q_{\theta} \cdot s : -\text{nil} \cdot \gamma \cdot s \cdot s_0) \]
\[ = \text{calculus} \]
\[ \text{E } s \cdot \text{Q}_s \cdot \text{Q}_{\theta} \cdot \text{Q}_s : (A \cdot s : \text{nil} \cdot \gamma \cdot s \cdot s_0 : Q \cdot s = Q_{\theta} \cdot s) \]
In theorem 2.14 an equivalent for $WM0$ that can be evaluated efficiently is given for general acyclic circuits.

2.14 Theorem

Let $C$ be a circuit. Then:

$$A0\cdot C \rightarrow (A \gamma : NST\cdot \gamma : WM0\cdot C\cdot \gamma = cgd0\cdot \gamma_a)$$

Proof

$$A0\cdot C$$

$$= \{\text{theorem 2.11}\}$$

$$(A \gamma : NST\cdot \gamma : A1\cdot C\cdot \gamma)$$

$$= \{\text{theorem 2.13}\}$$

$$(A \gamma : NST\cdot \gamma : WM\cdot C\cdot \gamma = cgd0\cdot \gamma_a)$$

$\square$
2.2 Concluding remarks on chapter 2

In chapter 1 we looked for a way to express \( WM0-C \cdot \gamma \) in terms of \( \gamma_{*} \), \( \gamma^{*} \), \( \gamma_{-1} \), \( \gamma^{1} \) (cf.1.37 and 1.44d). In chapter 1 we concluded that it is in general not possible to express \( WM0-C \cdot \gamma \) in terms of these fixpoints. For the classes of acyclic circuits defined in this chapter, it is possible; the calculation of \( WM0-C \cdot \gamma \) requires only \( \gamma_{*} \) (which can be calculated efficiently, cf. 1.31), namely:

\[
\begin{align*}
2.13 & \quad (A \circ C : CIR-C \land NST-\gamma : A1-C \cdot \gamma \RightarrowWM0-C \cdot \gamma = cgdo-\gamma_{*}) \\
2.14 & \quad (A \circ C : CIR-C : A0-C \Rightarrow (A \circ NST-\gamma : WM0-C \cdot \gamma = cgdo-\gamma_{*})
\end{align*}
\]

Acyclicness is not a complete classification of well-matchedness, i.e., not all well-matched combinations \( C, \gamma \) are acyclic. Consider, for instance, circuit \( C_{1} \) with source-connection \( \delta = \{(s,\emptyset), (y,\{H\})\} \) depicted below. Since the only feasible state, \( \{(s,1), (x,\{H\}), (y,\{H\})\} \), is completely gate defined-0, \( WM0-C_{1} \cdot \delta \) holds. But, since \( s \) is conducting in this state, also \( \neg A1-C_{1} \cdot \delta \) holds (and, hence, also \( \neg A0-C_{1} \)).

Circuit \( C_{1} \) with \( \delta \):

Since well-matchedness can be calculated efficiently in case of acyclicness, it seems useful to regard circuits that are acyclic w.r.t. their initial source-connection. Note that general acyclic circuits are less interesting since, due to the absence of cycles, general acyclic circuits cannot be sequential. Whether a circuit is acyclic w.r.t. a specific source-connection is, in general, not an easy question, but in many examples acyclicness w.r.t. a source-connection is easy to determine. We will give a simple example here and return to this point in chapter 7.

2.15 Example

Consider the circuit \( C \) depicted below — a latch — with the source-connection \( \gamma \) as depicted. Since, on account of monotonicity of \( R \cdot \gamma \) (prop. 1.18c), for each feasible state \( (Q,\Delta) : R \cdot \gamma \in \Delta \subseteq R \cdot \gamma \cdot T \) , we can conclude from \( R \cdot \gamma \cdot F : \gamma \subseteq \{H\} \) and \( R \cdot \gamma \cdot T : \gamma = \{H\} \), that \( \Delta \cdot \gamma : \{H\} \). As a result of this \( \neg Q \cdot s_{1} \) holds, and, hence, \( C \) is acyclic w.r.t. \( \gamma \), that is, \( A1-C \cdot \gamma \) holds. The resulting state, \( \gamma_{*} \), is 1 in \( x_{0}, x_{2} \), and \( x_{5} \); 0 in \( x_{1}, x_{3}, \) and \( x_{4} \); \{L\} in \( x, x_{0}, \) and \( x_{5} \), and \{H\} in \( y \) and \( z_{1} \). Since this state is completely gate defined-0, that is, \( cgdo-\gamma_{*} \) holds, this \( C \) and \( \gamma \) are well-matched (use 2.13): \( WM0-C \cdot \gamma \).

\[ C, \gamma \text{ from example 2.15:} \]
CHAPTER 3  REACTION - DELAYS

In this chapter, restriction (c) from section 0.3, that is, reaction-delays are uniform (and positive and finite), is weakened. The other restrictions used in chapter 1 — (a), (b), (d), and (e) (see section 0.3) — are still used. Our approach is to extend the basic model from chapter 1 in such a way that arbitrary — but positive and finite — reaction-delays can be captured, and restrictions on reaction-delay can be expressed.

As said in section 0.2, reaction-delay models the time it takes for a switch to adapt its conductance state to the state of its gate node. We do not regard absolute estimates for reaction-delays. We consider relative reaction-delays, that is, we consider possible orders in which switches that are willing to change state do change state.

In the first section we regard arbitrary reaction-delays, and we study how they can be modelled within the framework of the basic model. The only assumption on reaction-delays is that they are positive and finite. In section 3.1 we investigate the properties of the newly defined notions, and the relation between them and the notions from chapter 1. In section 3.2 reaction-delay restrictions are discussed. In section 3.3 the notion well-functioning (see section 0.2) is considered briefly. Finally, in section 3.4 some concluding remarks on chapter 3 are made.

3.0  Modelling arbitrary reaction-delays

In this section we study arbitrary — but positive and finite — relative reaction-delays. We will redefine some of the notions from chapter 1.

Let $C$ be a circuit and $\gamma$ a source-connection of $C$.

Reaction-delay models the time it takes for a switch to adapt its conductance state to the state of its gate node. Consequently, arbitrary reaction-delay means that a switch can, in each step, either adapt its conductance state to the state of its gate node, or remain in the current conductance state. Now recall that a state transition in the basic model (def. 1.21 ($next\theta\gamma$)) is defined as a transition from a state $(Q_0,\Gamma_0)$ to a state $(Q, R\cdot\gamma\cdot Q)$ where $Q$ is restricted by:

$$(A:\text{SW} : consistent\theta\gamma\cdot Q : S)$$
For a switch $s$, consistent$\gamma^Q\cdot Q\cdot s$ means that the conductance state of $s$ is adapted to the state of its gate node. So, with arbitrary reaction-delays, a state transition from $(Q_0, \Gamma_0)$ to $(Q_1, \Gamma_1, \gamma^Q)$ should satisfy $(A s : SW \cdot s : Q_1, s \in \{Q_0, s, Q, s\})$ with $Q$ as above. Since the choice between $Q_0, s$ and $Q, s$ must be arbitrary, this means that the only restriction on $Q_1$ is:

$$(A s : SW \cdot s : \neg\text{consistent}^\gamma \Gamma_0, Q_1, s \rightarrow (Q_1, s = Q_0, s))$$

Note that this condition only restricts the behaviour of gate defined-0 switches (see prop. 1.13a); as in chapter 1, we do not wish to restrict the behaviour of switches that are not gate defined-0.

Besides arbitrary, reaction-delays are assumed to be positive and finite. Modelling a state transition as described above already forces delays to be positive, viz., the conductance state of a switch is adapted to the current state of the gate and not to a future state of the gate.

It is necessary to model the finiteness of reaction-delay in a convenient way. Not taking care of this finiteness in the definition of the next-state function would give rise to a finite-delay requirement in the definition of feasible states or to a restriction on the feasible states to be considered (cf. 'transient cycle', [BrzS1]). We will, therefore, take care of finiteness of reaction-delay in the next-state function; we will do this by means of a countdown function.

This means that we associate a natural number—the counter—with each switch. For consistent-0 switches the counter is 0 (R2.0 below). If a switch becomes inconsistent-0, the counter is set on an arbitrary number denoting the relative delay (R2.2). The counter then counts down (by 1 each step) to 0 (R2.1). As long as the counter is positive ("the delay is active"), the switch remains in its state (R0.1); if the counter is 0, the switch state will be adapted in the next step to the state of the gate node (R0.0).

This can be formalised by choosing a set of delay counters as $DC = SW \rightarrow \mathbb{N}$, and extending, for the use in the next-state function only, the states $\Pi \in STO$ to $(\Pi, \Gamma, df)$ with $df \in DC$.

Then $((Q_0, \Gamma_0), dc_0)$ is a successor of $(Q_0, \Gamma_0), dc_0)$ if:

- R0.0: $(A s : dc_0, s = 0 : \text{consistent}^\gamma \Gamma_0, Q_1, s)$
- R0.1: $(A s : dc_0, s > 0 : Q_1, s = Q_0, s)$
- R1.0: $\Gamma_1 = \Gamma_0, \gamma^Q$
- R2.0: $(A s : \neg\text{consistent}^\gamma \Gamma_0, Q_1, s : dc_1, s = 0)$
- R2.1: $(A s : \neg\text{consistent}^\gamma \Gamma_1, Q_1, s \land (dc_0, s > 0) : dc_1, s = dc_0, s - 1)$
- R2.2: $(A s : \neg\text{consistent}^\gamma \Gamma_1, Q_1, s \land (dc_0, s = 0) : dc_1, s \in \mathbb{N})$

Restrictions R0.1 restrict the next switch-state, restriction R1.0 restricts the next node-state; and restrictions R2.1 restrict the next reaction-delay counter. They are explained above.
Remark on R0.1 Notice that the reaction-delay counter thus defined, rc, denotes the exact number of steps the reaction-delay is supposed to be active. It may seem to allow more freedom to choose rc as the maximum number of steps. Then R0.1 changes into

\[ \text{R0.1alt} \quad \text{(A s : rc_0 \cdot s > 0 : (Q_1 \cdot s = Q_0 \cdot s) V consistent0-Γ_0 \cdot Q_1 \cdot s)} \]

On account of the arbitrary choice of the initial value of rc (in R2.2), the resulting models are equivalent w.r.t. the described behaviour. The choice for R0.1 in favor of R0.1alt allows a simpler modelling of reaction-delay restrictions.

The main advantage of using such a countdown function is that it takes care of finiteness of delay in the next-state function. It also has advantages for defining delay restrictions, as we will see in section 3.2. A disadvantage of the way we use the countdown function is that 'vacuous' transitions are possible, viz., if none of the counters of inconsistent-0 switches is 0, and hence, the state need not change. This is only a minor disadvantage since these vacuous transitions can, if necessary, be filtered out later.

3.0 Definition

The set of delay counters DC is defined by:

\[ \text{DC = SW - \#} \]

The bottom element of DC is denoted by 0; it satisfies:

\[ \text{(A s : SW \cdot s : 0 \cdot s = 0)} \]

The set of next states of an extended state (TLrc) is defined as the set of elements of ST0×DC that satisfy the restrictions R0.0 .. R2.2 formulated above:

3.1 Definition

next₁ ∈ NST → (ST0×DC) → 〈ST0×DC〉 is defined by:

\[ \text{next₁} \cdot γ ((Q_0, Γ_0, rc_0) \cdot ((Q_1, Γ_1), rc_1)) = \]

\[ \text{(A s : rc_0 \cdot s = 0 : consistent0-Γ_0 \cdot Q_1 \cdot s)} \]

\[ \land \text{(A s : rc_0 \cdot s > 0 : Q_1 \cdot s = Q_0 \cdot s)} \]

\[ \land \text{(Γ_1 = R \cdot γ Q_0)} \]

\[ \land \text{(A s : consistent0-Γ_1 \cdot Q_1 \cdot s : rc_1 \cdot s = 0)} \]

\[ \land \text{(A s : ~consistent0-Γ_1 \cdot Q_1 \cdot s \land (rc_0 \cdot s > 0) : rc_1 \cdot s = rc_0 \cdot s - 1)} \]

\[ \land \text{(A s : ~consistent0-Γ_1 \cdot Q_1 \cdot s \land (rc_0 \cdot s = 0) : rc_1 \cdot s \in \#)} \]

In a similar way as in chapter 1 (recall prop. 1.22b and def. 1.23) we can define stable and feasible states. The reaction-delay counter is for use in the next-state function only; it is hidden in the definition below (as a result of which stable₁ and feasible₁ have the same type as stable₀ and feasible₀, and, hence, the notions can be compared more easily).

We can define WM₁ (def. 3.3) in a similar way as WM₀ (def. 1.25).
3.2 Definition

Stable and feasible, both with type NST → ST0 → Σ, are defined by:

\[
\text{stable} \cdot \gamma \cdot \Pi = (B \cdot rc \cdot DC \cdot rc \cdot (nextI \cdot \gamma \cdot (\Pi_{Lc}) \cdot (\Pi_{rc})))
\]

\[
\text{feasible} \cdot \gamma \cdot \Pi = (B \cdot rc \cdot DC \cdot rc \cdot (nextI \cdot \gamma) \cdot (\Pi_{rc}) \cdot (\Pi_{rc}))
\]

3.3 Definition

\( WM1 \in CIR \rightarrow NST \rightarrow \Sigma \) is defined by:

\( WM1 \cdot C \cdot \gamma = (\text{feasible} \cdot \gamma \leq \text{cgdi}) \)

In the next section we investigate properties of the notions defined above and the relation between them and the notions from chapter 1. At the end of the next section we will regard the two additional correctness criteria defined in section 1.4 (CF and csdi), and consider their interpretation in the extended model in this chapter.

3.1 Properties of WM1 and the relation to the basic model

The investigation of WM1 is inspired by the investigation of WM0 in chapter 1. In order to check whether similar results as achieved for WM0 in chapter 1 hold for WM1, we investigate the relation between the notions defined in this chapter and the notions from chapter 1.

3.4 Property

a) \( (\gamma \in NST \cdot \gamma \cdot \text{stable} \cdot \gamma \subseteq \text{feasible} \cdot \gamma) \)

b) For all \( \gamma \in NST \) and \( (\Pi_{0}, \Pi_{1}) \subseteq ST0 \):

\( \text{nextI} \cdot \gamma \cdot (\Pi_{0}, \Pi_{0}) \cdot (\Pi_{1}, 0) = \text{nextI} \cdot \gamma \cdot (\Pi_{1}, \Pi_{1}) \)

Property 3.4a below follows directly from definition 3.2. Property 3.4b is proved below.

Proof of 3.4b

Let \( \gamma \in NST \) and \( \{(Q_{0}, \Gamma_{0}), (Q_{1}, \Gamma_{1})\} \subseteq ST0 \). Then:

\( \text{nextI} \cdot \gamma \cdot ((Q_{0}, \Gamma_{0}), 0) \cdot ((Q_{1}, \Gamma_{1}), 0) \)

\( = \{ \text{def. 3.1 (nextI)} \}

(\( A \cdot s \cdot 0 \cdot s = 0 \cdot \text{consistent} \cdot \Gamma_{0} \cdot Q_{1} \cdot s \)

\( \land (A \cdot s \cdot 0 \cdot s > 0 \cdot Q_{1} \cdot s = Q_{0} \cdot s) \)

\( \land (\Gamma_{1} = R \cdot \gamma \cdot Q_{0}) \)

\( \land (A \cdot s \cdot \text{consistent} \cdot \Gamma_{1} \cdot Q_{1} \cdot s \cdot 0 \cdot s = 0) \)

\( \land (A \cdot s \cdot \neg \text{consistent} \cdot \Gamma_{1} \cdot Q_{1} \cdot s \land (0 \cdot s > 0) \cdot 0 \cdot s = 0 \cdot s - 1) \)

\( \land (A \cdot s \cdot \neg \text{consistent} \cdot \Gamma_{1} \cdot Q_{1} \cdot s \land (0 \cdot s = 0) \cdot 0 \cdot s \in \mathbb{N}) \)
\[= \{ \text{calculus, def. 1.12 (coco0)} \} \]
\[\text{coco0} \cdot \Gamma_0 \cdot Q_1 \land (\Gamma_1 = R \cdot \gamma \cdot Q_1) \]
\[= \{ \text{prop. 1.22a} \} \]
\[\text{next} \cdot \gamma \cdot (Q_0, \Gamma_0) \land (Q_2, \Gamma_1) \]
\[
\qed
\]

The relation between \textit{stable1} and \textit{stable0} is quite simple:

\subsection*{3.5 Lemma}
\textit{stable1} = \textit{stable0}

\textbf{Proof}

Let \( \gamma \notin \text{NST} \) and \((Q, \Gamma) \in \text{ST0}\). Then:

\[\text{stable1} \cdot \gamma (Q, \Gamma)\]
\[= \{ \text{def. 3.2 (stable1)} \} \]
\[(E \text{rec} : \text{DC-rec} : \text{next} \cdot \gamma \cdot ((Q, \Gamma) \text{rec}) \land ((Q, \Gamma) \text{rec})) \]
\[= \{ \text{def. 3.1 (next))} \} \]
\[(E \text{rec} : \text{DC-rec} : (A \text{s} : \text{rec-s} = 0 : \text{consistent0} \cdot \Gamma \cdot Q \cdot s) \land (A \text{s} : \text{rec-s} > 0 : Q \cdot s = Q \cdot s) \land (\Gamma = R \cdot \gamma \cdot Q) \land (A \text{s} : \text{consistent0} \cdot \Gamma \cdot Q \cdot s \land (\text{rec-s} = 0) : \text{rec-s} = \text{rec-s} - 1) \land (A \text{s} : \neg \text{consistent0} \cdot \Gamma \cdot Q \cdot s \land (\text{rec-s} = 0) : \text{rec-s} \in \mathbb{N}) \} \]
\[= \{ \text{calculus} \} \]
\[(E \text{rec} : \text{DC-rec} : (A \text{s} :: (\text{rec-s} = 0) = \text{consistent0} \cdot \Gamma \cdot Q \cdot s) \land (\Gamma = R \cdot \gamma \cdot Q) \land (A \text{s} :: \neg \text{consistent0} \cdot \Gamma \cdot Q \cdot s \land (\text{rec-s} > 0) : \text{rec-s} = \text{rec-s} - 1) \} \]
\[= \{ \text{calculus} \} \]
\[(E \text{rec} : \text{DC-rec} : (A \text{s} :: (\text{rec-s} = 0) = \text{consistent0} \cdot \Gamma \cdot Q \cdot s) \land (\Gamma = R \cdot \gamma \cdot Q) \land (A \text{s} :: \neg \text{consistent0} \cdot \Gamma \cdot Q \cdot s) \} \]
\[= \{ \text{calculus} \} \]
\[(E \text{rec} : \text{DC-rec} : (\text{rec} = 0) \land (\Gamma = R \cdot \gamma \cdot Q) \land \text{coco0} \cdot \Gamma \cdot Q) \]
\[= \{ \text{calculus, using } 0 \in \text{DC} \} \]
\[(\Gamma = R \cdot \gamma \cdot Q) \land \text{coco0} \cdot \Gamma \cdot Q \]
\[= \{ \text{def. 1.19 (stable0)} \} \]
\[\text{stable0} \cdot \gamma (Q, \Gamma) \]
\[
\qed
\]
With the introduction of arbitrary reaction-delays versus uniform reaction-delays in chapter 1, one expects an increase of the number of feasible states. This is expressed in property 3.6b below. As a result of this, correctness of all 'new' feasible states implies correctness of all 'old' feasible states: property 3.6c.

3.6 Property
a  \((A \gamma, \Pi) : NST.\gamma \land ST0.\Pi) \Rightarrow stable1.\gamma.\Pi = nexti.\gamma.(\Pi,0) \setminus (\Pi,0)\)
b  \((A \gamma : NST.\gamma) \Rightarrow feasibele0.\gamma \subseteq feasibele1.\gamma\)
c  \((A C,\gamma : CIR-C \land NST.\gamma) : WM1.C.\gamma \Rightarrow WM0.C.\gamma\)

Property 3.6a follows from lemma 3.5 and properties 3.4b and 1.22b. Property 3.6b is proved below, and property 3.6c follows directly from the definitions of WM1 (3.3) and WM0 (1.25) and property 3.6b.

Proof of 3.6b
Let \(\gamma \in NST\) and \(\Pi \in ST0\). Then:
\[\text{feasibele0}.\gamma.\Pi\]
\[= \{ \text{def. 1.23 (feasibele0)} \}\]
\[\text{(next0}.\gamma^{+} \cdot \Pi.\Pi\)
\[\Rightarrow \{ \text{prop. 3.4b} \}\]
\[\text{(next1}.\gamma^{+} \cdot (\Pi,0) \setminus (\Pi,0)\)
\[\Rightarrow \{ \text{def. 3.2 (feasibele1), using } \theta \in DC \} \]
\[\text{feasibele1}.\gamma.\Pi\]

As an immediate result of def. 3.3 (WM1) and prop. 3.4a we have:

3.7 Property
\[(A C,\gamma : CIR-C \land NST.\gamma) : WM1.C.\gamma \Rightarrow ( stable1.\gamma \subseteq c\&d\theta )\]

Similar as in chapter 1, we have investigated the converse of this property (compare with 1.27 and 1.28). The converse turns out to be true: th. 3.8 below. The proof of this theorem is very similar to the proof of th. 1.28; it is given in appendix B1.

3.8 Theorem
\[(A C,\gamma : CIR-C \land NST.\gamma) : WM1.C.\gamma = ( stable1.\gamma \subseteq c\&d\theta )\]

Proof: See appendix B1
Theorem 3.8 is a powerful result. Using this theorem we can now prove that $WM1$ and $WM0$ are equal, which is a very interesting result since it means that — as far as the model thus far is concerned — we need not regard the refined model with arbitrary reaction-delays to calculate well-matchedness. As a consequence of this, the complete correctness criterion for initial behaviour in the basic model (in 1.49) is equal to the one in this model (cf. end of this section).

3.9 Theorem

$WM1 = WM0$

Proof

$WM1\cdot C\cdot \gamma$

$= \{ \text{th. 3.8} \}$

$\text{stabeI}_1 \cdot \gamma \subseteq \text{cgdo}$

$= \{ \text{lemma 3.5} \}$

$\text{stabeI}_0 \cdot \gamma \subseteq \text{cgdo}$

$= \{ \text{th. 1.28} \}$

$WM0\cdot C\cdot \gamma$

$\Box$

In case of well-matchedness, we are also interested in the set of resulting states $\text{feasibleI}_1 \cdot \gamma$. In general, $\text{feasibleI}_1 \cdot \gamma$ and $\text{feasibleI}_0 \cdot \gamma$ are not equal. The question arises whether they are equal in case of well-matchedness. We have investigated the resulting states in theorem 3.10 analogously to theorem 1.39 and corollary 1.40.

3.10 Theorem

a. $(A \cdot C \cdot \gamma; \text{CIR} \cdot C \land \text{NST} \cdot \gamma; WM1\cdot C\cdot \gamma = (\text{stabeI}_1 \cdot \gamma = \{ \gamma_s \} \land \text{cgdo} \cdot \gamma_s))$

b. $(A \cdot C \cdot \gamma; \text{CIR} \cdot C \land \text{NST} \cdot \gamma; WM1\cdot C\cdot \gamma = (\text{feasibleI}_1 \cdot \gamma = \{ \gamma_s \} \land \text{cgdo} \cdot \gamma_s))$

c. $(A \cdot C \cdot \gamma; \text{CIR} \cdot C \land \text{NST} \cdot \gamma; WM1\cdot C\cdot \gamma \Rightarrow (\text{feasibleI}_1 \cdot \gamma = \text{stabeI}_1 \cdot \gamma))$

Theorem 3.10a follows directly from the theorem 3.9 and corollary 1.40a since the set $\text{stabeI}_1 \cdot \gamma$ is equal to $\text{stabeI}_0 \cdot \gamma$ (lemma 3.5). Notice that theorem 3.10b does not follow that easily, since the set $\text{feasibleI}_1 \cdot \gamma$ is larger than the set $\text{feasibleI}_0 \cdot \gamma$ (prop. 3.6b). In appendix B1 we have proven a helpful lemma (cf. proof 3.10b). Theorem 3.10c follows directly from 3.10a and 3.10b.

Proof of theorems 3.10a and 3.10b

Let $C \in \text{CIR}$ and $\gamma \in \text{NST}$. Then:

$WM1\cdot C\cdot \gamma$

$= \{ \text{th. 3.9} \}$

$WM0\cdot C\cdot \gamma$
\( \{ \text{cor. 1.40a} \} \)
\[ \text{stable}^0 \cdot \gamma = \{ \gamma_s \} \land \text{cg} \overline{d} \cdot \gamma_s \]
\( \{ \text{lemma 3.5} \} \)
\[ \text{stable}_1 \cdot \gamma = \{ \gamma_s \} \land \text{cg} \overline{d} \gamma_s \]
\( \{ \text{lemma B1 in appendix B1 (for \( \overrightarrow{a} \)), and prop. 3.4a, 1.34a, and lemma 3.5 (for \( \overrightarrow{a} \))} \} \)
\[ \text{feasible}_1 \cdot \gamma = \{ \gamma_s \} \land \text{cg} \overline{d} \gamma_s \]

**Other correctness criteria**

The first correctness criterion regarding initial behaviour is \( WM1 \). As in section 1.4, we have, besides well-matchedness, two other correctness criteria. The second correctness criterion requires that all resulting states are conflict-free (def. 1.46). The third correctness criterion (\( \text{cst}^0 \): def. 1.47) requires that all transitions between resulting states are correct, that is, all switches that are, in some resulting state, in the process of changing state (are inconsistent-4) have stable gates. Similar as in chapter 1 (see section 1.5), the set of resulting states for a circuit \( C \) with a source-connection \( \gamma \) is, if \( WM1 \land \gamma \) holds, the singleton \( \{ \gamma_s \} \) (th. 3.10b). Consequently, the additional requirement for the second and third criterion is: \( CF \cdot \gamma_s \) (compare with section 1.5).

### 3.2 Modelling restricted reaction - delays

Thus far, we have only considered the correctness of the behaviour of a circuit with a given source-connection if the previous state of the circuit is unknown (and have expressed this in the notion \( \text{Well-Matched} \)). Besides this, we are interested in the correctness of the behaviour of a circuit starting in a known state when the source-connection changes (cf. section 0.2). The latter is to be expressed in the notion \( \text{Well Functioning} \).

On the one hand the knowledge about the 'previous' state of the circuit restricts the states to be considered, viz., using this knowledge it is possible to exclude certain feasible states as possible resulting states because they are not reachable from the previous state. On the other hand this knowledge enlarges the number of states to be considered, since it is possible to indicate the 'intermediate states'.

With the help of some small examples we will show that the functioning even of very small and simple circuits is not 'well' at all if the delays are not restricted. We therefore analyse how delay restrictions can be modelled. Before considering well-functioning any further (in section 3.3), the consequences of restrictions on delays for well-matchedness are investigated.
3.11 Example

Consider the circuit depicted in figure 3.0a below; the usual CMOS inverter. Let the circuit be in the resulting stable state of \( \gamma = \{ (x_1, L), (y_0, L), (y_1, H), (z, 0) \} \), which is \( \gamma_s = \{ (s_0, 0), (s_1, 1) \}, \{ (x_1, L), (y_0, L), (y_1, H), (z, H) \} \). Let \( x \) be changed to \( H \), that is, let the source connection be changed to \( \delta = \{ (x_1, H), (y_0, L), (y_1, H), (z, 0) \} \). The resulting state is \( \delta_s = \{ (s_0, 1), (s_1, 0) \}, \{ (x_1, H), (y_0, L), (y_1, H), (z, L) \} \). The intermediate states can vary for different reaction-delays. They are schematically depicted in figure 3.0b, where the states are given by the values of \( s_0, s_1, x, \) and \( z \). The arrows (in fig. 3.0b) lead to successor-states under \( \delta \), which are the first components of the elements of next\( f \)-\( \delta \) (the reaction-delay counter is hidden). Vacuous transitions are filtered out. The starting state \( \gamma_s \) has only one successor-state under \( \delta \) (ignoring differences of the reaction-delay counter, i.e., \( #_{(\Pi, rc)} : nextf\)-\( \delta \)-\( (\gamma_s, 0) \)-(\( \Pi, rc \)) : \( \Pi \} = 1 ; \) use def. 3.1). Depending on whether the reaction-delay of \( s_0 \) is less than, equal to, or greater than the reaction-delay of \( s_1 \), the intermediate states are given in the leftmost, middle, or rightmost path of states respectively. In order to avoid the transient conflict in the leftmost path, the reaction-delay of \( s_0 \) should be at least the reaction-delay of \( s_1 \).

We return to this example later in this chapter.

![Figure 3.0a](image)

As a first step in analysing how reaction-delay restrictions can be modelled, we rewrite restriction R2.2 (section 3.0) as

\[
R2.2' \quad (\text{E } rd : rd \in DC : (A \ s : \neg \text{consistent} \ \Gamma_1 : Q_1 \ s \land (rc_0 \ s = 0) : rc_1 \ s = rd \ s))
\]

Expressing restrictions on reaction-delays is possible by restricting the domain of \( rd \) in expression R2.2' to a subset of \( DC \). For instance, in example 3.11 above we have analysed that, in order to avoid a transient conflict on \( z \) (with starting state \( \gamma_s \) and the source-connection is changed to \( \delta \)), it is necessary to restrict the domain to \( \{ rd : DC-\text{rd} \land rd \ s_0 \geq rd \ s_1 : \text{rd} \} \).

This modelling seems a good first step, but it allows only local restrictions, i.e., restrictions on the reaction-delay of switches that become inconsistent-0 in the same transition. We can easily overcome this disadvantage by using \( rd \) as a global function, that is, by regarding, for each function in the restricted domain, the transitions that are possible for that specific function.
This can be formalised as follows: we define \(\text{next}_{\text{rd}}\gamma\) by replacing \text{R2.2} in the definition of \(\text{next}\gamma\) by \text{R2.2} below, and then consider the behaviours for each \(\text{rd}\) in the definition of the set of feasible states.

\[
(\mathcal{A}, s : \neg\text{consistent}_0\Gamma_1Q_1 : s \land (\text{rd}_0 : s = 0) : \text{rd}_1 : s = \text{rd} : s)
\]

Before we formalise this, we will show in example 3.12 below that in some cases global restrictions are necessary.

### 3.12 Example

Consider the circuit depicted below. Let the circuit be in the stable state where input \(x\) is connected to \(\{H\}\), and the values of \(s_0, s_1, s_2, s_3, s_4, x, y_0, y_1, y_2\), and \(z\) are 0, 1, 1, 0, 0, 1, \(\{H\}\), \(\{L\}\), \(\{H\}\), and \(\{L\}\) respectively. If, starting in this stable state, \(x\) is changed to \(\{L\}\), transient undefinedness of gates \(y_0\) and \(y_1\) can be avoided by restricting the reaction-delays of \(s_0, s_1, s_2, \) and \(s_3\) by \(\text{rd} : s_0 = \text{rd} : s_1\) and \(\text{rd} : s_2 = \text{rd} : s_3\). Notice that this can be done by restricting local delay functions (see above). In order to avoid a transient conflict on \(z\), the reaction-delay of \(s_2\) must be at most the sum of the reaction-delays of \(s_0, s_2,\) and \(s_4\). Since this seems a reasonable assumption, we would like to be able to express it in our formalism. Switches \(s_0, s_2,\) and \(s_4\), however, become inconsistent in different transitions, and, hence, this requirement cannot be expressed by restricting local reaction-delay functions. It can be expressed if global reaction-delay functions are used, namely, by restricting the global delay function, \(\text{rd}\), by:

\[
\text{rd} : s_5 \leq \text{rd} : s_0 + \text{rd} : s_2 + \text{rd} : s_4
\]

The circuit of example 3.12:

![Circuit Diagram]

The next-state function for reaction-delay \(\text{rd}\) can be defined as follows.

### 3.13 Definition

For \(\text{rd} \in \text{DC}\), \(\text{next}_{\text{rd}}\gamma \in \text{NST} - \text{STOPDC} = \mathcal{A}(\text{STOPDC})\) is defined by:

\[
\text{next}_{\text{rd}}\gamma((Q_0, \Gamma_0, x_0), (Q_1, \Gamma_1, x_1)) =
\]

\[
(\mathcal{A} : s : \text{rd}_0 : s = 0 : \text{consistent}_0\Gamma_0 : Q_1 : s )
\]

\[
\land (\mathcal{A} : s : \text{rd}_0 : s > 0 : Q_1 : s = Q_0 : s )
\]

\[
\land (\Gamma_1 = R : \gamma : Q_1)
\]

\[
\land (\mathcal{A} : s : \text{consistent}_1\Gamma_1 : Q_1 : s : \text{rd}_1 : s = 0 )
\]

\[
\land (\mathcal{A} : s : \neg\text{consistent}_1\Gamma_1 : Q_1 : s : \text{rd}_0 : s = 0 : \text{rd}_1 : s = \text{rd} : s - 1 )
\]

\[
\land (\mathcal{A} : s : \neg\text{consistent}_1\Gamma_1 : Q_1 : s : \text{rd}_0 : s = 0 : \text{rd}_1 : s = \text{rd} : s )
\]
Notice that: \[ (\Lambda \Pi_r, rd : ST0 \cdot \Pi \land DC \cdot rc \land DC \cdot rd : c gd0 \cdot \Pi \Rightarrow (\#(nextl_{rd} \cdot \gamma \cdot (\Pi, rc)) = 1) ) \]
(use prop. 1.13 and def. 3.13).
Since \( cgd0 \) is a correctness criterion on states, this means that the next-state function defined above is deterministic for correct states (compare with prop. 1.22).

In the following definition set \( RD \), with \( RD \subseteq DC \), is the set of restricted reaction-delays.

3.14 Definition

For \( RD \subseteq DC \), \( feasible_{RD} \) and \( stable_{RD} \), both with type \( \text{NST} \rightarrow \text{ST0} \rightarrow \text{\$} \), and \( WM1_{RD} \in \text{CIR} \rightarrow \text{NST} \rightarrow \text{\$} \) are defined by:

\[
\begin{align*}
\text{feasible}_{RD} \cdot \gamma \cdot \Pi &= (B \text{ rd}, rc : RD \cdot rd \cdot (nextl_{rd} \cdot \gamma) \cdot (\Pi, rc) \cdot (\Pi, rc)) \\
\text{stable}_{RD} \cdot \gamma \cdot \Pi &= (B \text{ rd}, rc : RD \cdot rd \cdot (nextl_{rd} \cdot \gamma) \cdot (\Pi, rc) \cdot (\Pi, rc)) \\
WM1_{RD} \cdot C \cdot \gamma &= (\text{feasible}_{RD} \cdot \gamma \subseteq cgd0)
\end{align*}
\]

3.15 Remark

It is possible to model reaction-delay restrictions for upgoing and for downgoing switch-states separately. This can be done by considering two functions, say \( rd_{H} \) and \( rd_{L} \), instead of one function for delays, where \( rd_{H} \) models the reaction-delay of upgoing switch-states (from 0 to 1), and \( rd_{L} \) models the reaction-delay of downgoing switch-states. This can then be modelled by using the following restriction instead of 'R2.2':

\[ (A \ s : (\neg \text{consistent}(\Gamma_{1}, Q_{1} \cdot s) \land (rc_{0} \cdot s = 0) : (Q_{1} \cdot s \Rightarrow (rc_{1} \cdot s = rd_{H} \cdot s)) \land (Q_{1} \cdot s \Rightarrow (rc_{1} \cdot s = rd_{L} \cdot s))) \]

For instance, the reaction-delay restriction derived in ex. 3.11 to avoid a transient conflict on \( z \) in the transition from \( \gamma_{a} \) to \( \delta_{a} \), can now be expressed as: \( rd_{H} \cdot z_{0} \geq rd_{L} \cdot z_{1} \).

Before considering well-functioning, we investigate the consequences of delay restrictions for well-matchedness.

3.16 Property

a \[ (A RD_{\gamma} : RD \subseteq DC \land \text{NST} \cdot \gamma : stable_{RD} \cdot \gamma \subseteq feasible_{RD}) \]
b \[ (A rd, rc, \gamma, \Pi : DC \cdot rd \land DC \cdot rc \land \text{NST} \cdot \gamma \land ST0 \cdot \Pi : nextl_{rd} \cdot \gamma \cdot (\Pi, rc) \subseteq nextl_{rd} \cdot \gamma \cdot (\Pi, rc) ) \]
c \[ (A \gamma, \Pi, rc : \text{NST} \cdot \gamma \land ST0 \cdot \Pi \land DC \cdot rc : nextl_{rd} \cdot \gamma \cdot (\Pi, rc) = (\nu rd : DC \cdot rd : nextl_{rd} \cdot \gamma \cdot (\Pi, rc) ) ) \]

Property 3.16a follows directly from definition 3.14. Property 3.16b follows directly from definitions 3.2 \((nextl)\) and 3.13 \((nextl_{rd})\). Property 3.16c is proved below:

Proof of prop. 3.16c

Let \( \gamma \in \text{NST} \), \( \Pi \in ST0 \), \( rc \in DC \) and \( rc_{1} \in DC \). From definitions 3.2 \((nextl)\) and 3.13 \((nextl_{rd})\) follows:

\[ nextl_{rd} \cdot \gamma \cdot (\Pi, rc) \cdot (\Pi, rc_{1}) = nextl_{rd} \cdot \gamma \cdot (\Pi, rc) \cdot (\Pi, rc_{1}) \cdot (\Pi, rc_{1}) \]

Hence: \( nextl_{rd} \cdot \gamma \cdot (\Pi, rc) \subseteq (\nu rd : DC \cdot rd : nextl_{rd} \cdot \gamma \cdot (\Pi, rc) ) \).

Using property 3.16b completes the proof.
3.17 Lemma

\[(\text{A RD} : \text{RD} \subseteq \text{DC} \land \text{RD} \neq \emptyset : \text{stable1}_{\text{RD}} = \text{stable1})\]

Proof

Let RD be a non-empty subset of DC, and let \(\gamma \in \text{NST}\) and \(\Pi \in \text{STO}\). Then:

\[\text{stable1}_{\text{RD}} : \gamma : \Pi\]

\[= \{\text{def. 3.14 (stable1}_{\text{RD}})\}\]

\[(\text{B rd,rc : RD-rd : nextt}_{\text{rd}} \gamma : (\Pi,rc)-(\Pi,rc)\}\]

\[\Rightarrow \{\text{prop. 3.16b}\}\]

\[(\text{B rc : : nextt}_{\text{rc}} : (\Pi,rc)-(\Pi,rc)\}\]

\[= \{\text{def. 3.3 (stable1)}\}\]

\[\text{stable1}_{\gamma} : \Pi\]

Furthermore:

\[\text{stable1}_{\gamma} : \Pi\]

\[= \{\text{prop. 3.6a}\}\]

\[\text{nextt}_{\gamma} : (\Pi,0)-(\Pi,0)\]

\[= \{\text{def. 3.1 (nextt)} \text{ and def. 3.13 (nextt}_{\text{rd}})\}\]

\[(\text{A rd : DC-rd : nextt}_{\text{rd}} \gamma : (\Pi,0)-(\Pi,0)\}\]

\[\Rightarrow \{\text{calculus, using RD \subseteq DC \land RD \neq \emptyset}\}\]

\[(\text{B rd : RD-rd : nextt}_{\text{rd}} \gamma : (\Pi,0)-(\Pi,0)\}\]

\[\Rightarrow \{\text{def. 3.14 (stable1}_{\text{RD}})\}\]

\[\text{stable1}_{\text{RD}} : \gamma : \Pi\]

\[\Box\]

3.18 Property

a \(\text{(A} \gamma,\text{RD} : \text{NST},\gamma \land \text{RD} \subseteq \text{DC} \land \text{RD} \neq \emptyset : \text{feasible1}_{\text{RD}} : (\gamma,\Pi) \leq \text{feasible1 : } \gamma\)

b \(\text{(A} \gamma : \text{NST} : \gamma : \text{feasible0 : } \gamma = \text{feasible1 : } (\emptyset : \gamma)\)

The proof of property 3.18a is given below. Property 3.18b gives the relation between the feasible-0 states and the feasible-1 states with restricted delay. We will not use property 3.18b, and leave the proof for the reader.

Proof of property 3.18a

Let RD be a non-empty subset of DC, and let \(\gamma \in \text{NST}\) and \(\Pi \in \text{STO}\). Then:

\[\text{feasible1}_{\text{RD}} : \gamma : \Pi\]

\[= \{\text{def. 3.14 (feasible1}_{\text{RD}})\}\]

\[(\text{B rd,rc : RD-rd : (nextt}_{\text{rd}} \gamma^* : (\Pi,rc)-(\Pi,rc)\}\]

\[\Rightarrow \{\text{prop. 3.16b}\}\]

\[(\text{B rc : : (nextt}_{\text{rc}} \gamma^* : (\Pi,rc)-(\Pi,rc)\}\)
3.19 Theorem

\((A \text{ RD} : \text{RD} \subseteq \text{DC} \land \text{RD} \neq \emptyset : WM_{\text{RD}} = WM)\)

Proof

Let \(\text{RD}\) be a non-empty subset of \(\text{DC}\), and let \(C \in \text{CIR}\) and \(\gamma \in \text{NST}\). Then:

\[WM_{\text{RD}} \cdot C \cdot \gamma\]

\(= \) \{def. 3.14 \((WM_{\text{RD}})\)\}

\[\text{feasible}_{\text{RD}} \cdot \gamma \subseteq cgd0\]

\(\Rightarrow\) \{prop. 3.16a\}

\[\text{stable}_{\text{RD}} \cdot \gamma \subseteq cgd0\]

\(= \) \{lemma. 3.17, using \(\text{RD} \subseteq \text{DC} \land \text{RD} \neq \emptyset\)\}

\[\text{stable} \cdot \gamma \subseteq cgd0\]

\(= \) \{th. 3.8\}

\[WM_{\text{C}} \cdot \gamma\]

Furthermore:

\[WM_{\text{C}} \cdot \gamma\]

\(= \) \{def. 3.3 \((WMI)\)\}

\[\text{feasible} \cdot \gamma \subseteq cgd0\]

\(\Rightarrow\) \{prop. 3.18a\}

\[\text{feasible}_{\text{RD}} \cdot \gamma \subseteq cgd0\]

\(= \) \{def. 3.14 \((WM_{\text{RD}})\)\}

\[WM_{\text{RD}} \cdot C \cdot \gamma\]

\(\square\)

As a result of lemma 3.17, prop. 3.18a, and theorem 3.19, similar results as those formulated in theorem 3.10 hold for \(WM_{\text{RD}}\). Using theorems 3.19 and 3.9, we can conclude that, for all \(\text{RD}, C, \text{ and } \gamma\) such that \(\text{RD} \subseteq \text{DC} \land \text{RD} \neq \emptyset \land \text{CIR} \cdot C \land \text{NST} \cdot \gamma\), \(WM_{\text{RD}} = WM_0\) holds and that the set of resulting states, in case \(WM_{\text{RD}} \cdot C \cdot \gamma\) holds, is \(\{\gamma_s\}\). So well-matchedness for restricted delays does not lead to additional problems.

It is left to the reader to verify that these results also hold for the formalisation described in remark 3.15.

The next step is to regard well-functioning. This is to be done in the next section.
3.3 Remarks on well-functioning

As discussed in the introduction of the previous section, we are not only interested in initial behaviour of a circuit, but also in dynamic behaviour of a circuit (cf. also section 3.2). The latter is the behaviour of a circuit starting in a - known - state when the source-connection changes. In examples 3.11 and 3.12 we considered such cases, and we demonstrated the calculation of intermediate states and the use of reaction-delay restrictions to avoid 'undesirable' intermediate states. In this section we - informally - introduce the correctness criteria we use regarding dynamic behaviour, thus explaining when intermediate states are classified as 'undesirable', and we - briefly - consider the possibility to formalise the necessary notions regarding dynamic behaviour within the model presented in this chapter.

Correctness of dynamic behaviour

Let a circuit be given in a certain state and let the source-connection change. For correctness of the resulting (dynamic) behaviour we require that all intermediate states, including the starting state and the possible resulting states are completely gate defined and conflict-free. This means that we choose to classify transient undefinedness and transient conflicts as 'undesirable'. We return to this choice in section 4.2, where we will also consider alternative choices.

Furthermore, all state transitions leading from an intermediate state to a next intermediate state should be correct, i.e., inconsistent switches must have stable gate values (cf. def. 1.47 (cst0)).

Modelling well-functioning

Let the starting state of a circuit be extended with a reaction-delay counter to $\langle \Pi, rc \rangle$, let the new source-connection be $\delta$, and let $\text{RD}$ be the set of reaction-delays. The set of intermediate and resulting states can then be defined as $\{ \text{rd, i} : \text{RD} = \text{rd} \land \text{OESi} : (nextf_{\mu}0)^{t}(\Pi, rc) \}$. Note that for states that are stable w.r.t. the previous source-connection the corresponding reaction-delay counter is 0 (namely, all switches are consistent; use restriction R2.2 from section 3.0). Note also that in many cases 'undesirable' behaviour (see above) can be avoided by the use of a suitably chosen set $\text{RD}$ (cf. e.g. ex. 3.11).

Let us reconsider the circuit given in example 3.11 with the formalisation of reaction-delay restrictions described in remark 3.15. In 3.11 we analysed that, for starting state $\gamma_s$ and new source-connection $\delta$, the reaction-delay $rd$ must satisfy $rd_{H:Z0} \geq rd_{L:Z1}$ in order to avoid a transient conflict on node $z$. Similarly, for starting state $\delta_s$ and new source-connection $\gamma$, the reaction-delay $rd$ must satisfy $rd_{H:Z1} \geq rd_{L:Z0}$ in order to avoid a transient conflict on node $z$. If node $z$ is a gate node, a transient floating of node $z$ must also be avoided since it leads to a transient undefined switch. It can easily be verified (see fig. 3.0b) that this leads to the following restriction on reaction delay $rd : rd_{H:Z0} = rd_{L:Z1} \land rd_{H:Z1} = rd_{L:Z0}$. 


In the next chapter we will see that this restriction is too strong if charge storage node $z$ is taken into account (namely, if gate node $z$ can store charge, it is allowed to be — temporarily — non-driven). We can conclude that, although a formalisation of dynamic behaviour and of well-functioning is possible with the model presented in this chapter, this would lead to an insufficiently accurate description. We will therefore first extend the model with a notion of charge storage in the next chapter, and then return to well-functioning (section 4.2).

### 3.4 Concluding remarks on chapter 3

In this chapter we have extended the basic model in such a way that arbitrary — but positive and finite — reaction-delays can be captured, and restrictions on reaction-delays can be expressed.

**Modelling (restricted) reaction-delays**

Reaction-delays are modelled using a countdown function. The main advantage of countdown functions is that the finiteness of the delays is taken care of in the next-state function, as a result of which the delays can be hidden (abstracted from) in the notions of stability and feasibility of states. Furthermore, the countdown functions enable a simple modelling of delay restrictions.

Restrictions on reaction-delays are modelled as elements of $SW \rightarrow N$. Of course, it is possible to extend this function in such a way that it depends on aspects of the current state. For instance, in remark 3.15 is explained how to discriminate restrictions for upgoing and downgoing transitions.

Restrictions on delays can be used in two ways. Firstly, for a given set of delay restrictions, the correctness of dynamic behaviour can be verified, and, secondly, minimal delay restrictions can be derived that guarantee this correctness. For instance, in example 3.11, for starting state $\gamma_s$ and new source-connection $\delta$, it can be verified that the restriction $rd_{H-s_0} = rd_{I-s_1}$ guarantees absence of a transient conflict on node $z$, but we can derive that for absence of a transient conflict on node $z$ it is sufficient and necessary to require that $rd_{H-s_0} \geq rd_{I-s_1}$.

**Initial behaviour**

We have concentrated mainly on initial behaviour and on the correctness criteria regarding initial behaviour. The main result is that the introduction of reaction-delays does not change the notion of well-matchedness or the resulting states in case of well-matchedness. This is expressed in theorems 3.9, 3.10, and 3.19 and in the remark following theorem 3.19 (for comparison: see th. 1.39 and cor. 1.40). These results are listed below. Note that this also means that the results of chapter 2 w.r.t. $WM_0$ (theorems 2.13 and 2.14) also hold for $WM_1$. 
The list of results for initial behaviour:

For arbitrary reaction-delays, using theorems 3.9, 3.10, and 1.39:

\[ WMI = WM0 \]

\[ (A, C, \gamma; \text{CIR}\cdot C \land \text{NST}\cdot \gamma; WMI\cdot C\cdot \gamma \Rightarrow (\text{feasible}1\cdot \gamma = \text{feasible}0\cdot \gamma) ) \]

For each non-empty set of reaction-delays RD, using th. 3.9, 3.19, and the remark below 3.19:

\[ WMI_{RD} = WM0 \]

\[ (A, C, \gamma; \text{CIR}\cdot C \land \text{NST}\cdot \gamma; WMI_{RD}\cdot C\cdot \gamma \Rightarrow (\text{feasible}1_{RD}\cdot \gamma = \text{feasible}0\cdot \gamma) ) \]

As explained at the end of section 3.1, the complete correctness criterion for initial behaviour of a circuit C and a source connection \( \gamma \) is equal to the one derived in chapter 1 (cf. 1.49), viz.

\[ WM0\cdot C\cdot \gamma \land CF\cdot \gamma = . \]

**Dynamic behaviour**

We have informally introduced dynamic behaviour and well-functioning in sections 3.2 and 3.3. The formalisation of well-functioning will be given in the next chapter after the model is extended with a notion of capacitance.
CHAPTER 4   CHARGE STORAGE

In the model presented in the previous chapters, the capability of nodes or wires to store charge is not considered (restriction (d), section 0.3). This chapter handles 'charge storage' by means of a notion of capacitance. This notion of capacitance is used only for charge storage; the influence of capacitances on delays is not considered. The notions regarding initial behaviour, with as most important ones: well-matchedness, conflict-freeness, and correctness of state transitions, are redefined, and the relation to the previous versions (from chapters 1 and 3) is given. As announced in section 3.3, we will, after charge storage is included in our model, finally formalize, besides initial behaviour, dynamic behaviour of circuits. The modelling of resistances (restriction (e), section 0.3) is discussed briefly.

Chapter 4 is organised as follows.

In section 4.0 the capability of nodes to store charge is discussed. The aspects of charge storage that are important for the formalisation are analysed. Following the same line of reasoning as in chapters 1 and 3, the familiar notions from those chapters are redefined such that charge storage can be handled. For this purpose, several new notions are introduced. This section concentrates on initial behaviour of circuits. At the end of it a renewed version of well-matchedness, called $WM_2$, is defined.

In section 4.1 the properties of the redefined notions, and the relation to their counterparts from chapters 1 and 3, are investigated. The meaning, in the new model, of the additional correctness criteria for initial behaviour is discussed.

In section 4.2 the informal discussion from section 3.3 on dynamic behaviour is resumed. Correctness criteria regarding dynamic behaviour are formalised. The notion well-functioning, discussed informally in sections 0.2 and 3.3, is defined and illustrated by a number of examples. The model presented in sections 4.0, 4.1, and 4.2 abstracts from differences in capacitance-strength. In section 4.3 a generalisation of this model to multiple capacitance-strengths is discussed. The modelling of resistances, which can be done by using a similar notion of strength, is also discussed. Both topics are discussed informally, and not developed in detail. As usual, the final section (4.4) gives a summary and discussion of the major results.
4.0 Introduction to charge storage

Introduction
All circuit nodes have a certain capacitance. By means of this capacitance they are capable of storing charge. This means that if a node is isolated from sources, it can, by means of this capacitance, "remember" the source type it was last connected to. Such a node will retain its voltage value until it is either connected to a source or connected to a node with a stored value of a different type. In the first case, the source value will override the stored value. In the second case, the stored value of the node with the strongest capacitance will override the other one, or, if the capacitances of the nodes are equally strong, a 'conflict at capacitance level' will occur.

We will concentrate on the charge-storage aspect of capacitances and we will not consider the influence capacitances have on the delays.

The assumptions we make on capacitances, i.e., on stored charges, are the following.

(0a) Node capacitances are so large that a stored high or low value at the gate node can keep a switch in its state provided that the state of that switch corresponds to the stored value at its gate node. That is, a stored low value can control a conducting p-switch or a nonconducting n-switch, and, similarly, a stored high value can control a nonconducting p-switch or a conducting n-switch.

(0b) The control of a switch by a stored charge -- as is assumed to be (partially) possible in (0a) -- must be temporary.

(1a) The stored value at a node is determined by the previous states -- including the stored values -- of all nodes connected to it (including the node itself). This means, among other things, that charge sharing is possible. Conflicts at capacitance level are assumed to be harmless, that is, they are assumed not to damage the circuitry.

(1b) Node capacitances are so small that once a node is connected to a source, the source value overrides the stored value.

The purpose of stored charges is to preserve (part of) the current state of the circuit. Therefore assumption (0a) is made. For the same reason, we make no assumptions on the capability of stored charges to change the state of a switch if this state does not correspond to the stored value at the gate node; such a situation is assumed to be incorrect.

Before continuing the discussion, we illustrate these ideas in a simple example.
4.0 Example

Consider the circuit depicted in figure 4.0a: a cascade of two inverters. For simplicity, the permanent sources are not labelled. Node \( x \) is the input node.

Let \( x \) be connected to an H-source and, as a result, let the circuit be in the stable state \( \{(s_0,0),(s_1,1),(s_2,1),(s_3,0)\}, \{(x,\{\text{H}\}),(y,\{\text{L}\}),(z,\{\text{H}\})\} \) : state 0 in figure 4.0b.

Assume that the reaction delay is such that \( rd_{\text{H}1} > rd_{\text{H}2} \) and \( rd_{\text{L}1} > rd_{\text{H}2} \). Let \( x \) be changed into \( \{\text{L}\} \) : state 1 in figure 4.0b. Switch \( s_1 \) will become nonconducting before \( s_0 \) becomes conducting. In the temporary state where both \( s_0 \) and \( s_1 \) are nonconducting, state 2 in fig. 4.0b, node \( y \) is floating. Due to its capacitance, node \( y \) retains a low charge, which is sufficient to keep \( s_2 \) and \( s_3 \) from changing (cf. (0a) above). Once \( s_0 \) is conducting, the high source value overrides the low stored value: state 3, cf. (1b). The protocol is depicted in figure 4.0b, where state 4 is the resulting stable state.

\[
\begin{array}{cccccc}
\text{H} & \text{H} & x & s_0 & s_1 & y & s_2 & s_3 & z \\
\hline
0 & \{\text{H}\} & 0 & 1 & \{\text{L}\} & 1 & 0 & \{\text{H}\} \\
1 & \{\text{L}\} & 0 & 1 & \{\text{L}\} & 1 & 0 & \{\text{H}\} \\
2 & \{\text{L}\} & 0 & 0 & \text{low} & 1 & 0 & \{\text{H}\} \\
3 & \{\text{L}\} & 1 & 0 & \{\text{H}\} & 1 & 0 & \{\text{H}\} \\
4 & \{\text{L}\} & 1 & 0 & \{\text{H}\} & 0 & 1 & \{\text{L}\} \\
\end{array}
\]

figure 4.0a

figure 4.0b

Modelling charge storage

In order to simplify the modelling of charge storage using the model from the previous chapter as a framework, we will - as a start - not distinguish between different strengths of capacitances and assume that each circuit node has a capacitance. In section 4.3 we return to these restrictions and show how to avoid them.

In a state where the stored value of a gate controls - according to assumption (0a) - a switch, we say that the capacitance is active in that switch. That is (cf. (0a)), the capacitance is active in a switch \( s \) if the gate node has a stored low value and \( Q:s = \{t:s = \{\text{L}\}\} \), or if the gate node has a stored high value and \( Q:s = \{t:s = \{\text{H}\}\} \), where \( Q \) is the current switch-state.

For a state \( \Pi \) we denote that the capacitance is active in a switch \( s \) by \( \text{CA} \cdot \Pi \cdot s \). Assumption (0a) now means that a state transition from \( (Q_0,\Gamma_0) \) to \( (Q_1,\Gamma_1) \) must satisfy:

\[
\text{CA} \cdot (Q_0,\Gamma_0) \cdot s \Rightarrow (Q_1,s = Q_0,s)
\]

In order to formalise this notion we need some knowledge about the stored values, possibly incorporated in the notion 'state'. This knowledge cannot be expressed in terms of \( \mathcal{R}(\text{L},\text{H}) \), since that would lead to difficulties in distinguishing source values and stored values (note that making this distinction is necessary: see (1a) and (1b)). We therefore introduce symbols \( l \) and
denoting a stored low value and a stored high value respectively (compare with the weak and strong 0's and 1's in \([\text{BeK}]\). Nodes that are not connected to sources can have values \(\{l, h\}\), \(\{l\}\), \(\{h\}\), or \(\emptyset\) denoting 'a conflict at capacitance level', 'a stored low value', 'a stored high value', and 'uncharged' respectively. This means we extend node-states by subdividing the class of floating states.

4.1 Definition

Sets \(\text{nst}\) and \(\text{NSTC}\) are defined by:

\[
\text{nst} = N \setminus \mathcal{R}(\{l,h\})
\]
\[
\text{NSTC} = N \setminus (\mathcal{R}(\{L,H\}) \cup \mathcal{R}(\{l,h\}))
\]
\(\text{NSTC}\) is the new set of node-states.

The order \(\preceq\) on \(\mathcal{R}(\{L,H\}) \cup \mathcal{R}(\{l,h\})\) is defined by the Hasse diagram depicted above. The order \(\preceq\) on \(\text{NST}\) is extended to \(\text{NSTC}\) as the componentwise order \(\preceq\), that is, for \(\Gamma, \Delta \in \text{NSTC}\):

\[
\Gamma \preceq \Delta \iff (A_x : N_x : \Gamma_x \preceq \Delta_x).
\]

Note that \(\text{NST} \subseteq \text{NSTC}\), and that both \(\mathcal{R}(\{L,H\}) \cup \mathcal{R}(\{l,h\})\), \(\preceq\) and \((\text{NSTC}, \preceq)\) are lattices. Their least upper bound operators are both denoted by \(\cup\). Notice that, for \(\Gamma, \Delta \in \text{NSTC}\):

\[
(A_x : N_x : (\Gamma \cup \Delta)_x = \Gamma_x \cup \Delta_x).
\]

The chosen order, that is, stored values are smaller than source values, expresses part of condition (1b). Conditions (1a) and (1b) will be expressed in the notion 'state transition', which is discussed later on in this section. Before we discuss 'state transitions', we introduce some additional notions, extend some old ones, and formalise \(\text{CA}\). First of all, as a consequence of the new definition of node-states, we redefine the set of circuit states.

4.2 Definition

The set of circuit states \(\text{ST1}\) is defined by:

\[
\text{ST1} = \text{SST} \times \text{NSTC}
\]

The order \(\preceq\) on \(\text{ST1}\) is defined by:

\[
(Q_0, \Gamma_0) \preceq (Q_0, \Gamma_1) \iff (Q_0 \preceq Q_1 \land \Gamma_0 \preceq \Gamma_1)
\]

Typical names used for elements of \(\text{NSTC}\) are \(\Gamma\) or \(\Delta\), and for elements of \(\text{ST1}\) : \(\Omega\), \(\Pi\).

We can now formalise when the capacitance is active in a switch:

4.3 Definition

\(\text{CA} \in \text{ST1} \rightarrow \text{SW} \rightarrow \mathcal{B}\) is defined by:

\[
\text{CA} \cdot (Q, \Gamma)_\mathcal{S} = \left( \left( \Gamma \cdot (g.s) = \{l\} \land Q \cdot s = \{l, s = \{L\}\} \right) \lor \left( \Gamma \cdot (g.s) = \{h\} \land Q \cdot s = \{l, s = \{H\}\} \right) \right)
\]
4.4 Definition
store ∈ NSTC → nst is defined by:
\( \{ l \in \text{store} \cdot \Gamma \cdot x \} = (\{ l \in \Gamma \cdot x \} \cup \{ L \in \Gamma \cdot x \}) \)
\( \{ h \in \text{store} \cdot \Gamma \cdot x \} = (\{ h \in \Gamma \cdot x \} \cup \{ H \in \Gamma \cdot x \}) \)
destore ∈ NSTC → NST is defined by:
destore \cdot \Gamma \cdot x = (\Gamma \cdot x \cap \{ L, H \})
destore is extended to \( \text{ST1} \rightarrow \text{ST0} \) by:
destore \cdot (Q, \Gamma) = (Q, \text{destore} \cdot \Gamma)

Function store records the stored information about a (previous) state; it will be used to define
the state transitions between elements of ST1. Function destore abstracts from the stored
values; it is used to define extensions of well-known notions, and − in the next section − to
relate the notions from this chapter to the previously defined notions.
In order to define state transitions in a similar way as in the previous chapter, we need to extend
the notions consistent0, coco0, and the response function \( R \) (from definitions 1.12 and 1.17).

4.5 Definition
consistent0 is extended to NSTC → SST → SW → \( \mathbb{B} \) by:
\( \text{consistent0} \cdot \Gamma = \text{consistent0} \cdot (\text{destore} \cdot \Gamma) \)
coco0 is extended to NSTC → SST → \( \mathbb{B} \) by:
\( \text{coco0} \cdot \Gamma = \text{coco0} \cdot (\text{destore} \cdot \Gamma) \)

It is left to the reader to verify that the extensions of notions we define are conservative
extensions, which means that equality of both definitions holds on the old domain. For def. 4.5
this is easily seen, since \( \text{destore} \cdot \Gamma = \Gamma \) for \( \Gamma \in \text{NST} \).

4.6 Definition
The response function \( R \) is extended to NSTC → SST → NSTC by:
\( R \cdot \Delta \cdot Q \cdot x = (y : cp \cdot Q \cdot [x, y] : \Delta y) \)

State transitions
As explained before, assumption (Qa) means that a state transition − in ST1 from \( (Q_0, \Gamma_0) \) to
\( (Q_1, \Gamma_1) \) must satisfy, for \( s \in \text{SW} : \) CA \( (Q_0, \Gamma_0) \cdot s \Rightarrow (Q_0 \cdot s = Q_1 \cdot s) \).
This leads, using restrictions R0.0 and R0.1 from chapter 3, which define the next switch state
in the model of chapter 3, to their revised versions R0.2 and R0.3 below.
Notice that, since CA \( (Q_0, \Gamma_0) \cdot s \Rightarrow \text{consistent0} \cdot \Gamma_0 \cdot Q_0 \cdot s \) (cf. prop. 4.19a,b), the behaviour defined
by R0.2 and R0.3 is a restriction of the behaviour defined by R0.0 and R0.1 (cf. lemmas 4.21 and 4.23).

In order to calculate the next node-state, the stored values of the previous node-state must be taken into account. This means that we need to consider, besides the connections to sources (as expressed by \( \gamma \)), the connections to the stored values of the previous state (as expressed by store-\( \Gamma_0 \)). This can be formulated as \( \Gamma_1 = R \cdot (\gamma \cup \text{store-}\Gamma_0) \cdot Q_1 \), which captures both assumptions (1a) and (1b) since the stored values are smaller than the source values (def. 4.1). This restriction is expressed in R1.1, which replaces R1.0 from chapter 3 and defines the next node-state.

Assumption (0b) will be dealt with by an additional correctness criterion on feasible states requiring finiteness of capacitance. We define this criterion at the end of this section. Notice the difference between finiteness of capacitances and finiteness of delays. Infinite capacitances correspond to incorrect physical behaviour and must, therefore, be dealt with by a correctness criterion. Whereas infinite delays do not correspond to physical behaviour, and, hence, must not be included in modelled behaviour.

The restrictions R2.1, which define the next reaction delay counter, need not be changed.

As explained above, a state transition from \((Q_0,\Gamma_0),rc_0\) to \((Q_1,\Gamma_1),rc_1\) must satisfy:

- **R0.2**: \( A \cdot s \cdot (rc_0 \cdot s = 0) \land \neg \text{CA} \cdot (Q_0,\Gamma_0) \cdot s \cdot \text{consistent-} \Gamma_0 \cdot Q_1 \cdot s \)
- **R0.3**: \( A \cdot s \cdot (rc_0 \cdot s > 0) \lor \text{CA} \cdot (Q_0,\Gamma_0) \cdot s \cdot Q_1 \cdot s = Q_0 \cdot s \)
- **R1.1**: \( \Gamma_1 = R \cdot (\gamma \cup \text{store-}\Gamma_0) \cdot Q_1 \)
- **R2.0**: \( A \cdot s \cdot \text{consistent-} \Gamma_1 \cdot Q_1 \cdot s \land rc_1 \cdot s = 0 \)
- **R2.1**: \( A \cdot s \cdot \neg \text{consistent-} \Gamma_1 \cdot Q_1 \cdot s \land (rc_0 \cdot s > 0) \land rc_1 \cdot s = rc_0 \cdot s \land 1 \)
- **R2.2**: \( A \cdot s \cdot \neg \text{consistent-} \Gamma_1 \cdot Q_1 \cdot s \land (rc_0 \cdot s = 0) \land rc_1 \cdot s \in \mathbb{N} \)

Similar to def. 3.1 and def. 3.2 in chapter 3, we can now define next2, stable2, and feasible2.

### 4.7 Definition

**next2** \( \in NST \times (ST\times DC) \times R(ST\times DC) \) is defined by:

\[ \text{next2}\cdot\gamma\cdot((Q_0,\Gamma_0),rc_0);((Q_1,\Gamma_1),rc_1) = \]

\[ (A \cdot s \cdot (rc_0 \cdot s = 0) \land \neg \text{CA} \cdot (Q_0,\Gamma_0) \cdot s \cdot \text{consistent-} \Gamma_0 \cdot Q_1 \cdot s) \]
\[ \lor (A \cdot s \cdot (rc_0 \cdot s > 0) \lor \text{CA} \cdot (Q_0,\Gamma_0) \cdot s \cdot Q_1 \cdot s = Q_0 \cdot s) \]
\[ \lor (\Gamma_1 = R \cdot (\gamma \cup \text{store-}\Gamma_0) \cdot Q_1) \]
\[ \lor (A \cdot s \cdot \text{consistent-} \Gamma_1 \cdot Q_1 \cdot s \land rc_1 \cdot s = 0) \]
\[ \lor (A \cdot s \cdot \neg \text{consistent-} \Gamma_1 \cdot Q_1 \cdot s \land (rc_0 \cdot s > 0) \land rc_1 \cdot s = rc_0 \cdot s \land 1) \]
\[ \lor (A \cdot s \cdot \neg \text{consistent-} \Gamma_1 \cdot Q_1 \cdot s \land (rc_0 \cdot s = 0) \land rc_1 \cdot s \in \mathbb{N}) \) \]
4.8 Definition

stable2 and feasible2, both with type \( \text{NST} \rightarrow \text{ST1} \rightarrow \text{B} \), are defined by:

\[
\begin{align*}
stable2 \cdot \gamma \cdot \Pi &= (\mathbb{E} \text{rc} : \text{DC} \cdot \text{rc} \cdot \text{next2} \cdot \gamma \cdot (\Pi, \text{rc}) \cdot (\Pi, \text{rc}) ) \\
feasible2 \cdot \gamma \cdot \Pi &= (\mathbb{E} \text{rc} : \text{DC} \cdot \text{rc} \cdot (\text{next2} \cdot \gamma^* \cdot (\Pi, \text{rc}) \cdot (\Pi, \text{rc}) )
\end{align*}
\]

Well-matchedness:
The correctness criterion expressed in well-matchedness needs to be revised. First of all, a switch is not only gate-defined if its gate is high or low, but also if the capacitance is active in that switch (cf. (0a) and def. 4.3 (CA)). The revised version of gate-definedness is called gate-defined-1 \((\text{gd}1)\) and is defined below (def. 4.10). The previous notion of gate-definedness \((\text{gd}0)\) is extended in definition 4.9 below.

4.9 Definition

\( \text{gd}0 \) is extended to \( \text{NSTC} \rightarrow \text{SW} \rightarrow \text{B} \) by:

\[ \text{gd}0 \cdot \Gamma = \text{gd}0 \cdot (\text{destore} \cdot \Gamma) \]

\( \text{cg} \text{gd}0 \) is extended to \( \text{ST}1 \rightarrow \text{B} \) by:

\[ \text{cg} \text{gd}0 \cdot \Pi = \text{cg} \text{gd}0 \cdot (\text{destore} \cdot \Pi) \]

4.10 Definition

\( \text{gd}1 \in \text{ST}1 \rightarrow \text{SW} \rightarrow \text{B} \) and \( \text{cg} \text{gd}1 \in \text{ST}1 \rightarrow \text{B} \) are defined by:

\[ \text{gd}1 \cdot (\text{Q}, \Gamma) \cdot s = (\text{gd}0 \cdot \Gamma \cdot s \lor \text{CA} \cdot (\text{Q}, \Gamma) \cdot s) \]

\[ \text{cg} \text{gd}1 \cdot \Pi = (A \cdot s : \text{SW} \cdot s : \text{gd}1 \cdot \Pi \cdot s) \]

According to assumption (0b), the control of a switch by a stored value — as is assumed to be (partially) possible in assumption (0a) — must be temporary. As a second correctness criterion we therefore require that a switch is not controlled by a stored value in every state of a cycle of feasible states. Well-matchedness can then be defined as (informally): every cycle of feasible states is completely gate defined-1 in all elements and satisfies the ‘finite capacitance-control requirement’ described above.

In order to define this second correctness criterion, we need to define such a cycle of feasible states. This is done in the notion Feasible State List 2 \((\text{FSL}2)\), which is defined below using the list notions defined in section 0.5. A state list \( \Psi \) is a cycle of feasible states if all successors in \( \Psi \) are, combined with a suitably chosen reaction-delay counter, successors w.r.t. \( \text{next2} \cdot \gamma \);

\( \text{FSL}2 \cdot \gamma ^* \cdot \Psi \) denotes that \( \Psi \) is a cycle of feasible states.

4.11 Definition  Feasible state list 2

\( \text{FSL}2 \in \text{NST} \rightarrow \text{L} \cdot (\text{ST}1) \rightarrow \text{B} \) is defined by:

\[
\begin{align*}
\text{FSL}2 \cdot \gamma \cdot \Psi &= ( \mathbb{E} \ r_l : r_l \in \text{L} \cdot (\text{DC}) \\
&\quad \cdot (\# \cdot r_l = \# \cdot \Psi) \land (A \cdot i : 0 \leq i < \# \cdot \Psi : \text{next2} \cdot \gamma \cdot (\Psi, r_l) \cdot (\Psi, r_{i+1}, r_{i+1}) )
\end{align*}
\]

where \( \Psi_i \) and \( r_{l_i} \) for \( i = \# \cdot \Psi \) are defined as \( \Psi_0 \) and \( r_{l_0} \).
4.12 Lemma

\((A \gamma : NST \cdot \gamma : (\nu \Psi : FSL2 \cdot \gamma \cdot \Psi : \{ i : 0 \leq i < \# \cdot \gamma \cdot \Psi_1 \})) = \text{feasible2} \cdot \gamma\)

Proof

Let \(\gamma \in \text{NST}\). Then:

\[
\Pi \in (\nu \Psi : FSL2 \cdot \gamma \cdot \Psi : \{ i : 0 \leq i < \# \cdot \gamma \cdot \Psi_1 \})
\]

= \{calculus\}

\[
(\nu \Psi, i : FSL2 \cdot \gamma \cdot \Psi \land (0 \leq i < \# \cdot \gamma \cdot \Psi) : \Pi = \Psi_1)
\]

= \{introduction dummy\}

\[
(\nu \Psi, m, i : FSL2 \cdot \gamma \cdot \Psi \land (m = \# \cdot \gamma \cdot \Psi) \land (0 \leq i < m) : \Pi = \Psi_1)
\]

= \{def. 4.11 (FSL2)\}

\[
(\nu \Psi, r, l, m : L^\text{ST1} \cdot \Psi \land L^\text{DC} \cdot r \land (m = \# \cdot \Psi) \land (0 \leq i < m)
\]

\[
: (\Pi = \Psi_j) \land (\# \cdot r = m)
\]

\[
\land (A j : 0 \leq i < m : \text{next2} \cdot \gamma \cdot (\Psi_{j+i}, r, l) \cdot (\Psi_j \mod m \land r_j \mod m))
\]

= \{renaming\}

\[
(\nu \Psi, r, l, m : L^\text{ST1} \cdot \Psi \land L^\text{DC} \cdot r \land (m = \# \cdot \Psi)
\]

\[
: (\Pi = \Psi_0) \land (\# \cdot r = m)
\]

\[
\land (A j : 0 \leq i < m : \text{next2} \cdot \gamma \cdot (\Psi_{j+i}, r, l) \cdot (\Psi_j \mod m \land r_j \mod m))
\]

= \{calculus\}

\[
(\nu \Psi, r, l, m : \text{DC-rec} \land (1 \leq m) : (\text{next2} \cdot \gamma) \cdot (\Pi_{r,l}) \cdot (\Pi_{r,l})
\]

= \{def. (O)\}

\[
(\nu \Psi, r, l : \text{DC-rec} \cdot (\text{next2} \cdot \gamma) \cdot (\Pi_{r,l}) \cdot (\Pi_{r,l})
\]

= \{def. 4.8 (feasible2)\}

\[
\text{feasible2} \cdot \gamma \Pi
\]

\(\square\)

The additional requirement needed for finiteness of capacitance-control, for state lists in

\(FSL2 \cdot \gamma\), can be formulated as follows:

4.13 Definition {finite capacitance-control for finite state lists}

\(\text{FCA0} \in L^\text{ST1} \rightarrow \nu \Psi\) is defined by:

\[
\text{FCA0} \cdot \Psi = (A s : \text{SW} \cdot s : (B i : 0 \leq i < \# \cdot \Psi \land \neg \text{CA} \cdot \Psi_1 \cdot s))
\]

To be able to give an elegant definition for well-matchedness it is convenient to extend \(cgdi\) as follows.

4.14 Definition \(\text{cgdi}\) for lists of states

\(\text{cgdi} \cdot L \in (L^\text{ST1} \cup L^\text{DC}) \rightarrow \nu \Psi\) is defined by:

\[
\text{cgdi} \cdot L \cdot \Psi = (A i : 0 \leq i < \# \cdot \Psi \land \neg \text{cgdi} \cdot \Psi_1)
\]
As a definition for well-matchedness we then have argued:

4.15 Definition
\[ WM2 \in CIR \rightarrow NST \rightarrow \emptyset \text{ is defined by:} \]
\[ WM2.C.\gamma = (\text{FS}L2.\gamma \subseteq (\text{FCA}_0 \cap \text{cgsdl})) \]

In the next section we investigate properties of the defined notions and the relation between them and the notions from the previous chapter. The two other correctness criteria for initial behaviour (\textit{CF} and \textit{cstb}; def. 1.46 and 1.47) are considered at the end of the next section.

4.1 Properties of WM2

This section is organised as follows. In the first part (preliminaries) the notion \textit{cocol} is extended and a number of properties of the newly defined notions are given. In the second part (relation to chapter 3) the relation between \textit{next2} and \textit{next1}, \textit{stable2} and \textit{stable1}, and \textit{feasible2} and \textit{feasible1} are investigated. The third part of this section gives properties of WM2. Finally, the major results regarding WM2 are discussed, and the two additional correctness criteria for initial behaviour (\textit{CF} and \textit{cstb}; def. 1.46 and 1.47) are reconsidered.

Preliminaries

The notions \textit{consistent1} and \textit{cocol} are extended to the new set of node states (cf. def. 1.15).

4.16 Definition
\textit{consistent1} is extended to NSTC \rightarrow SST \rightarrow SW \rightarrow \emptyset by:
\[ \text{consistent1} \Gamma .Q .s = (Q .s \otimes (A x .y : px .s = [x .y] : \Gamma .x = \Gamma .y )) \]
\textit{cocol} is extended to NSTC \rightarrow SST \rightarrow \emptyset by:
\[ \text{cocol} \Gamma .Q = (A s : SW .s : \text{consistent1} \Gamma .Q .s) \]

The remainder of this subsection gives a number of properties that are convenient for relating this model to the model from chapter 3 (next subsections). In 4.17 and 4.19b,c some properties from chapter 1, which also hold for the extended versions of notions, are restated (viz. (2) above 1.15 in cf. 4.17a; 1.18a in 4.17c; 1.13a in 4.19b; and 1.12 in 4.19c). Some properties of the new notions are given in 4.18, 4.19a, and 4.20.
4.17 Property
For $\Delta \in \text{NSTC}$ and $Q \in \text{SST}$:

a. $\text{cocol-}\Delta\cdot Q = (A : x, y : \text{bcp-}Q[x, y] : \Delta \cdot x = \Delta \cdot y)$
b. $\text{cocol-}\Delta\cdot Q = (A : x, y : \text{cp-}Q[x, y] : \Delta \cdot x = \Delta \cdot y)$
c. $\text{cocol-}(R \cdot \Delta\cdot Q) \cdot Q$

Property 4.17a follows from def. 4.16 and def. 1.14 (bcp). Property 4.17b follows from prop. 4.17a and def. 1.14 (cp). Property 4.17c follows from prop. 4.17b, def. 4.6 (R), and transitivity of cp- Q (from def. 1.14 (cp)).

4.18 Property
For $\Gamma \in \text{NSTC}$, $\Delta \in \text{NSTC}$, $x \in \text{N}$, and $y \in \text{N}$:

a. $(\Gamma \uplus \Delta) \cdot x = \Gamma \cdot x \uplus \Delta \cdot x$
b. $(\Gamma \cdot x \uplus \Delta \cdot y) \cap \{L, H\} = (\Gamma \cdot x \cap \{L, H\}) \uplus (\Delta \cdot y \cap \{L, H\})$
c. destore-(\Gamma \uplus \Delta) = destore-\Gamma \uplus destore-\Delta$
d. $\Delta \in \text{nst} \Rightarrow (\text{destore-} \Delta = 1)$

Properties 4.18a and 4.18b follow from def. 4.1 and the definition of \(\uplus\) (below 4.1). Prop. 4.18c follows from def. 4.4 (destore) and prop. 4.18a,b. Prop. 4.18d follows from def. 4.4 (destore).

4.19 Property
For $(Q, \Gamma) \in \text{ST}$ and $s \in \text{SW}$:

a. $\text{ca-(}Q, \Gamma) \cdot s \Leftrightarrow \neg \partial \partial \Gamma \cdot s$
b. $\neg \partial \partial \Gamma \cdot s \Leftrightarrow \text{consistent0-1} \cdot Q \cdot s$
c. $\text{cgc-}(Q, \Gamma) = (A : s : \text{SW-}s : \partial \partial \Gamma \cdot s)$

Property 4.19a follows from definitions 4.3 (CA) and 4.9 (gd0). Prop. 4.19b follows from def. 4.5 (consistent0) and 4.9 (gd0). Prop. 4.19c follows directly from def. 4.9 ((c)gd0).

4.20 Lemma
a. For $\Delta \in \text{NSTC}$ and $Q \in \text{SST}$: $\text{destore-}(R \cdot \Delta\cdot Q) = R \cdot \text{destore-}\Delta \cdot Q$

b. For $\Gamma \in \text{NSTC}$: $\Gamma = \text{destore-} \Gamma \uplus \text{store-} \Gamma$

Proof of 4.20a
Let $x \in \text{N}$. Then:

$\text{destore-}(R \cdot \Delta \cdot Q) \cdot x$

= \{def. 4.4 (destore)}

$R \cdot \Delta \cdot Q \cdot x \cap \{L, H\}$

= \{def. 4.6 (R)}

$(\emptyset \cdot y : \text{cp-}Q[x, y] : \Delta \cdot y) \cap \{L, H\}$
\[\begin{align*}
= \{\text{prop. 4.18b}\} \\
(u, y : \text{cp-Q} \cdot \{x, y\} : \Delta \cdot \gamma \cap \{L, H\}) \\
= \{\text{def. 4.4 (dstore)}\} \\
(u, y : \text{cp-Q} \cdot \{x, y\} : \text{dstore} \cdot \Delta \cdot \gamma) \\
= \{\text{def. 4.6 (R)}\} \\
R \cdot (\text{dstore} \cdot \Delta) \cdot Q \cdot x
\end{align*}\]

Proof of 4.20b

Let \( x \in N \). Then:
\[\Gamma \cdot x \cap \{L, H\} = \emptyset\]
\[\Rightarrow \{\text{use def. 4.4 ((de)store)}\} \\
\text{dstore} \cdot \Gamma \cdot x = \emptyset \land \text{store} \cdot \Gamma \cdot x = \Gamma \cdot x\]
\[\Rightarrow \{\text{use definition of \( \varepsilon \)}\} \\
\Gamma \cdot x = \text{dstore} \cdot \Gamma \cdot x \equiv \text{store} \cdot \Gamma \cdot x\]

Furthermore:
\[\Gamma \cdot x \cap \{L, H\} \neq \emptyset\]
\[\Rightarrow \{\text{use def. 4.1 : range of \( \Gamma \)}\} \\
\Gamma \cdot x = \Gamma \cdot x \cap \{L, H\}\]
\[= \{\text{def. 4.4 ((de)store)}\} \\
\Gamma \cdot x = \text{dstore} \cdot \Gamma \cdot x \land (\text{store} \cdot \Gamma \cdot x \subseteq \Gamma \cdot x)\]
\[\Rightarrow \{\text{use definition of \( \varepsilon \)}\} \\
\Gamma \cdot x = \text{dstore} \cdot \Gamma \cdot x \equiv \text{store} \cdot \Gamma \cdot x\]

Using property 4.18a completes the proof.

\[\square\]

Relation to chapter 3

The relations between next2 and next1, stable2 and stable1, and feasible2 and feasible1 are investigated in the following three lemmas.

4.21 Lemma

Let, for \( 0 \leq i < 2 \), \( \Pi_i \in \text{ST}1 \) and \( r_c \in DC \). Then:
\[\text{next2} \cdot \gamma \cdot (\Pi_0, r_c) \cdot \gamma (\Pi_1, r_c) = \text{next1} \cdot \gamma \cdot (\text{dstore} \cdot \Pi_0, r_c) \cdot (\text{dstore} \cdot \Pi_1, r_c)\]

Proof

Let, for \( 0 \leq i < 2 \), \( (\Gamma_i, Q_i) \in \text{ST}1 \) and \( r_c \in DC \). Then:
\[\text{next2} \cdot \gamma \cdot ((Q_0, \Gamma_0), r_c) \cdot ((Q_1, \Gamma_1), r_c)\]
\[= \{\text{def. 4.7 (next2)}\} \]
\((A \vdash (rc_0 \cdot s = 0) \land \neg \text{CA} \cdot (Q_0, \Gamma_0) \cdot s : \text{consistent0} \cdot \Gamma_0 \cdot Q_1 \cdot s)\)

\(\land (A \vdash (rc_0 \cdot s > 0) \lor \text{CA} \cdot (Q_0, \Gamma_0) \cdot s : Q_1 \cdot s = Q_0 \cdot s)\)

\(\land (\Gamma_1 = R \cdot (\gamma \triangleright \text{store} \cdot \Gamma_0) \cdot Q_1)\)

\(\land (A \vdash \text{consistent0} \cdot \Gamma_1 \cdot Q_1 \cdot s : \text{rc}_1 \cdot s = 0)\)

\(\land (A \vdash \neg \text{consistent0} \cdot \Gamma_1 \cdot Q_1 \cdot s \land (\text{rc}_0 \cdot s > 0) : \text{rc}_1 \cdot s = \text{rc}_0 \cdot s - 1)\)

\(\land (A \vdash \neg \text{consistent0} \cdot \Gamma_1 \cdot Q_1 \cdot s \land (\text{rc}_0 \cdot s = 0) : \text{rc}_1 \cdot s \in \mathbb{N})\)

\[\{\text{note 0, note 1, and def. 4.5 (consistent0)}\}\]

\(\land (A \vdash (\text{destore} \cdot (Q_0, \Gamma_0), \text{rc}_0) \cdot (\text{destore} \cdot (Q_1, \Gamma_1), \text{rc}_1))\)

\(\land (A \vdash (\text{destore} \cdot \Gamma_1 = R \cdot \gamma \cdot Q_1)\)

\(\land (A \vdash \text{consistent0} \cdot \text{(destore} \cdot \Gamma_1, \text{destore} \cdot Q_1) \cdot s : \text{rc}_1 \cdot s = 0)\)

\(\land (A \vdash \text{consistent0} \cdot \text{(destore} \cdot \Gamma_1) \cdot Q_1 \cdot s \land (\text{rc}_0 \cdot s > 0) : \text{rc}_1 \cdot s = \text{rc}_0 \cdot s - 1)\)

\(\land (A \vdash \text{consistent0} \cdot \text{(destore} \cdot \Gamma_1) \cdot Q_1 \cdot s \land (\text{rc}_0 \cdot s = 0) : \text{rc}_1 \cdot s \in \mathbb{N})\)

\[\{\text{definitions 4.7 (next2) and 4.4 (extension destore)}\}\]

next2 \(\gamma \cdot \text{(destore} \cdot (Q_0, \Gamma_0), \text{rc}_0) \cdot \text{(destore} \cdot (Q_1, \Gamma_1), \text{rc}_1)\)

\[\{\text{note 0}\}\]

\(\land (A \vdash (\text{rc}_0 \cdot s = 0) \land \neg \text{CA} \cdot (Q_0, \Gamma_0) \cdot s : \text{consistent0} \cdot \Gamma_0 \cdot Q_1 \cdot s)\)

\(\land (A \vdash (\text{rc}_0 \cdot s > 0) \lor \text{CA} \cdot (Q_0, \Gamma_0) \cdot s : Q_1 \cdot s = Q_0 \cdot s)\)

\[=\{\text{calculus}\}\]

\(\land (A \vdash (\text{rc}_0 \cdot s = 0) \land \neg \text{CA} \cdot (Q_0, \Gamma_0) \cdot s : \text{consistent0} \cdot \Gamma_0 \cdot Q_1 \cdot s)\)

\(\land (A \vdash (\text{rc}_0 \cdot s = 0) \land \text{CA} \cdot (Q_0, \Gamma_0) \cdot s : Q_1 \cdot s = Q_0 \cdot s)\)

\(\land (A \vdash (\text{rc}_0 \cdot s > 0) : Q_1 \cdot s = Q_0 \cdot s)\)

\[\{\text{properties 4.19a and 4.19b}\}\]

\(\land (A \vdash (\text{rc}_0 \cdot s = 0) \land \neg \text{CA} \cdot (Q_0, \Gamma_0) \cdot s : \text{consistent0} \cdot \Gamma_0 \cdot Q_1 \cdot s)\)

\(\land (A \vdash (\text{rc}_0 \cdot s = 0) \land \text{CA} \cdot (Q_0, \Gamma_0) \cdot s : \text{consistent0} \cdot \Gamma_0 \cdot Q_1 \cdot s)\)

\(\land (A \vdash (\text{rc}_0 \cdot s > 0) : Q_1 \cdot s = Q_0 \cdot s)\)

\[=\{\text{calculus}\}\]

\(\land (A \vdash \text{rc}_0 \cdot s = 0 : \text{consistent0} \cdot \Gamma_0 \cdot Q_1 \cdot s)\)

\(\land (A \vdash \text{rc}_0 \cdot s > 0 : Q_1 \cdot s = Q_0 \cdot s)\)

\[\{\text{note 1}\}\]

\(\Gamma_1 = R \cdot (\gamma \triangleright \text{store} \cdot \Gamma_0) \cdot Q_1\)

\[\{\text{calculus}\}\]

\(\text{destore} \cdot \Gamma_1 = \text{destore} \cdot (R \cdot (\gamma \triangleright \text{store} \cdot \Gamma_0) \cdot Q_1)\)

\[\{\text{lemma 4.20a}\}\]

\(\text{destore} \cdot \Gamma_1 = R \cdot (\text{destore} \cdot (\gamma \triangleright \text{store} \cdot \Gamma_0) \cdot Q_1)\)

\[\{\text{properties 4.18c and 4.18d}\}\]

\(\text{destore} \cdot \Gamma_1 = R \cdot \gamma \cdot Q_1\)

\[\Box\{\text{end proof 4.21}\}\]
The following lemma gives the relation between \( \text{stable2} \) and \( \text{stable1} \). Note that, according to this lemma, all stable-1 states are stable-2. Furthermore, stable-2 states differ from stable-1 states only at floating nodes. This way, a stable-1 state can correspond to several stable-2 states, with different stored values chosen at floating nodes (but such that connected nodes have the same value (coco1, cf. prop. 4.17b)).

### 4.22 Lemma

a) For \( \gamma \in \text{NST} \) and \((Q, \Gamma) \in \text{ST1} : \)

\[
\text{stable2} \cdot \gamma \cdot (Q, \Gamma) = (\text{stable1} \cdot \gamma \cdot (\text{destore} \cdot (Q, \Gamma)) \land \text{coco1} \cdot \Gamma \cdot Q)
\]

b) For \( \gamma \in \text{NST} : \)

\(\text{stable1} \cdot \gamma \subseteq \text{stable2} \cdot \gamma \)

Lemma 4.22b follows from lemma 4.22a using prop. 4.17c and def. 4.8 (stable2).

#### Proof of 4.22a

\[
\text{stable2} \cdot \gamma \cdot (Q, \Gamma) = \{ \text{def. 4.8 (stable2)} \}
\]

\[
(\mathcal{B} \mathcal{r}c : \mathcal{D} \mathcal{C} \cdot \mathcal{r}c : \text{next2} \cdot \gamma \cdot ((Q, \Gamma), rc) : ((Q, \Gamma), rc))
\]

\[
= \{ \text{def. 4.7 (next2)} \}
\]

\[
(\mathcal{B} \mathcal{r}c : \mathcal{D} \mathcal{C} \cdot \mathcal{r}c : \begin{align*}
&\mathcal{A} s : (rc-s = 0) \land \neg \mathcal{C} \mathcal{A} \cdot (Q, \Gamma) \cdot s : \mathcal{C} \mathcal{O} \mathcal{S} \mathcal{T} \cdot \Gamma \cdot Q \cdot s \\
&\lor (\mathcal{A} s : (rc-s > 0) \lor \mathcal{C} \mathcal{A} \cdot (Q, \Gamma) \cdot s : Q \cdot s = Q \cdot s )
\end{align*}
\land (\Gamma = R \cdot \gamma \cup \text{store} \cdot \Gamma \cdot Q) \\
\land (\mathcal{A} s : \mathcal{C} \mathcal{O} \mathcal{S} \mathcal{T} \cdot \Gamma \cdot Q \cdot s : rc-s = 0) \\
\land (\mathcal{A} s : \neg \mathcal{C} \mathcal{O} \mathcal{S} \mathcal{T} \cdot \Gamma \cdot Q \cdot s \land (rc-s > 0) : rc-s = rc-s - 1) \\
\land (\mathcal{A} s : \neg \mathcal{C} \mathcal{O} \mathcal{S} \mathcal{T} \cdot \Gamma \cdot Q \cdot s \land (rc-s = 0) : rc-s \in \mathbb{N})
\}
\]

\[
= \{ \text{properties 4.19a and 4.19b} \}
\]

\[
(\mathcal{B} \mathcal{r}c : \mathcal{D} \mathcal{C} \cdot \mathcal{r}c : \begin{align*}
&\mathcal{A} s : (rc-s = 0) : \mathcal{C} \mathcal{O} \mathcal{S} \mathcal{T} \cdot \Gamma \cdot Q \cdot s  \\
&\lor (\mathcal{A} s : (rc-s > 0) : Q \cdot s = Q \cdot s) \\
&\lor (\Gamma = R \cdot \gamma \cup \text{store} \cdot \Gamma \cdot Q) \\
&\lor (\mathcal{A} s : \mathcal{C} \mathcal{O} \mathcal{S} \mathcal{T} \cdot \Gamma \cdot Q \cdot s : rc-s = 0) \\
&\lor (\mathcal{A} s : \neg \mathcal{C} \mathcal{O} \mathcal{S} \mathcal{T} \cdot \Gamma \cdot Q \cdot s \land (rc-s > 0) : rc-s = rc-s - 1) \\
&\lor (\mathcal{A} s : \neg \mathcal{C} \mathcal{O} \mathcal{S} \mathcal{T} \cdot \Gamma \cdot Q \cdot s \land (rc-s = 0) : rc-s \in \mathbb{N})
\}
\]

\[
= \{ \text{calculus, def. 4.5 (coco0)} \}
\]

\[
\text{coco0} \cdot \Gamma \cdot Q \land (\Gamma = R \cdot \gamma \cup \text{store} \cdot \Gamma \cdot Q)
\]

\[
= \{ \text{note 0 and note 1} \}
\]

\[
\text{coco0} \cdot \Gamma \cdot Q \land (\text{destore} \cdot \Gamma = R \cdot \gamma \cdot Q) \land \text{coco1} \cdot \Gamma \cdot Q
\]

\[
= \{ \text{def. coco0} \}
\]

\[
\text{coco0} \cdot (\text{destore} \cdot \Gamma \cdot Q) \land (\text{destore} \cdot \Gamma = R \cdot \gamma \cdot Q) \land \text{coco1} \cdot \Gamma \cdot Q
\]

\[
= \{ \text{def. 1.19 (stable0), note that destore} \cdot \Gamma \in \text{NST} \}
\]

\[
\text{stable0} \cdot \gamma \cdot (Q, \text{destore} \cdot \Gamma) \land \text{coco1} \cdot \Gamma \cdot Q
\]
\[
\begin{align*}
&= \text{[def. 4.4 (extension destore) and lemma 3.5]} \\
&\quad\text{stable1-}\gamma(\text{destore}(Q,\Gamma)) \land \text{coco1-}\Gamma\cdot Q \\
\text{note } Q: \quad &\Gamma = R\cdot(\gamma \not\in \text{store-}\Gamma)\cdot Q \\
\Rightarrow &\quad (\text{see note 1 of lemma 4.21, and use prop. 4.17c}) \\
&\quad (\text{destore-}\Gamma = R\cdot\gamma\cdot Q) \land \text{coco1-}\Gamma\cdot Q
\end{align*}
\]

\[
\begin{align*}
\text{note 1:} \quad &\text{Assume } (\text{destore-}\Gamma = R\cdot\gamma\cdot Q) \land \text{coco1-}\Gamma\cdot Q. \text{ Let } x \in \mathbb{N}. \text{ Then:} \\
&\quad \Gamma\cdot x \\
&= \quad \text{[lemma 4.20b and prop. 4.18a]} \\
&\quad \text{destore-}\Gamma\cdot x \uplus \text{store-}\Gamma\cdot x \\
&= \quad \text{[assumption]} \\
&\quad R\cdot\gamma\cdot Q\cdot x \uplus \text{store-}\Gamma\cdot x \\
&= \quad \text{[def. 4.6 (R)]} \\
&\quad (\forall y : \text{cp-Q} \cdot [x,y] : y \cdot y) \uplus \text{store-}\Gamma\cdot x \\
&= \quad \text{[prop. 4.17b, assumption]} \\
&\quad (\forall y : \text{cp-Q} \cdot [x,y] : y \cdot y) \uplus (\forall y : \text{cp-Q} \cdot [x,y] : \text{store-}\Gamma\cdot y) \\
&= \quad \text{[associativity of } \uplus] \\
&\quad (\forall y : \text{cp-Q} \cdot [x,y] : y \cdot y) \uplus \text{store-}\Gamma\cdot y \\
&= \quad \text{[prop. 4.18a]} \\
&\quad (\forall y : \text{cp-Q} \cdot [x,y] : (y \not\in \text{store-}\Gamma) \cdot y) \\
&= \quad \text{[def. 4.6 (R)]} \\
&\quad R\cdot(\gamma \not\in \text{store-}\Gamma)\cdot Q\cdot x \\
\%
\end{align*}
\]

\(\square\) (end proof 4.22)

4.23 Lemma

For \(\gamma \in \text{NST} \) and \(\Pi \in \text{ST1} : \)

\[
\text{feasible2-}\gamma\cdot \Pi = \text{feasible1-}\gamma\cdot (\text{destore-}\Pi)
\]

Proof

\[
\begin{align*}
\text{feasible2-}\gamma\cdot \Pi \\
&= \quad \text{[def. 4.8 (feasible2)]} \\
&= \quad (B \cdot rc : \text{DC-rc} : (\text{next2-}\gamma)^+ \cdot (\Pi,rc) \cdot (\Pi,rc)) \\
\Rightarrow &\quad \text{[lemma 4.21]} \\
&= \quad (B \cdot rc : \text{DC-rc} : (\text{next1-}\gamma)^+ \cdot (\text{destore-}\Pi,rc) \cdot (\text{destore-}\Pi,rc)) \\
&= \quad \text{[def. 3.3 (feasible1)]} \\
&\quad \text{feasible1-}\gamma\cdot (\text{destore-}\Pi)
\end{align*}
\]

\(\square\)
Lemmas 4.21 and 4.23 show that the behaviour defined in this model is not an extension of the behaviour defined the previous model (chapter 3). The following example shows that the newly defined behaviour is a restriction of the previously defined one.

4.24 Example

Consider the circuit depicted in figure 4.1a. According to the model from chapter 3, states \( \Pi_0 \) and \( \Pi_1 \) (fig. 4.1b) can oscillate, viz. \( \text{next} \gamma \Pi_0 \Pi_1 \land \text{next} \gamma \Pi_1 \Pi_0 \). This behaviour is not possible according to the extended model presented here. Namely, in order for \( \Pi_2 \) to be a successor of \( \Pi_3 \) and vice versa, with \( \text{destore} \Pi_0 \Pi_2 = \Pi_1 \ (0 \leq 2) \), \( \Pi_2 \) and \( \Pi_3 \) must be as given in figure 4.1b. But these states are not feasible-2, and lead to the stable state \( \Pi_4 \) (fig. 4.1b).

\[
\begin{array}{cccc}
\text{state} & \text{next} \gamma \Pi_0 \Pi_1 \land \text{next} \gamma \Pi_1 \Pi_0 \\
\Pi_0 & 0 & 1 & (H) \emptyset \\
\Pi_2 & 1 & 0 & (H) \{L\} \\
\Pi_3 & 1 & 0 & (L) \emptyset \\
\Pi_4 & 1 & 1 & (H) \{L\} \\
\end{array}
\]

Properties of WM2

The relation between WM2 and WM1 is investigated and some properties of WM2 are given. We will, in particular, investigate whether the counterparts of the main results from chapters 1 and 3, i.e., corollary 1.40, theorem 3.9, and corollary 3.10, also hold for WM2. The results of the investigation are discussed at the end of this section.

4.25 Lemma

\((A \ C \gamma : \text{CIR} \land \gamma : \text{WM1} \land \gamma : \text{WM2} \land \gamma)\)

Proof

Let \( C \in \text{CIR} \) and \( \gamma \in \text{NST} \). Then:

\[WM1 \land \gamma = (\text{def. 3.3 (WM1)})\]

\[\text{feasible1} \land \gamma \subseteq \text{cgdo}\]

\[\Rightarrow (\text{lemma 4.23})\]

\[(A \Pi : \text{feasible2} \land \gamma : \Pi : \text{cgdo} \land \text{destore} \land \Pi)\]

\[= (\text{def. 4.9 (cgdo)})\]

\[\text{feasible2} \land \gamma \subseteq \text{cgdo}\]

\[\Rightarrow (\text{note 0})\]

\[\text{FSL2} \land \gamma \subseteq (\text{FCA0} \land \text{cgdiL})\]

\[= (\text{def. 4.15 (WM2)})\]

\[WM2 \land \gamma\]
Assume $feastable2 \cdot \gamma \subseteq cgdl0$. Then:

$FSL2 \cdot \gamma \Psi$

$\Rightarrow \{\text{lemma 4.12 and assumption}\}$

$(A, i : 0sL < \# \cdot \Psi : cgdl0 \cdot \Psi_i)$

$\Rightarrow \{\text{dummy introduction}\}$

$(A, i, Q, \Gamma : (0sL < \# \cdot \Psi) \land (\Psi_i = (Q, \Gamma)) \land cgdl0 \cdot (Q, \Gamma))$

$\Rightarrow \{\text{prop. 4.19c}\}$

$(A, i, Q, \Gamma, s : (0sL < \# \cdot \Psi) \land (\Psi_i = (Q, \Gamma)) \land SW \cdot s : cgdl0 \cdot \Gamma \cdot s)$

$\Rightarrow \{\text{prop. 4.19a}\}$

$(A, i, Q, \Gamma, s : (0sL < \# \cdot \Psi) \land (\Psi_i = (Q, \Gamma)) \land SW \cdot s : cgdl0 \cdot \Gamma \cdot s \land \neg CA \cdot (Q, \Gamma) \cdot s)$

$\Rightarrow \{\text{def. 4.10 (gdi)}\}$

$(A, i, Q, \Gamma, s : (0sL < \# \cdot \Psi) \land (\Psi_i = (Q, \Gamma)) \land SW \cdot s : gdi \cdot (Q, \Gamma) \cdot s \land \neg CA \cdot (Q, \Gamma) \cdot s)$

$\Rightarrow \{\text{dummy introduction}\}$

$(A, i, s : (0sL < \# \cdot \Psi) \land SW \cdot s : gdi \cdot \Psi_i \cdot s \land \neg CA \cdot \Psi_i \cdot s)$

$\Rightarrow \{\text{definitions 4.13 (FCA0) and 4.14 (cgdl})\}$

$FCA0 \cdot \Psi \land cgdl \cdot \Psi$

$\square$

The counterpart of theorems 1.28 and 3.8 is expressed in part a of the following theorem, the counterpart of theorem 3.9 is expressed in part b. Notice that the new well-matchedness criterion is equal to the old ones.

4.26 Theorem

a $(A, C, \gamma : \text{CIR} \cdot C \land \text{NST} \cdot \gamma : WM2 \cdot C \cdot \gamma = \left(\text{stable2} \cdot \gamma \subseteq cgdl0\right))$

b $(WM2 = WMI) \land (WM2 = WMO)$

Proof

Let $C \in \text{CIR}$ and $\gamma \in \text{NST}$. Then:

$WM2 \cdot C \cdot \gamma$

$\Rightarrow \{\text{note 0}\}$

$\text{stable2} \cdot \gamma \subseteq cgdl0$

$\Rightarrow \{\text{lemma 4.22 using def. 4.9 (cgdl)}\}$

$\text{stable1} \cdot \gamma \subseteq cgdl0$

$\Rightarrow \{\text{theorem 3.8}\}$

$WMI \cdot C \cdot \gamma$

$\Rightarrow \{\text{lemma 4.25}\}$

$WM2 \cdot C \cdot \gamma$
Observe that all five assertions above are equal. Th. 3.9 completes the proof of 4.26b.

\[ \begin{align*}
\text{Let } C & \in \text{CIR and } \gamma \in \text{NST. Then:} \\
\text{WM2-C-}\gamma &= \{ \text{def. 4.15 (WM2)} \}
\text{FSL2-}\gamma \subseteq (\text{FCA0} \land \text{cgdlL}) \\
\Rightarrow & \quad \{ \text{calculus} \}
(A \Psi : \text{FSL2-}\gamma \Psi \land (\# \Psi = 1) : \text{FCA0-}\Psi \land \text{cgdlL-}\Psi) \\
& \quad \{ \text{definitions 4.13 (FCA0), 4.14 (cgdlL), and 4.10 (cgdlI)} \}
(A \Psi : \text{FSL2-}\gamma \Psi \land (\# \Psi = 1) : (A s : \text{SW-} s \land \neg \text{CA-} \Psi_0 s \land \text{gdl-} \Psi_0 s)) \\
& \quad \{ \text{def. 4.10 (gdlI)} \}
(A \Psi, Q, \Gamma : \text{FSL2-} \gamma \Psi \land (\# \Psi = 1) \land (\Psi_0 = (Q, \Gamma)) : (A s : \text{SW-} s : gdl- \Gamma s)) \\
& \quad \{ \text{prop. 4.19c} \}
(A \Psi : \text{FSL2-}\gamma \Psi \land (\# \Psi = 1) : \text{cgdl-} \Psi_0) \\
& \quad \{ \text{def. 4.11 (FSL2)} \}
(A \Pi : (\text{B rc} : \text{DC-rc} : \text{next2-}\gamma (\Pi, \text{rc}) - (\Pi, \text{rc})) : \text{cgdl-} \Pi) \\
& \quad \{ \text{def. 4.8 (stable2)} \}
\text{stable2-}\gamma \subseteq \text{cgdl0}
\end{align*} \]

\[ \square \]

The counterpart of cor. 1.40 and th. 3.10 is expressed in the following theorems.

4.27 Theorem
\[ \begin{align*}
(A, C, \gamma) : \text{CIR-C \land NST-}\gamma \\
: \text{WM2-C-}\gamma &= \{ \text{stable2-}\gamma = \{ Q, \Gamma : (\text{destore-}(Q, \Gamma) = \gamma_\ast) \land \text{cocol-} \Gamma \cdot Q : (Q, \Gamma) \} } \\
\land \text{cgdl0-}\gamma_\ast \}
\end{align*} \]

Proof
\[ \begin{align*}
\text{Let } C & \in \text{CIR and } \gamma \in \text{NST. Then:} \\
\text{WM2-C-}\gamma &= \{ \text{th. 4.26b} \}
\text{WM1-C-}\gamma = \{ \text{th. 3.10a} \}
\text{stable1-}\gamma = \{ \gamma_\ast \} \land \text{cgdl0-}\gamma_\ast \\
& \quad \{ \text{lemma 4.22} \}
\text{stable2-}\gamma = \{ Q, \Gamma : (\text{destore-}(Q, \Gamma) = \gamma_\ast) \land \text{cocol-} \Gamma \cdot Q : (Q, \Gamma) \} \land \text{cgdl0-}\gamma_\ast
\end{align*} \]

\[ \square \]
4.28 Theorem  
\((A,C,\gamma) \in \text{CIR-C} \land \text{NST-}\gamma : \text{WM2-C-}\gamma \rightarrow (\text{feasible2-}\gamma \equiv \text{stable2-}\gamma)\)

Proof

Let \(C \in \text{CIR}\) and \(\gamma \in \text{NST}\).
Assume \(\text{WM2-C-}\gamma\). On account of th. 4.26b we conclude: \(\text{WM1-C-}\gamma\). Then:

(i) Let \(\text{ST1-}\Pi\). Then:
\[\text{feasible2-}\gamma\cdot \Pi\]
\[\Rightarrow (\text{lemma 4.23})\]
\[\text{feasible1-}\gamma\cdot (\text{destore-}\Pi)\]
\[\Rightarrow (\text{from assumption: WM1-C-}\gamma, \text{ use lemma 3.10c})\]
\[\text{stable1-}\gamma\cdot (\text{destore-}\Pi)\]

(ii) Let \(\text{ST1-(Q,}\Gamma)\). Then:
\[\text{feasible2-}\gamma\cdot (Q,\Gamma)\]
\[\Rightarrow (\text{def. 4.8 (feasible2): (Q,}\Gamma) \text{ has a predecessor})\]
\[\text{B}\Gamma_1 : \text{NSTC-}\Gamma_1 : \Gamma = R\cdot (\gamma \wedge \text{store-}\gamma\cdot Q)\]
\[\Rightarrow (\text{prop. 4.17c})\]
\[\text{coco1-}\Gamma\cdot Q\]

From (i), (ii), and lemma 4.22 follows: \(\text{feasible2-}\gamma \subseteq \text{stable2-}\gamma\).

From definition 4.8 follows that \(\text{stable2-}\gamma \subseteq \text{feasible2-}\gamma\), which completes the proof.

\(\square\)

The new well-matchedness notion, \(\text{WM2}\), is equal to the previously defined well-matchedness notions \(\text{WM0}\) and \(\text{WM1}\) (th. 4.26b). Theorems 4.27 and 4.28 are comparable with corollaries 1.40 and 3.10. As in the models from chapters 1 and 3, the resulting states are, in case \(\text{WM2-C-}\gamma\) holds, all stable and equal — abstracted from stored values — to \(\gamma\).

Additional correctness criteria

The two additional correctness criteria for initial behaviour, \(\text{CF}\) (def. 1.46) and \(\text{cst0}\) (def. 1.47), must be extended to the the new set of states \(\text{ST1}\). Since conflicts at capacitance level are assumed to be harmless (assumption (1a)), conflict-freeness in this model means, as before, that no nodes are connected to both types of sources. Correctness of state transitions in this model means, as before, that all switches that are possibly in the process of changing state, i.e. are inconsistent-0, have stable gates. The two notions are extended in the definition below (compare with definitions 1.46 and 1.47).
4.29 Definition

$CF$ is extended to $ST1 \rightarrow \theta$ by:

$CF(Q, \Gamma) = (A.s : N \cdot x : \Gamma \cdot x \neq \{L, H\})$

$cst0$ is extended to $ST1 \rightarrow ST1 \rightarrow \theta$ by:

$cst0(Q, \Gamma_0) \rightarrow (Q, \Gamma_1) = (A.s : \neg \text{consistent0} \cdot \Gamma_0 \cdot Q \cdot s : \Gamma_0 \cdot (g \cdot s) = \Gamma_1 \cdot (g \cdot s))$

4.30 Property

For all $\Pi, \Pi_1 \in ST1$:

a $CF \cdot \Pi = CF \cdot (\text{destore} \cdot \Pi)$

b $cst0 \cdot \Pi \cdot \Pi_1 = cst0 \cdot (\text{destore} \cdot \Pi) \cdot (\text{destore} \cdot \Pi_1)$

Property 4.30a follows from def. 4.29 ($CF$) and 4.4 ($\text{destore}$). Property 4.30b is proved below.

Proof of 4.30b

Let $ST1 \cdot (Q, \Gamma)$ and $ST1 \cdot (Q, \Gamma_1)$. Then:

$cst0 \cdot (Q, \Gamma) \cdot (Q, \Gamma_1) = \{\text{def. } 4.29 \text{ (cst0)}\}$

$(A.s : \neg \text{consistent0} \cdot \Gamma \cdot Q \cdot s : \Gamma \cdot (g \cdot s) = \Gamma_1 \cdot (g \cdot s))$

= $\{\text{note 0}\}$

$(A.s : \neg \text{consistent0} \cdot \Gamma \cdot Q \cdot s : \text{destore} \cdot \Gamma \cdot (g \cdot s) = \text{destore} \cdot \Gamma_1 \cdot (g \cdot s))$

= $\{\text{def. } 4.5 \text{ (consistent0)}\}$

$(A.s : \neg \text{consistent0} \cdot (\text{destore} \cdot \Gamma) \cdot Q \cdot s : \text{destore} \cdot \Gamma \cdot (g \cdot s) = \text{destore} \cdot \Gamma_1 \cdot (g \cdot s))$

= $\{\text{def. } 4.29 \text{ (cst0)}\}$

$cst0 \cdot (Q, \text{destore} \cdot \Gamma) \cdot (Q, \text{destore} \cdot \Gamma_1) = \{\text{def. } 4.4 \text{ (extension destore)}\}$

$cst0 \cdot (\text{destore} \cdot (Q, \Gamma)) \cdot (\text{destore} \cdot (Q, \Gamma_1))$

Note 0

Let $ST1 \cdot (Q, \Gamma), ST1 \cdot (Q, \Gamma_1)$, and $SW \cdot s$. Then:

$\neg \text{consistent0} \cdot \Gamma \cdot Q \cdot s$

$\Rightarrow \{\text{prop. } 4.19b\}$

$gdo \cdot \Gamma \cdot s$

= $\{\text{definitions } 4.9 \text{ (gdo) and } 1.11 \text{ (gdo)}\}$

$\text{destore} \cdot \Gamma \cdot (g \cdot s) \in \{\{L\}, \{H\}\}$

= $\{\text{def. } 4.4 \text{ (destore)}\}$

$(\Gamma \cdot (g \cdot s) \in \{\{L\}, \{H\}\}) \land (\text{destore} \cdot \Gamma \cdot (g \cdot s) = \Gamma \cdot (g \cdot s))$

$\Rightarrow \{\text{calculus, using def. } 4.4 \text{ (destore)}\}$

$(\Gamma \cdot (g \cdot s) = \Gamma_1 \cdot (g \cdot s)) = (\text{destore} \cdot \Gamma \cdot (g \cdot s) = \text{destore} \cdot \Gamma_1 \cdot (g \cdot s))$
Let $C \in \text{CIR \ , \ } y \in \text{NST \ , \ and \ } \Pi \in \text{ST1} \text{ satisfy } WM2\cdot C \cdot y \wedge \text{feasible}\cdot 2\cdot y \cdot \Pi$.

On account of theorems 4.27 and 4.28 state $\Pi$ satisfies stable\cdot 2\cdot y \cdot \Pi \wedge cgs\cdot 0\cdot y \cdot \Pi$. This means that $\Pi$ has only one successor state, viz. $\Pi$ itself (the proof is left to the reader: use def. 4.7 (next2)). Since destore\cdot y \cdot \Pi = \gamma_y$ (th. 4.27) and csg\cdot 0\cdot y \cdot \Pi holds (use def. 4.29 (csg0)), the additional correctness criterion for initial behaviour, besides $WM2\cdot C\cdot y$, is, as in the previous chapters (cf. sections 1.5 and 3.1 end): $CF\cdot \gamma_y$ (use prop. 4.30(a)). Consequently, the complete correctness criterion for initial behaviour is, as before: $WM0\cdot C\cdot y \wedge CF\cdot \gamma_y$.

### 4.2 Well-Functioning

As explained in sections 0.2 and 3.3 we are not only interested in initial behaviour of a circuit,
but also in dynamic behaviour of a circuit. The latter is the behaviour of a circuit starting in a
— known — state when the source-connection changes. This type of behaviour is discussed
informally in section 3.3. Here, we resume this discussion, formalise correctness criteria for
dynamic behaviour, and define the notion well-functioning. Several examples will illustrate this
notion.

In chapter 3 we have seen that we need to be able to express delay-restrictions in order to
describe dynamic behaviour in a sensible way (cf. ex. 3.11, ex. 3.12, and section 3.3). Similar as
in chapter 3 we define the next-state function for a specific reaction delay $rd$ by replacing
restriction R2.2 in the definition of the next-state function (def. 4.7) by restriction R2.2" (cf.
section 3.2):

#### 4.31 Definition

For $rd \in DC$, $next_2 \in \text{NST} \rightarrow (ST1 \times DC) \rightarrow (ST1 \times DC)$ is defined by:

\[
next_2 \cdot y \cdot ((Q_0, \Gamma_0), rc_0) \cdot ((Q_1, \Gamma_1), rc_1) =
\begin{align*}
&\left( (A \cdot s : (rc_0 \cdot s = 0) \wedge \neg CA \cdot (Q_0, \Gamma_0) \cdot s : \text{consistent} \cdot 0 \cdot Q_1 \cdot s ) \right) \\
&\wedge (A \cdot s : (rc_0 \cdot s > 0) \vee CA \cdot (Q_0, \Gamma_0) \cdot s : Q_1 \cdot s = Q_0 \cdot s ) \\
&\wedge (\Gamma_1 = R \cdot y \cdot \text{store} \cdot \Gamma_0 \cdot Q_1 ) \\
&\wedge (A \cdot s : \text{consistent} \cdot 0 \cdot \Gamma_1 \cdot Q_1 \cdot s : rc_1 \cdot s = 0 ) \\
&\wedge (A \cdot s : \neg \text{consistent} \cdot R \cdot Q_1 \cdot s \wedge (rc_0 \cdot s > 0) : rc_1 \cdot s = rc_0 \cdot s - 1 ) \\
&\wedge (A \cdot s : \neg \text{consistent} \cdot R \cdot Q_1 \cdot s \wedge (rc_0 \cdot s = 0) : rc_1 \cdot s = rc_0 \cdot s ) \\
\end{align*}
\]

Let $C$ be a circuit, let the starting state of the circuit — extended with a reaction delay counter
be $(\Pi, rc)$, let the new source-connection be $y$, and let $RD$ be the set of reaction-delays.
The set of intermediate and resulting states can then be defined as (cf. section 3.3):

\[
\{ rd \cdot 1 : RD \cdot rd \wedge 0 \cdot s : (next_2 \cdot y) \cdot (\Pi, rc) \}.
\]
As correctness criteria for the (dynamic) behaviour of this circuit we will use:

i) all states in the above set are completely gate defined

ii) all states in the above set are conflict free

iii) the capacitance-control is finite in every list of successive states in the above set

iv) all state transitions between successive states in the above set are correct

Before formalising these correctness criteria, we need to define lists of successive states starting
in a specific state, see iii and iv. Such a list, say $L$, starting in $(\Pi,rc)$, is characterised by:

$L_0 = (\Pi,rc) \land (R \triangleright RD-rd : (A i : 0 \leq i : L_{i+1} = \text{next}2_{rd} \gamma \Psi_{i+1}))$.

Abstracting from the reaction-delay counter then leads to the following definition.

4.32 Definition (resulting state lists 2)

For $rd \in DC$, $RSL2_{rd} \in \text{NST} \rightarrow (ST1 \times DC) \rightarrow \mathcal{O}(ST1) \rightarrow B$ is defined by:

$RSL2_{rd} \gamma (\Pi,rc) \cdot \Psi = (R \triangleright r1 : \mathcal{O}(DC) \cdot r1 : (\Psi_0 \cap \Psi_{r1})) = (\Pi,tc)

\land (A i : 0 \leq i : \text{next}2_{rd} \gamma (\Psi_i \cap \Psi_{i+1}))$.

Note that the set of these lists represents the set of intermediate and resulting states mentioned
above, abstracted from the reaction-delay counter.

Correctness criterion i above is already defined in def. 4.14 ($\text{cgdlL}$). The other three criteria
are formalised in the definition below, where a, b, and c formalise ii, iii, and iv respectively.

4.33 Definition

a) $CFL \in \mathcal{O}(ST1) \cup \mathcal{O}(ST1) \rightarrow B$ (conflict freeness for state lists) is defined by:

$CFL \cdot \Psi = (A i : 0 \leq i : \#^\gamma \Psi : CF \cdot \Psi_{i+1})$

b) $FCAI \in \mathcal{O}(ST1) \rightarrow B$ (finite capacitance-control for infinite state lists) is defined by:

$FCAI \cdot \Psi = (A j,s : 0 \leq j : SW \cdot s : (B l : j \leq i : \neg \text{CA} \cdot \Psi_{i+1}))$

c) $cstOL \in \mathcal{O}(ST1) \rightarrow B$ (correctness state transitions for infinite state lists) is defined by:

$cstOL \cdot \Psi = (A i : 0 \leq i : cstOL \cdot \Psi_{i+1})$

These four correctness criteria for dynamic behaviour are expressed in the notion $WF2$, which
stands for well-functioning 2. As a definition of $WF2$ we have argued:

4.34 Definition

For $RD \subseteq DC$ as the set of reaction-delays,

$WF2_{RD} \in \text{CIR} \rightarrow \text{NST} \rightarrow (ST1 \times DC) \rightarrow B$ is defined as:

$WF2_{RD} \cdot \gamma (\Pi,rc) = (R \triangleright RD : RSL2_{rd} \gamma (\Pi,rc) \cap (\text{cgdlL} \cap CFL \cap FCAI \cap cstOL))$. 
Before illustrating this definition by means of some examples, we return to the discussion on correctness criteria (section 3.3 and above).

4.35 Remark on correctness criteria

The correctness criteria we use are defined on elements of \((\cup rd \in RD \cdot rd : RSL2_{sH}\gamma : (\Pi,\gamma) )\). Alternative choices for correctness criteria are not hard to formalise.

For instance, we have chosen to classify transient conflicts as 'undesirable'; expressed in \(CFL\). An alternative choice is to classify transient conflicts as 'harmless', and only permanent conflicts as 'undesirable'. This can be formalised by replacing correctness criterion \(CFL\) in the definition of \(WF2\) by \(FCL \in \mathcal{L}(ST1) \rightarrow \mathcal{B}\) (\(FCL\) stands for 'finiteness conflicts for infinite state lists'), which is defined as: \(FCL : \Psi = (A \cdot x : \nexists x \cdot A \forall i : (B \cdot i, \Sigma, \Gamma : \exists j \cdot \Psi_j = (Q, \Gamma) : \Gamma \cdot x \in \{L, H\})\).\)

The following example illustrates the correctness criteria. Notice that well-functioning does not hold in ex. 4.36a, b, and c due to violation of \(CFL, FCAI,\) and \(custo\) respectively. In example 4.36a the inverter (ex. 3.11) is reconsidered.

4.36 Example

a The circuit given in ex. 3.11 (fig. 4.2a), with starting state \((y_0, 0)\) and new source-connection \((\delta)\) (see ex. 3.11), is not well-functioning without delay restrictions: \(\neg WF2_{ DC \cdot C \cdot \delta \cdot (y_0, 0)}\). Note that, for all \(rd \in DC:\ RSL2_{sH} \delta \cdot (y_0, 0) \in (cgd1L \cap FCAI \cap custoL)\), but for \(rd\) satisfying \(rd_{HL} > rd_{LS}\), \(RSL2_{sH} \delta \cdot (y_0, 0) \notin CFL\) (cf. the left-most path in figure 4.2b).

If the reaction-delay is restricted by \(rd_{HL} > rd_{LS}\) (or: \(rd_{HS} > rd_{LS}\)), the circuit is well-functioning. That is, with \(RD = \{rd : DC \cdot rd \land (rd_{HL} > rd_{LS}) : rd\}\) as the set of restricted reaction-delays, \(WF2_{ DC \cdot C \cdot \delta \cdot (y_0, 0)}\) holds (cf. fig. 4.2b).

Similarly, if the starting state is the resulting state of figure 4.2b and the new source-connection is \(\{x, \{L\}, \{x, \delta\}\}\) (a downgoing input transition), the necessary and sufficient requirement for well functioning is \(rd_{HL} > rd_{LS}\). Consequently, the inverter functions correctly for both an upgoing and a downgoing transition if and only if the set of reaction-delays \(RD\) satisfies \(RD \subseteq \{rd : (rd_{HL} > rd_{LS}) \land (rd_{HS} > rd_{LS}) : rd\}\).

\[
\begin{array}{c|c|c|c}
\hline
& x & \text{H} & \text{L} \\
\hline
\text{I} & f_1 & 0 & 1 \\
\text{r} & 0 & 1 & 0 \\
\hline
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c}
\hline
& \text{H} & \text{L} & x & z \ \\
\hline
0 & 1 & \{L\} & \{H\} \ \\
\hline
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c}
\hline
& \text{H} & \text{L} & x & \text{H} & \text{L} \ \\
\hline
1 & 1 & \{H\} & \{L, H\} & 0 & 0 & \{H\} & \{h\} \ \\
\hline
1 & 0 & \{H\} & \{L\} & \_ & \_ & \ \\
\hline
\end{array}
\]

figure 4.2a: ex. 4.36a,b; circuit

figure 4.2b: ex. 4.36a, behaviour w.r.t. DC
b. Consider again the circuit from example 3.11 with starting state \( \gamma_s \), now with 1 as the new source-connection. The immediate successor of \( \gamma_s \) is \( \Pi_0 \), where \( \Pi_0 \) is the stable state \( \{(s_0,0),(s_1,1)\},(x,\{i\}),c_t,\{H\}) \). This state satisfies \( \text{cfl} \wedge \Pi_0 \wedge \text{cst}0 \wedge x=1 \), hence, \( \text{RSL}2 \Pi_0 \wedge \text{cst}0 \wedge \Pi_0 \), and hence, \( \text{RSL}2 \Pi_0 \subseteq \text{cfl} \wedge \Pi_0 \), for each \( r \in DC \). However, since \( \text{CA} \Pi_0 \wedge 0 \leq s \leq 1 \) holds, \( \text{RSL}2 \Pi_0 \wedge \gamma_s \not\subseteq \text{FCS}1 \) holds for each \( r \in DC \), and, consequently, for \( RD \) such that \( RD \not= \emptyset \wedge RD \subseteq DC \) : \( \neg \text{WF}2 \Pi_0 \). 

c. Consider the circuit drawn in figure 4.2c with starting state \( \Pi_0 = \{(s_0,0),(s_1,1),(s_2,0)\},(x,\{H\}),y,\{(L)\},(c_t,\{L\}) \), which is the resulting stable state of source-connection \( (x,\{H\}),y,\{(L)\},(c_t,\{L\}) \). Let the new source-connection be \( \delta = \{(x,\delta),(y,\delta),(z,\{H\})\} \), and assume that the reaction delay \( r_d \) satisfies \( r_d = r_{d_1} \). The resulting state list is given in figure 4.2d, where \( t_0 \), \( t_1 \) denote the state transitions. Note that this state list satisfies \( \text{cfl} \wedge \Pi_0 \). The transitions \( t_0 \) and \( t_1 \) are correct, but \( t_2 \) and \( t_3 \) are not correct. Consequently, the state list does not satisfy \( \text{cst}0 \) and, hence, \( \neg \text{WF}2 \Pi_0 \). 

\[
\text{figure 4.2c: ex. 4.36c, circuit} \quad \text{figure 4.2d: ex. 4.36c, state list}
\]

4.37 Property

Let \( \text{CIR-C} \wedge \text{NST-C} \wedge \text{STXDC-}\). If \( RD \not= \emptyset \), then \( \text{WF}2 \Pi_0 \wedge \text{C-C} \iff \text{WF}2 \Pi_0 \wedge \text{C-C} \). Property 4.37 follows directly from definition 4.34.

The following example illustrates the notion of well-functioning by means of the flip-flop from example 2.15.

4.38 Example

Reconsider the circuit from example 2.15, depicted in figure 4.3a below. In ex. 2.15 we argued that, for source-connection \( \gamma = \{(s,\{L\}),y,\{H\},(c_0,\delta),(c_1,\delta),(c_2,\delta)\} \), the resulting state is \( \gamma_s = \{(s,\{L\}),y,\{H\},(c_0,\delta),(c_1,\delta),(c_2,\delta)\} \), which holds (and hence, cf. th. 4.26b, also \( \text{WM}2 \)). For brevity, the permanent sources, depicted in figure 4.3a, are omitted in node-states (e.g. in \( \gamma_s \)). Similarly, in stable circuit states, the switch-state is omitted, since it follows directly from the node-state (e.g. in \( \gamma_s \)).
Let RD1 = \{rd : RD \rightarrow rd \land (rd \land s_2 = rd \land s_3) \land (rd \land s_4 = rd \land s_3) : rd \}.

a With \((y_0,0)\) as starting state, let the source-connection change in y from H to L, that is, let the new source-connection be \(\delta_0 = \{(x,\{L\}),(y,\{L\}),(x_0,\emptyset),(x_1,\emptyset),(x_2,\emptyset)\}\). The resulting state lists are schematically given in fig. 4.3b. From this figure we can conclude that \(WF2_{RD1} \land C \land \Psi(y,0)\) holds, and that the resulting stable state is \(\Delta_0 = \{(x,\{L\}),(y,\{L\}),(x_0,\emptyset),(x_1,\{H\}),(x_2,\{L\})\}\).

b Let the source-connection now change in x from L to H, i.e., the new source-connection is \(\delta_1 = \{(x,\{H\}),(y,\{L\}),(x_0,\emptyset),(x_1,\emptyset),(x_2,\emptyset)\}\). The immediate successor of \(\Delta_0\) is \(\Delta_1 = \{(x,\{H\}),(y,\{L\}),(x_0,\emptyset),(x_1,\{H\}),(x_2,\{L\})\}\), which is a stable state. The (unique) resulting state list is \(\Psi\) defined by: \(\Psi_0 = \Delta_0 \land (A_i : 0 \leq i : \Psi_i = \Delta_1)\). It is now easily verified that \(WF2_{RD1} \land C \land \delta_1 \land (\Delta_0,0)\) holds.

c Let the source-connection now change in y from L to H, i.e., the new source-connection is \(\delta_2 = \{(x,\{H\}),(y,\{H\}),(x_0,\emptyset),(x_1,\emptyset),(x_2,\emptyset)\}\). Figure 4.3c shows that if the reaction-delay \(rd\) satisfies \(rd_{H} < rd_{L} \land s_1\), then the resulting state lists do not satisfy \(CFL\). The resulting state lists for reaction-delays restricted by \(rd_{H} \geq rd_{L} \land s_1\) are given in fig. 4.3d. From fig. 4.3d we can conclude that, for \(RD0 \subset RD1 \land \{rd : rd_{H} < rd_{L} \land s_1 : rd\}\), \(WF2_{RD0} \land C \land \delta_2 \land (\Delta_1,0)\) holds (see prop. 4.37), and that the resulting stable state is \(\{(x,\{H\}),(y,\{H\}),(x_0,\{H\}),(x_1,\{L\}),(x_2,\{H\})\}\).
For later use, in chapter 7, we present two lemmas on well-functioning. The first lemma (4.39a) shows that correct states (extended with a reaction-delay counter) have unique next states. As a result of this, correct resulting state lists have a suffix consisting of a repeated cycle of feasible states (lemma 4.39b). Example 4.40 shows that well-functioning circuits need not end in a stable state.

4.39 Lemma
a For all \( \text{rd} \), \( \text{DC} \cdot \text{rd} \):
\[
c_{\text{gd}} \cdot \Pi \Rightarrow (\text{A:} \text{DC} \cdot \text{rc} \cdot \text{#}(\text{next} \cdot \text{rd} \cdot \gamma(\Pi, \text{rc})) = 1)
\]
b For all \( \text{rd} \), \( \text{DC} \cdot \text{rd} \):
\[
\text{RS} \cdot \text{rd} \cdot \gamma(\Pi, \text{rc}) \cdot \psi \wedge c_{\text{gd}} \cdot \text{L} \cdot \psi
\Rightarrow (E \cdot \Phi \cdot \Phi \cdot L \cdot (\text{ST}) \cdot \Phi \cdot \Phi \wedge \text{FS} \cdot \text{rd} \cdot \gamma \cdot \Phi \cdot \Phi \cdot \psi = \text{ca} \cdot \Phi \cdot \Phi \cdot \Phi \cdot \Phi (\Phi \cdot \Phi))
\]

The proofs of 4.39a and 4.39b are fairly straightforward.

A result of the last lemma is that in case of \(WF2\) all reachable state lists "lead to" a feasible state list. So the "resulting states" can be defined as these reachable feasible states. The following example shows that such a resulting state need not be stable.

4.40 Example
Consider the circuit depicted in figure 4.4a with source-connection \( \{(x_0, \{L\}), (x_1, \emptyset), (x_2, \emptyset), (x_3, \emptyset), (x_4, \emptyset)\} \). The resulting stable state, \( \Pi_0 \), is given in figure 4.4b. Let the source-connection change in \( x_0 \) from \( \{L\} \) to \( \{H\} \). The resulting state list, with \( \text{rd}=0 \), is given in figure 4.4b. Although the circuit does not reach a stable state, it can easily be verified (using fig. 4.4b) that it is well-functioning, that is, \(WF2_{\{0\}} \cdot C \cdot \delta \cdot (\Pi_0, 0)\) holds.

![figure 4.4a: circuit ex. 4.40](image)

![figure 4.4b: state list ex. 4.40](image)
4.3 Further research

In the model presented in the previous sections we do not distinguish different strengths of capacitances (cf. section 4.0 'modelling charge storage'). In this section we show how these different strengths can be modelled by generalising our model. We also discuss the modelling of resistances using a similar notion of strength. Both topics are discussed informally, and not developed in detail. The consequences of these extensions for our model are also discussed.

**Multiple capacitance-strengths**

Nodes can have different capacitances. If two nodes with stored values of different type (one \( l \), the other \( h \) ) become connected, the stored value of the node with the strongest capacitance will override the other stored value, or, if the capacitances of the nodes are equally strong, a 'conflict at capacitance level' will occur.

As in the models in [Bry1], [DK], [Se], and [Wil], different capacitance-strengths can be modelled by attaching a strength component to the values \( H \) and \( L \). This way the class of stored values is subdivided. The basic idea is that a signal with a larger strength overrides a signal with a smaller strength (compare with assumption (1b) in section 4.0). Let the set of strengths be \( \text{STR} = \{ i : 1 \leq i \leq n \} \), with \( n \in \mathbb{N} \land n \geq 2 \). Source values (H and L in the model presented in the previous sections) have the largest strength attached to them, i.e. \( n \). Signals with a smaller strength (i with \( 1 \leq i < n \) ) attached to them are stored values. Notice that, due to the basic idea given above, conflicts can occur only between values with equal strength, and damaging conflicts only at level \( n \).

The set of values in this generalisation then is \( \text{VG} = \cup i : 1 \leq i \leq n : \exists \{ L_i, H_i \} \) , with the order \( \preceq \) given in the Hasse diagram depicted below. Consequently, the set of node-states changes into \( \text{NSTCG} = \text{N} \cdot \text{VG} \), with the order \( \preceq \) as the component-wise order \( \preceq \).

Nodes have a certain capacitance-strength, given by a function \( \text{NS} : \text{N} \rightarrow \text{STR} \). The stored information about the previous state, recorded in the function \( \text{store} \), then changes as follows (compare with def. 4.4):

\[
\text{storeG} \cdot \Gamma \cdot x = \{ X_j : X_j \in \Gamma \cdot x : X_{\text{min} + j} \} , \text{ where } j = \text{NS} \cdot x .
\]

The order \( \preceq \) on \( \text{VG} \):

- **source values**
  - \( \{ L_n, H_n \} \)
  - \( \{ L_{n-1}, H_{n-1} \} \)
  - \( \ldots \)
  - \( \{ L_1, H_1 \} \)
  - \( \emptyset \)

- **stored values**

- **uncharged**

- **uncharged**
The generalisation to multiple capacitance-strengths can be worked out in a similar way as the model presented in the previous sections. The results regarding well-matchedness are similar, e.g., well-matchedness in this generalisation equals WMO. Well-functioning in this generalisation is implied by WF2.

Note that the model presented in the previous sections is a special case of this generalisation, viz. using $m=2$ in the definition of VG.

The following example shows that it can be useful, regarding well-functioning, to choose different capacitances at nodes in order to avoid charge-sharing, and, consequently, to distinguish different capacitance-strengths in the model.

4.4.1 Example

Consider the circuit depicted in fig. 4.5a. In order to avoid charge-sharing, of charges stored in $y$ and $z$, node $z$ — the node that is intended to store information — is given a stronger capacitance. The model presented in the previous sections, which does not consider the differences in capacitance-strengths, shows an occurrence of charge-sharing and its damaging effect on the stored information. The generalised model described above, which does consider these differences, shows that charge-sharing can be avoided in this case, and that the stored charge in node $z$ can be protected. Note that this is particularly important if node $z$ is a gate node.

In the circuit depicted in figure 4.5a, input nodes $x_0$ and $x_1$ are changed from both low to both high and, after a stable state is reached, vice versa. Input node $x_0$, however, is delayed by surrounding circuitry: this is modelled by the assumption $(rd_{-s_0} > rd_{-s_1}) \land (rd_{-s_1} = rd_{-s_2})$. If both inputs are low the circuit reaches stable state $\Pi_0$ (figure 4.5b), and if both inputs are high the circuit reaches stable state $\Pi_2$. States $\Pi_1$ and $\Pi_3$ are intermediate states. If nodes $y$ and $z$ have equal capacitance strengths (or a difference is not considered), the model shows a conflict at capacitance level due to charge-sharing, cf. case A in figure 4.5b. Granting node $y$ a larger capacitance-strength than node $z$ can be modelled (in the generalisation described above) with $m=3$, node $z$ strength 2, and node $y$ strength 1. This is shown in figure 4.5b as case B, where $H_3$ and $L_2$ denote a connection to an $H$-source and a connection to an $L$-source respectively.

![Figure 4.5a](image1)

![Figure 4.5b](image2)
Resistances
Resistances are discussed informally in section 0.2. Here, we informally discuss a way of modelling resistances, its complications, and some of the consequences of including resistances explicitly in our model.

The influence of resistances on delays can be modelled as wire-delays, which are discussed in section 5.2. The resistances we consider are assumed to be so small that a passing signal is able to charge a node and control a switch. On the other hand, resistances are assumed to be so large that a signal that did not pass a resistance overrules a signal that passed a resistance.

As discussed in [Bry1] and [W11], resistances can be given a conductance-strength; the larger the resistance, the smaller its conductance-strength. As above, a strength component is attached to signals. Besides the subdivision of stored values $h$ and $l$ into $H_i$ and $L_i$ with $1 \leq i \leq n$, also the source values $H$ and $L$ are subdivided into $H_i$ and $L_i$ with $m \leq i \leq n$. As before, a value with a larger strength overrides a value with a smaller strength. The conductance-strength of a conducting path can be defined as the minimal conductance-strength of the resistances on the path (with range $i$: $m \leq i \leq n$, and, hence, defined as $n$ if there are no resistances on the path).

The value of a node cannot, however, be determined by considering only the strengths of the paths to the various sources because of possible interaction at intermediate nodes, see ex. 4.42.

As a result of this, the node-state cannot be calculated — in the model extended with resistances — using a simple extension of the response function $R$.

The validity of the usage of conductance-strengths for modelling resistances is explained in [W11]. The problem described above — w.r.t. the paths — is discussed in [Bry1].

Example 4.43 below shows that, using resistances in our model, circuits need not have stable states w.r.t. each source-connection. This means, cf. corollaries 1.34a and 1.40, that results regarding well-matchedness similar to those obtained before, cannot be obtained for an extension of the model as suggested above.

We will not include resistances in our model. We have taken care that not considering resistances in our model does not lead to a too optimistic modelling of circuit behaviour (cf. sect. 0.0 and 0.2). The usage of resistances is restricted mainly to n-mos and p-mos technology (for pull-up and pull-down). Since our model concentrates on C-mos and resistances are hardly used in C-mos, the pessimism of our model due to not including resistances explicitly is acceptable. Example 4.44 shows how resistances can be used fruitfully in C-mos, and (hence) demonstrates the pessimism of our model discussed above.

4.42 Example
Consider the circuit, source-connection combination depicted in figure 4.6a. Let $R1 \geq R2$, and let the conductance-strengths of $R1$ and $R2$ be 1 and 2 respectively. Since the n-switch is conducting in the resulting state (its gate is high), the low source value
passing it will override the high source value passing R2 and, hence, node x will be low with strength 2. Consequently, node y is low with strength 1 in the resulting state. Considering the paths between y and the sources only is not sufficient to determine the resulting value of y, viz., to both types of sources (H and L) a path with strength 1 exists (suggesting a conflict of strength 1 at y?).

4.43 Example
Consider the circuit, source-connection combination depicted in figure 4.6b (from [Wi1]). Switch s can be either conducting or nonconducting. If s is conducting, the high value passing resistance R is overruled by the low value passing x, and, consequently, x has value L_m. If s is nonconducting, the high voltage passing R charges node x to value H_{n-1}. Since both values for x can control s, the circuit will oscillate — within our model — between states \( ((s,1),(x,L_m)) \) and \( ((s,0),(x,H_{n-1})) \). No other feasible states exist.

4.44 Example
Consider the circuit depicted in figure 4.6c. Nodes x_0 and x_1 are the input nodes, and y_2 is the output node. If both inputs are high, the low value passing x_2 and x_3 overrules the weakened signal passing r, and y_0 becomes low. Consequently, node y_2 becomes high. Similarly, if both inputs are low, node y_2 becomes low. If one of the inputs is high and the other is low, the loop \( y_0 \rightarrow y_2 \rightarrow y_1 \) functions as a memory element and retains the value of y_2.

This implementation of the Muller C-element is suggested by Martin in [Ma2]. The right-bottom inverter (input y_2, output y_1) with the resistance r can be implemented as a trickle inverter (TWE), which is a weak inverter preventing y_0 to be floating when y_0 is not connected to a source by the switches x_0, x_1, x_2 and x_3.

\[ \text{figure 4.6a: ex. 4.42} \quad \text{figure 4.6b: ex. 4.43} \quad \text{figure 4.6c: ex. 4.44} \]
4.4 Concluding remarks on chapter 4

In this chapter, the capability of nodes to store charge is discussed. Following the same line of reasoning as in chapters 1 and 3, the model is extended in such a way that charge storage can be handled. Both initial behaviour, with well-matchedness-2, \( WM2 \), as the main notion, and dynamic behaviour, with well-functioning-2, \( WF2 \), as the main notion, are described.

Assumptions on charge storage
The capability of nodes to store charge is modelled by means of a notion of capacitance. The formalisation given of charge storage is based on the following assumptions (cf. sect. 4.0):

(0a) Node capacitances are so large that a stored high or low value at the gate node can keep a switch in its state provided that the state of that switch corresponds to the stored value at its gate node. That is, a stored low value can control a conducting p-switch or a nonconducting n-switch, and, similarly, a stored high value can control a nonconducting p-switch or a conducting n-switch.

(0b) The control of a switch by a stored charge — as is assumed to be (partially) possible in (0a) — must be temporary.

(1a) The stored value at a node is determined by the previous states — including the stored values — of all nodes connected to it (including the node itself). This means, among other things, that charge sharing is possible. Conflicts at capacitance level are assumed to be harmless, that is, they are assumed not to damage the circuitry.

(1b) Node capacitances are so small that once a node is connected to a source, the source value overrides the stored value.

In the formalisation given in the first three sections, different strengths of capacitances are not distinguished.

Initial behaviour
The major result regarding well-matchedness is given in theorem 4.26b; well-matchedness-2 equals the previously defined well-matchedness notions. Similar results as achieved for \( WM0 \) and \( WM1 \) are given in theorems 4.26a, 4.27, and 4.28 (compare with theorems 1.28 and 3.8 (4.26a) and with cor. 1.40 and th. 3.10 (4.27 and 4.28)). They are listed below.

Since stable-2 is an important notion in these theorems, note that, according to lemma 4.22, the stable-2 states equal — abstracted from stored charges — the stable-1 states.
As in chapters 1 and 3, the additional correctness criteria for initial behaviour, viz. all resulting states are conflict-free (CF) and all transitions between resulting states are correct (csn0), lead, besides \( WM2 \cdot C \cdot \gamma \), to correctness criterion: \( CF \cdot \gamma \) (cf. section 4.1 'additional correctness criteria'). This means that, as far as correctness of initial behaviour is concerned, reaction-delay and charge storage are irrelevant (i.e. the total correctness criterion for initial behaviour equals the one formulated in 1.49).

The results for \( WM2 \), as given in theorems 4.26, 4.27, and 4.28, are:

\[
( WM2 = WM1 ) \land ( WM2 = WMO )
\]

and, for \( C \in \text{CIR} \land \gamma \in \text{NST} : \)

\[
WM2 \cdot C \cdot \gamma = ( \text{stable}2 \cdot \gamma \equiv \text{cgdo} )
\]

\[
WM2 \cdot C \cdot \gamma = ( \text{cgdo} \cdot \gamma \wedge \text{stable}2 \cdot \gamma = \{ Q, \Gamma : \text{destor} \cdot ( Q, \Gamma ) = \gamma \wedge \text{coco}1 \cdot ( Q, \Gamma ) \} )
\]

\[
WM2 \cdot C \cdot \gamma = ( \text{feasible}2 \cdot \gamma = \text{stable}2 \cdot \gamma )
\]

Dynamic behaviour

The formalisation of \( WF2 \) is done in a straightforward way in section 4.2. The correctness criteria for dynamic behaviour can be varied to suit ones purposes (cf. remark 4.35).

As demonstrated in examples 4.36 and 4.38, well-functioning can, for a given set of reaction-delay restrictions, be evaluated efficiently. Rather surprisingly, the evaluation of the correctness criterion for initial behaviour is, in general, more difficult than the evaluation of the criterion for dynamic behaviour.

Example 4.36a shows how a minimal set of restrictions can be derived such that well-functioning holds for this set of restrictions.

Generalisations

In section 4.3 we discussed — informally — the generalisation of our model to multiple capacitance-strengths. This generalisation turns out to be relatively straightforward now that we already have included charge storage in our model, and leads to similar results regarding well-matchedness and well-functioning.

Although the basic idea of modelling resistances is similar to that of multiple capacitance-strengths, the actual modelling of resistances, also discussed informally in section 4.3, is much more difficult. Regarding well-matchedness, results similar to those obtained before cannot be achieved for the latter extension (cf. section 4.3 'resistances'). The presented model takes care that the abstraction from resistances does not lead to a too optimistic modelling of circuit behaviour, which explains not including resistances in the model.
CHAPTER 5 PASS-DELAYS AND WIRE-DELAYS

The concept of pass-delay models the time it takes a voltage transition to pass from one pass node of a conducting switch to the other. Wire-delay models the time it takes a voltage transition to pass from one end of a wire to the other. In the model developed in the previous chapters, we assumed pass-delays and wire-delays to be zero (cf. section 0.3, restriction (b)). In this chapter the model is extended such that arbitrary — but non-negative and finite — pass-delays and wire-delays are captured, and that restrictions on them can be expressed. As in chapter 3 (reaction-delays), we do not aim at absolute delay-estimates, but we consider relative delays, that is, we consider possible orders in which the corresponding events can occur.

The approach towards modelling these types of delay is the same as the one used for reaction-delays (ch. 3). The modelling of state transitions in this model is more complicated since the occurrence of pass-delays or wire-delays in a successor state and the succeeding node-state are mutual dependent (explained in section 5.0.0). As a result of this, modelling restrictions on pass-delays and wire-delays also is more complicated than modelling restrictions on reaction-delays.

We start with modelling pass-delays. Once they are included in our model, modelling wire-delays will turn out to be relatively simple.

Chapter 5 is organised as follows.

In the first section the model is extended such that arbitrary pass-delays are captured. The section concentrates on initial behaviour of circuits. The correctness criteria for initial behaviour defined in this section will turn out to equal those defined in the previous chapters.

In the second section the model is extended such that restrictions on pass-delays can be expressed. The notions regarding dynamic behaviour, as defined in section 4.2, are redefined. At the end of this section we conclude that pass-delays and reaction-delays cannot be modelled by one 'reaction-delay like' notion.

The third section shows the strong correspondence between wire-delays and pass-delays. As a result of this correspondence, wire-delays can be modelled in a similar way as pass-delays. The results w.r.t. correctness of circuit behaviour are similar to the results obtained in sections 5.0 and 5.1.

As usual, the final section summarizes and discusses the major results.
5.0 Arbitrary pass - delays; initial behaviour

In the first subsection the notion of pass-delay is analysed, and the modelling of pass-delays is discussed, in particular w.r.t. the definition of state transitions. Once state transitions are (re)defined, the extension of the model w.r.t. initial behaviour is, by now, standard. It is given in the second subsection and leads to a renewed version of well-matchedness. In the third subsection the properties of the redefined notions are investigated and the relation to the old versions is given. The new well-matchedness notion will turn out to equal the old well-matchedness notions. In the last subsection the familiar additional correctness criteria (CF and cso) are reconsidered and a new one — similar to correctness of state transitions for reaction-delays, cso — is defined.

5.0.0 Introduction

The concept of pass-delay models the time it takes a voltage transition to pass from one pass node of a conducting switch to the other. That is, pass-delay models the time of the (de)charging of one pass node of a conducting switch by the other. The delay of this (de)charging is caused by the resistance of the switch and the capacitance of the pass node that is (de)charged.

In our model this means that it may take, due to pass-delays, an — arbitrary but finite — number of steps to transport a change of state of one pass node of a conducting switch to the other pass node of that switch. This number of steps models the pass-delay. During these steps the pass-delay is called active in that switch. A switch that connects its pass nodes is called transporting, that is, a switch is called transporting if it is conducting and the pass-delay is not active in that switch. These concepts are illustrated in the example below.

5.0 Example

Consider the circuit depicted in figure 5.0a below. Let the circuit start in state $\Pi_0$ (fig. 5.0b), with both $s_0$ and $s_1$ conducting, and let the source-connection change in $x$ from (L) to (H) and remain $\emptyset$ in $y$ and $z$. The switches remain conducting, but, once one of the pass nodes of a switch has changed, it can, due to pass delay, take a number of steps

```
x s_0 y s_1 z
H H
```

$\Pi_0$ \{L\} \{L\} \{L\}
$\Pi_1$ \{H\} \{\emptyset\} \{\emptyset\}
$\Pi_2$ \{H\} \{H\} \{\emptyset\}
$\Pi_3$ \{H\} \{H\} \{H\}

figure 5.0a  figure 5.0b
before the switch becomes transporting. If both pass-delays are zero, the circuit goes immediately from state $\Pi_0$ to $\Pi_3$ (cf. fig. 5.0b). If both pass-delays are positive, the circuit visits intermediate states $\Pi_1$ and $\Pi_2$—necessarily in that order (cf. fig. 5.0b)—before reaching stable state $\Pi_3$. The pass-delay is active in $s_0$ in state $\Pi_1$ and it is active in $s_1$ in state $\Pi_2$. Consequently, $s_0$ is transporting in $\Pi_0$, $\Pi_2$, and $\Pi_3$, and $s_1$ is transporting in $\Pi_0$, $\Pi_2$, and $\Pi_3$.

In order to model pass-delays we use countdown functions in a similar way as they are used for reaction-delays (cf. section 3.0). Recall the motivation for using countdown functions given in section 3.0. Countdown functions take care of the finiteness of delays in the definition of state transitions (next), thereby avoiding finite-delay requirements in the definition of stable and feasible states, and enabling the abstraction from delays in the definitions of stable, feasible, and WM. Furthermore, they enable the formalisation of delay restrictions.

In order to formalise state transitions, we will, in a similar way as in section 3.0 (for reaction-delays), extend the states with a reaction-delay counter and a pass-delay counter. Let $\mathcal{Q}_0\Gamma_0 \rho_0 \rho_0$ be the current state, and let $\mathcal{Q}_1\Gamma_1 \rho_1 \rho_1$ be a successor of this state.

The next node-state, $\Gamma_1$, depends on the next switch-state, $Q_1$, and a pass-delay counter, viz., nodes are connected only if a transporting path between them exists. The pass-delay counter used to calculate the next node-state, however, depends on the next node-state itself, viz., the pass-delay counter of a switch is allowed to be positive (the pass-delay is active) only if one of the pass-nodes needs to be (de)charged. We will therefore use the pass-delay counter $p_1$ for the calculation of node-state $\Gamma_1$ (and vice versa).

As we have seen in chapter 3, the reaction-delay counter used for the calculation of the next state can be calculated from the current state, viz., in the restrictions on $Q_1$ the reaction-delay counter $r_0$ is used (cf. R0.i, sect. 3.0 and 4.0). This difference in the use of the two types of delay-counters is due to the initial assumptions (b) and (c) (section 0.3): in the basic model we assumed a positive and uniform reaction-delay and a zero pass-delay. Assuming a positive and uniform pass-delay in the basic model might have avoided the difficulty here ($\Gamma_1$ and $p_1$ depending on each other), but would have caused severe complications in the formalisation in the basic model.

In the remainder of this subsection we formalise the restrictions on $\mathcal{Q}_1\Gamma_1 \rho_1 \rho_1$, in order to be a successor of $\mathcal{Q}_0\Gamma_0 \rho_0 \rho_0$, and discuss the modelling of pass-delays.
Restrictions on $Q_1$

The restrictions on the next switch-state, $Q_1$, are equal to the restrictions given in chapter 4. They are explained in section 4.0.

R0.2 \hspace{1cm} (A \ s : (rc_0 \cdot s = 0) \land \neg CA\cdot Q_0 \cdot \Gamma_0 \cdot s : consistent \cdot \Gamma \cdot Q_1 \cdot s ) \\
R0.3 \hspace{1cm} (A \ s : (rc_0 \cdot s > 0) \lor CA\cdot Q_0 \cdot \Gamma_0 \cdot s : Q_1 \cdot s = Q_0 \cdot s )

Restrictions on $\Gamma_1$

A switch, say $s$, is transporting if it is conducting and the pass-delay is not active, that is, if $Q_1 \cdot s \land (pc_1 \cdot s = 0)$. Two nodes, say $x$ and $y$, are connected if a transporting path between them exists. That is, nodes $x$ and $y$ are connected if $cp\cdot Q_1 \land Z\cdot pc_1\cdot [x,y]$, where the Zero-function $Z \in DC \rightarrow SW \rightarrow B$ is defined by $Z\cdot pc_1\cdot s = (pc_1\cdot s = 0)$. Consequently, the next node-state is defined by restriction R1.2 below (compare with R1.1).

R1.2 \hspace{1cm} \Gamma_1 = R\cdot Q_1 \land store\cdot Q_0 \cdot \Gamma_0 \cdot (Q_1 \land Z\cdot pc_1 )

Restrictions on $rc_1$

The restrictions on the next reaction-delay counter need not be changed. They are explained in section 3.0.

R2.0 \hspace{1cm} (A \ s : consistent \cdot \Gamma \cdot Q_1 \cdot s : rc_1 \cdot s = 0 ) \\
R2.1 \hspace{1cm} (A \ s : \neg consistent \cdot \Gamma \cdot Q_1 \cdot s \land (rc_0 \cdot s > 0) : rc_1 \cdot s = rc_0 \cdot s - 1 ) \\
R2.2 \hspace{1cm} (A \ s : \neg consistent \cdot \Gamma \cdot Q_1 \cdot s \land (rc_0 \cdot s = 0) : rc_1 \cdot s \in \mathbb{N} )

Before we formalise pass-delay by giving the restrictions on $pc_1$, we formulate the assumptions on which our modelling of pass-delays is based, give a motivation for each assumption, and discuss their most striking features.

Assumptions on pass-delays

(0a) A pass-delay can be active in a switch only if the switch is conducting.

(0b) A pass-delay cannot be active in a switch if the pass nodes have the same value.

(1) A pass-delay can be active in a switch only if the value that needs to be transported by that switch is $\{L\}$ or $\{H\}$. This means that we assume — in our model — that the pass delay is zero if the value that needs to be transported is $\{L,H\}$ or a stored charge.

(2) A pass-delay must have an initial cause. This cause can either be that the switch becomes conducting, or that the value that needs to be transported has changed. For the latter case, we also assume that a pass-delay is not possible if the transported value was $\{L,H\}$ in the preceding state.
Motivation of assumptions (0a) and (0b)

Assumptions (0a) and (0b) result from the statement (earlier in this section): 'a pass-delay can be active if one of the pass nodes of a conducting switch needs to be (de)charged by the other pass node'. An interpretational problem arises if one of the pass nodes has a source value and the other one has a stored charge of the same type, that is, for a switch $s$ with $p_{s,s} = \{x,y\}$, if $(\Gamma_1^\top x \in \{\{L\},\{H\}\}) \land (\text{store}_{\Gamma_1^\top y} = \Gamma_1^\top y)$. On the one hand, one can argue that, since the pass nodes have the same type of charge (and, hence, neither of them needs to be (de)charged by the other), a pass-delay should not be allowed. This means that assumptions (0a,b) are a weaker version of the statement mentioned above. On the other hand, one can argue that the charges need not be equal since the stored charged can be weakened, and, hence, a pass-delay is possible in this case. The motivation for allowing, in our model, pass-delays in such a case is that it is a more liberal modelling of the behaviour, and, this way, our model allows both interpretations. Namely, the pass-delay can be chosen to be zero in the problem cases by a suitable choice of delay restrictions (as we will show in remark 5.19). In remark 5.1a below we show how the model can be changed such that pass-delays are not allowed in the cases mentioned above.

Motivation of assumption (1)

The motivation of assumption (1) is that not allowing pass-delays if the transporting value is $\{L,H\}$ or a stored charge can only lead — in some cases — to a pessimistic modelling of circuit behaviour (which is allowed: cf. section 0.0), and simplifies the modelling of pass-delays. The intuitive argument for this resulting pessimism is that the occurrence of undesirable events — like conflicts ($\{L,H\}$ as well as $\{I,H\}$) — is speeded up. This argument is explained further in appendix C. In fact, we will prove in this appendix that the restriction on pass-delays (to $\{L\}$ and $\{H\}$) does not change the correctness criterion for initial behaviour; it does not change the resulting states for initial behaviour if this behaviour is correct; and it does not weaken the correctness criterion for dynamic behaviour.

Motivation of assumption (2)

A pass-delay can start being active in a switch together with the need of one pass node to be (de)charged by the other. This can be the case if either the switch becomes conducting or if the state of one of the pass nodes has changed into $\{L\}$ or $\{H\}$ (using also assumption (1)). This is formulated in the first part of assumption (2) (cf. remark 5.1d below). The second part is added mainly because (A) it simplifies the formalisation of pass-delays, and (B) it simplifies (the use of) the model, since calculating with positive pass-delays is complicated, and, hence, the fewer cases the pass-delay is positive, the better. It restricts the behaviour only if the preceding state is not correct (more specifically: is not conflict-free).
Restrictions on pc:
Let \( s \) be a switch with pass nodes \( x \) and \( y \).
Assumptions (0a) and (0b) are expressed in restriction R3.0 below.
The value that needs to be transported by \( s \) is \( \Gamma_1 \cdot x \equiv \Gamma_1 \cdot y \). Assumption (1) can, therefore, be expressed as restriction R3.1 below.
Assumption (2) can be expressed as (directly "translated" from the text):
\[
(pc_{1:s} > 0 \land pc_{0:s} = 0) \Rightarrow (Q_0 \supset \land Q_{1:s})
\]
\[
\lor ((\Gamma_1 \cdot x \equiv \Gamma_1 \cdot y \neq \Gamma_0 \cdot x \equiv \Gamma_0 \cdot y) \land (\Gamma_0 \cdot x \equiv \Gamma_0 \cdot y \neq [L \cdot H]))
\]
Using restrictions R3.0 and R3.1 this expression reduces to:
\[
(pc_{1:s} > 0) \land (pc_{0:s} = 0) \Rightarrow (Q_0 \supset \lor (\Gamma_1 \cdot x \equiv \Gamma_1 \cdot y \neq \Gamma_0 \cdot x \equiv \Gamma_0 \cdot y))
\]
Restriction R3.2 expresses assumption (2) and the countdown property of the pass counter.
For all \( s, x, y \) such that \( SW \cdot s \land (pn \cdot s = [x,y]) \):
R3.0
\[
(pc_{1:s} > 0) \Rightarrow (Q_1 \cdot s \lor (\Gamma_1 \cdot x \neq \Gamma_1 \cdot y))
\]
R3.1
\[
(pc_{1:s} > 0) \Rightarrow (\Gamma_1 \cdot x \equiv \Gamma_1 \cdot y \in \{L, H\})
\]
R3.2
\[
(pc_{1:s} > 0) \Rightarrow ((pc_{0:s} > 0 \land pc_{1:s} = pc_{0:s} - 1)
\]
\[
\lor (pc_{0:s} = 0 \land (Q_0 \supset \lor (\Gamma_1 \cdot x \equiv \Gamma_1 \cdot y \neq \Gamma_0 \cdot x \equiv \Gamma_0 \cdot y)))
\]
Equally, in order to be a delay counter, \( pc_1 \) must be restricted by: \( pc_{1:s} \in \mathbb{N} \).
In remark 5.1c we show how the restrictions on \( pc_1 \), including DC-\( pc_1 \), can be written in a similar form as the restrictions on \( rc_1 \).

5.1a Remark
Not allowing pass-delays if one of the pass-nodes of a conducting switch has a source value and the other one a stored charge of the same type (cf. "Motivation of assumptions (0a) and (0b)"), can be modelled by replacing R3.0 by:
\[
(pc_{1:s} > 0) \Rightarrow (Q_1 \cdot s \land (\text{store} \cdot \Gamma_1 \cdot x \neq \text{store} \cdot \Gamma_1 \cdot y))
\]

5.1b Remark
Let \( ((Q_1, \Gamma_1), rc_1, pc_1) \) satisfy R1.2 (for some \( \Gamma_0 \in \text{NSTC} \)). Let \( SW \cdot s \land (pn \cdot s = [x,y]) \). Then:
true
\[
= \{ R1.2, \text{prop. 4.17c, and def. 4.16 (covol)} \}
\]
\[
\Rightarrow \{ \text{def of Z and def. 4.16 (consistent)} \}
\]
\[
(pc_{1:s} = 0) \Rightarrow \text{consistent} \cdot \Gamma_1 \cdot Q_1 \cdot s
\]
If \( ((Q_1, \Gamma_1), rc_1, pc_1) \) also satisfies R3.0, we can conclude (using def. 4.16 (consistent) again):
\[
(pc_{1:s} > 0) = (Q_1 \cdot s \land (\Gamma_1 \cdot x \neq \Gamma_1 \cdot y))
\]
that is:
\[
(pc_{1:s} = 0) = \text{consistent} \cdot \Gamma_1 \cdot Q_1 \cdot s
\]
5.1c Remark

Restriction R3.0 equals restriction r3.0 below (use definitions 4.16 and 1.15 (consistent)). Now using also R1.2 (see remark 5.1b above), the restrictions R3.1 and R3.2 can be written as r3.1 and r3.2 below. In order for p_{c1} to be a delay counter, we included the restriction p_{c1} \in W in the second part of r3.2. Notice that, on account of these restrictions, DC-p_{c0} = DC-p_{c1} holds.

r3.0 \quad (A \ s : consistent \ \Gamma_{1}\cdot Q_{1}s : p_{c1}s = 0 )

r3.1 \quad (A \ s : \neg consistent \ \Gamma_{1}\cdot Q_{1}s : TV_{\Gamma_{1}s} \in \{L,H\})

r3.2 \quad (A \ s : \neg consistent \ \Gamma_{1}\cdot Q_{1}s \land (p_{c0}s > 0 ) : p_{c1}s = p_{c0}s - 1 ) \land

\quad (A \ s : \neg consistent \ \Gamma_{1}\cdot Q_{1}s \land (p_{c0}s = 0 )

\quad : (p_{c1}s \in W ) \land (\neg Q_{0}s \land (TV_{\Gamma_{1}s} \notin TV_{\Gamma_{0}s}) )

where TV_{\Gamma_{1}s} denotes the value that needs to be transported by s in state \Gamma and is defined as:

TV_{\Gamma_{1}s} = (\forall z : z \in pn_{s} : \Gamma_{1}z )

The form of the restrictions on the pass-delay counter (r3.1) now strongly resembles the form of the restrictions on the reaction-delay counter (R2.1).

5.1d Remark

The second cause for the pass-delay to become active in a switch as described in assumption (2), i.e., 'the value that needs to be transported has changed' can be rephrased as 'at least one of the pass nodes has changed its state', with, of course, the same restriction on the transported value in the preceding state (see ass. (2)). This is explained below.

For the intended use of state transitions, the current state, ((Q_{0}, \Gamma_{0}),r_{c0},p_{c0}), will be a result of a state transition and, hence, will satisfy (use R1.2):

(E \ \Gamma_{1} : NSTC-\Gamma_{1} : \Gamma_{0} = R.(\gamma \equiv store_{\Gamma} -(Q_{0} \land Z-p_{c0} ) )).

As a result of this, the current state satisfies (use def. R, cf. remark 5.1b):

(p_{c0}s = 0 ) \land Q_{0}s \equiv \Gamma_{0}s = \Gamma_{0}y .

A successor of the current state, say ((Q_{1}, \Gamma_{1}),r_{c1},p_{c1}), satisfies, on account of R3.1:

(p_{c1}s > 0 ) \Rightarrow (\Gamma_{1}x \equiv \Gamma_{1}y = \Gamma_{1}x) \lor (\Gamma_{1}x \equiv \Gamma_{1}y = \Gamma_{1}y). Attention is also paid to the latter two implications, restriction R3.2 equals:

(p_{c1}s > 0 ) \Rightarrow (p_{c0}s > 1 ) \land p_{c1}s = p_{c0}s - 1)

\lor (p_{c0}s = 0 ) \land (\neg Q_{0}s \lor (E \ z : z \in pn_{s} : \Gamma_{1}z \notin \Gamma_{0}z ))

Consequently, we demonstrated that the two formulations of the second cause mentioned in assumption (2) as given above are equal within the model.

\eta (end remark 5.1)
5.0.1 Extension of the model

In this subsection, the modelling of initial behaviour is redefined in such a way that arbitrary pass-delays are captured. The line of reasoning is similar to the one in chapter 4. The properties of the redefined notions are investigated in the next subsection.

In order to define the next-state function, states are extended with delay-counters (cf. sect. 5.0.0). The set of next states of an extended state $(\Pi, rc, pc)$ is defined as the set of extended states that satisfy the restrictions formulated in the previous subsection. In the definition below restrictions $r3$ are used instead of restrictions $R3_i$ (cf. remark 5.1c).

5.2 Definition  

\[ \text{next}^3 - 3 \in \text{NST} \rightarrow (\text{ST} \times \text{DC} \times \text{DC}) \rightarrow (\text{NST} \times \text{DC} \times \text{DC}) \]  

is defined by:

\[ \text{next}^3 - 3 (Q_0, \Gamma_0, rc_0, pc_0, ((Q_1, \Gamma_1), rc_1, pc_1)) = \]

\[ \begin{align*}
(A \ s : & (rc_0, s = 0) \land \neg \text{CA}(Q_0, \Gamma_0, s) \land \text{consistent}(\Gamma_0, Q_1, s) \\
\land & (A \ s : (rc_0, s > 0) \lor \text{CA}(Q_0, \Gamma_0, s) \land Q_1, s = Q_0, s) \\
\land & (\Gamma_1 = R \cdot (y \in \text{store}(\Gamma_0)) (Q_1 \land Z \cdot pc_1)) \\
\land & (A \ s : \text{consistent}(\Gamma_1, Q_1, s) \land rc_1, s = 0) \\
\land & (A \ s : \neg \text{consistent}(\Gamma_1, Q_1, s) \land (rc_0, s > 0) \lor rc_1, s = rc_0, s - 1) \\
\land & (A \ s : \neg \text{consistent}(\Gamma_1, Q_1, s) \land (rc_0, s = 0) \lor rc_1, s \in N) \\
\land & (A \ s : \text{consistent}(\Gamma_1, Q_1, s) \land pc_1, s = 0) \\
\land & (A \ s : \neg \text{consistent}(\Gamma_1, Q_1, s) \land (pc_0, s > 0) \lor pc_1, s = pc_0, s - 1) \\
\land & (A \ s : \neg \text{consistent}(\Gamma_1, Q_1, s) \land (pc_0, s = 0) \\
\ & \quad : (pc_1, s \in N) \land (\neg Q_0, s \lor (TV \cdot \Gamma_1, s \notin TV \cdot \Gamma_0, s))) \\
\end{align*} \]

where

\[ Z \in \text{DC} \land SW = B \quad \text{and} \quad TV \in \text{NSTC} \land SW = (\text{store}(\{L, H\}) \cup \text{store}(\{L, H\})) \]  

are defined by:

\[ Z \cdot pc_0, s = (pc_0, s = 0) \quad \text{and} \quad TV \cdot \Gamma, s = (\{z : z \in \text{pc}_1, s : \Gamma, z\}) \]

Stable and feasible states can be defined in a similar way as in the previous chapters (cf. definitions 3.2 and 4.8). Recall that the delay-counters are used only for defining the next-state function; they are abstracted from in the definitions below.

5.3 Definition  

\[ \text{stable}^3 \text{ and } \text{feasible}^3, \text{ both of type } \text{NST} \rightarrow \Sigma, \text{ are defined by:} \]

\[ \text{stable}^3 - \gamma - \Pi = (B \cdot rc \cdot pc : DC \cdot rc \land DC \cdot pc : \text{next}^3 - 3 (\Pi, rc, pc)) \]

\[ \text{feasible}^3 - \gamma - \Pi = (B \cdot rc \cdot pc : DC \cdot rc \land DC \cdot pc : (\text{next}^3 - 3)^4 (\Pi, rc, pc)) \]
Recall (from chapter 4) that a circuit, source-connection combination is well-matched if (informally) every cycle of feasible states is completely gate defined-1 in all elements and satisfies the 'finite capacitance-control requirement' defined in chapter 4 (def. 4.13 (FCA0)).

Cycles of feasible states are redefined in the notion Feasible State List 3, FSL3. A state list $\Psi$ is a cycle of feasible states, expressed as $FSL3 \cdot \gamma \cdot \Psi$, if all successors in $\Psi$ are, combined with suitably chosen delay-counters, successors w.r.t. $next3 \cdot \gamma$.

FSL3 and WM3 are defined in definitions 5.4 and 5.5 below. For comparison, see definitions 4.11 and 4.15.

5.4 Definition {feasible state list 3}
$FSL3 \in NST \to L_{+}(ST1) \to B$ is defined by:
$FSL3 \cdot \gamma \cdot \Psi = (\exists r, p \cdot L_{+}(DC) \cdot r \land L_{+}(DC) \cdot p \land (\# \cdot r = \# \cdot \Psi) \land (\# \cdot p = \# \cdot \Psi))$
\[
\land (A : 0 \leq i < \# \cdot \Psi : next3 \cdot \gamma(\Psi_{i}, r_{i}, p_{i}) \cdot (\Psi_{i+1}, r_{i+1}, p_{i+1}))
\]
where $\Psi_{i}, r_{i}, p_{i}$ for $i = \# \cdot \Psi$ is defined as $\Psi_{0}, r_{0}, p_{0}$.

5.5 Definition {well-matched 3}
$WM3 \in CIR \to NST \to B$ is defined by:
$WM3 \cdot C \cdot \gamma = (FSL3 \cdot \gamma \subseteq (FCA0 \cap cd1L))$

5.0.2 Properties of WM3

In this section we investigate the properties of the newly defined notions and the relation between them and their counterparts from the previous chapters. The direction given to the investigation of properties of WM3 is inspired by the results for previous well-matchedness notions.

The relation between next2 and next3 is given in the following lemma. If the pass counter of the next state equals 0, that is, if the pass-delays are chosen to be inactive in all switches, then the state transition in this model equals the state transition in the previous model, which assumed a zero pass-delay (5.6b). Lemma 5.6a expresses remark 5.1b.

5.6 Lemma
For all $\Pi_{0}, \Pi_{1} \in ST1$ and $rc_{0}, rc_{1}, pc_{0}, pc_{1} \in DC$:
\[a \quad next3 \cdot \gamma(\Pi_{0}, rc_{0}, pc_{0})(\Pi_{1}, rc_{1}, pc_{1}) \supseteq (A : SW : consistent1 \cdot Q_{1} \cdot s = (pc_{1}, s = 0)\]
\[b \quad next2 \cdot \gamma(\Pi_{0}, rc_{0})(\Pi_{1}, rc_{1}) = next3 \cdot \gamma(\Pi_{0}, rc_{0}, pc_{0})(\Pi_{1}, rc_{1}, 0)\]
For the proof of 5.6a we refer to remark 5.1b. The proof of 5.6b is given below.
Proof of 5.6b

The restrictions used for defining \( \text{next2} \) are: \( R0.2, R0.3, R1.1, R2.0, R2.1, \) and \( R2.2. \) The restrictions used for defining \( \text{next3} \) are: \( R0.2, R0.3, R1.2, R2.0, R2.1, R2.2, R3.0, R3.1, \) and \( R3.2. \) Restriction \( R1.2 \) with \( \text{pc}_1 = 0 \) equals \( R1.1, \) since \( (Q \land Z \cdot 0) = Q. \) Observing that restrictions \( R3.0, R3.1, \) and \( R3.2 \) with \( \text{pc}_1 = 0 \) equal \( \text{true}, \) completes the proof.

\( \Box \)

In lemma 5.7a and 5.7b — quite similar — properties of \( \text{stable2} \) and \( \text{stable3} \) are given. Lemma 5.7c gives the relation between \( \text{stable2} \) and \( \text{stable3}. \)

5.7 Lemma

a \( (A \gamma, \Pi : \text{NST-}\gamma \land \text{ST1-}\Pi : \text{stable3-}\gamma, \Pi = \text{next3-}(\Pi,0,0)-(\Pi,0,0)) \)

b \( (A \gamma, \Pi : \text{NST-}\gamma \land \text{ST1-}\Pi : \text{stable2-}\gamma, \Pi = \text{next2-}(\Pi,0)-(\Pi,0)) \)

c \( \text{stable3} = \text{stable2} \)

Proof

a Let \( \text{NST-}\gamma, \text{ST1-}(Q, \Gamma), \) and \( \text{stable3-}\gamma(Q, \Gamma). \) On account of definition 5.3 (\( \text{stable3} \)) delay counters exist, say \( \text{rc} \) and \( \text{pc}, \) such that: \( \text{next3-}\gamma((Q, \Gamma), \text{rc}, \text{pc})=((Q, \Gamma), \text{rc}, \text{pc}). \)

Restrictions \( R2.0, R2.1, \) and \( R3.2 \) of def. 5.2 (\( \text{next3} \)) equal, for this case:

\[
(A \gamma, \Pi : \text{consistent}, \Gamma \cdot Q \cdot s : \text{rc} \cdot s = 0) \land \\
(A \gamma, \Pi : \text{consistent}, \Gamma \cdot Q \cdot s : \text{rc} \cdot s > 0 : \text{rc} \cdot s = \text{rc} \cdot s - 1) \land \\
(A \gamma, \Pi : \text{SW-}\gamma : (\text{pc} \cdot s > 0) \land (\text{pc} \cdot s = \text{pc} \cdot s - 1))
\]

From which directly follows that \( (\text{rc} = 0) \land (\text{pc} = 0). \)

b Similar to the proof of 5.7a.

c Let \( \text{NST-}\gamma \land \text{ST1-}\Pi. \) Then:

\[
\text{stable3-}\gamma, \Pi
= \{\text{lemma 5.7a}\}
\]

\[
\text{next3-(}\Pi,0,0)-(\Pi,0,0)
= \{\text{lemma 5.6b}\}
\]

\[
\text{next2-(}\Pi,0)-(\Pi,0)
= \{\text{lemma 5.7b}\}
\]

\[
\text{stable2-}\gamma, \Pi
\]

\( \Box \) (end proof 5.7)

Lemma 5.8a confirms the intended relation between \( \text{FSL3-}\gamma \) and \( \text{feasible3-}\gamma \) (comparable with lemma 4.12). The relation between feasible state (lists) defined in this chapter and those defined in the previous chapter is given in lemmas 5.8b and 5.8c.
5.8 Lemma

For all \( \gamma \in \text{NST} \):

a \( \left( \Lambda \Psi : FSL2-\gamma ; \Psi : \{ i : 0 \leq i < \# \Psi ; \Psi_i \} \right) \) = feasible3-\( \gamma \)

b \( FSL2-\gamma \subseteq FSL3-\gamma \)

c \( \text{feasible2-} \gamma \subseteq \text{feasible3-} \gamma \)

The proof of 5.8a is similar to the proof of lemma 4.12 (the counterpart of 5.8a for FSL2 and feasible3). Lemma 5.8b follows from lemma 5.6b using definitions 4.11 (FSL2) and 5.4 (FSL3). Lemma 5.8c follows directly from lemmas 5.8a, b and 4.12.

The properties of WM3 given in lemma 5.9 follow easily from the previous lemmas.

5.9 Lemma

a \( (A \in \text{C} \wedge \text{NST-} \gamma : WM3-\text{C-} \gamma \Rightarrow WM2-\text{C-} \gamma ) \)

b \( (A \in \text{C} \wedge \text{NST-} \gamma : WM3-\text{C-} \gamma \Rightarrow (\text{stable3-} \gamma \subseteq cgdl)) \)

Proof

Let \( \text{CIR-C} \wedge \text{NST-} \gamma \). Then:

\[
WM3-\text{C-} \gamma = \{ \text{def. 5.5 (WM3)} \} \\
FSL3-\gamma \subseteq (FCA0 \cap cgdl) \\
\quad \{ \text{lemma 5.8b} \} \\
FSL2-\gamma \subseteq (FCA0 \cap cgdl) \\
\quad \{ \text{def. 4.15 (WM2)} \} \\
WM2-\text{C-} \gamma \\
\quad \{ \text{theorem 4.26a} \} \\
\quad \text{stable2-} \gamma \subseteq cgdl \\
\quad \{ \text{lemma 5.7c} \} \\
\quad \text{stable3-} \gamma \subseteq cgdl
\]

\( \Box \)

Similar as in the previous chapters, we have investigated whether the converse of property 5.9b holds. It turns out to hold and is given in theorem 5.10 below. Theorem 5.10 is the counterpart of theorems 1.28, 3.8, and 4.26a. It is an important theorem, since it enables us to conclude equality of WM3 and the well-matchedness notions from the previous chapters (corollary 5.11). Although the proof of theorem 5.10 (given in appendix B2) has a similar construction as the proofs of theorems 1.28 and 3.8 (appendices B0 and B1), it is much more complicated.
5.10 Theorem

\((A, C, \gamma : \text{CIR-C} \land \text{NST-\gamma} : \text{WM3-C-\gamma} = (\text{stable3-\gamma} = c_{\text{gdo}}))\)

The proof of theorem 5.10 is given in appendix B2.

5.11 Corollary

\((\text{WM3 = WM2}) \land (\text{WM3 = WM1}) \land (\text{WM3 = WM0})\)

Corollary 5.11 follows directly from th. 5.10, th. 4.26, and lemma 5.7c.

The counterpart of cor. 1.40, th. 3.10, and theorems 4.27 and 4.28 is given in th. 5.12 below. Notice that, as in the previous chapters, the resulting states are, in case \(\text{WM3-C-\gamma}\) holds, all stable and equal — abstracted from stored values — to \(\gamma_+\) (follows from th. 5.12).

5.12 Theorem

For all \(C \in \text{CIR}\) and \(\gamma \in \text{NST}\):

a. \(\text{WM3-C-\gamma} = (c_{\text{gdo}} = \gamma_+ \land \text{stable3-\gamma} = (Q, I) : (\text{destore}(Q, I) = \gamma_+) \land c_{\text{gdo}}(Q) : (Q, I))\)

b. \(\text{WM3-C-\gamma} \Rightarrow (\text{feasible3-\gamma} = \text{stable3-\gamma})\)

Theorem 5.12a follows from cor. 5.11, th. 4.27, and lemma 5.7c. Theorem 5.12b follows from th. 5.12a and lemma B2 (in appendix B2).

5.0.3 Additional correctness criteria

Besides the familiar correctness criteria, \(CF\) and \(cst0\), a new one is introduced. This new correctness criterion is called \(cst1\), which stands for correct state transition 1, and corresponds to \(cst0\) for reaction-delays. This additional criterion turns out not to count for the total correctness criterion for initial behaviour, which then equals the one defined in the previous chapters.

For a state \((Q, I) \in \text{ST1}\), we have seen that \(\text{"consistent0-Q-s"}\) corresponds to the reaction-delay is active in \(s\), and, similarly, that \(\text{"consistent1-Q-s"}\) corresponds to the pass-delay is active in \(s\). In section 1.4 we argued that as long as the reaction-delay is active in switch \(s\), the cause of the delay — being the value of the gate node of \(s\) — must be invariant. This is expressed in correctness criterion \(cst0\). A similar argument as given in section 1.4 leads to the conclusion that as long as the pass-delay is active in a switch, the cause for this delay must also be invariant. This means that as long as the pass-delay is active in a switch, the switch must remain conducting and the value that needs to be transported by that switch must remain unchanged. This criterion is called \(cst1\) and is defined below. Notice the resemblance with \(cst0\) (def. 4.29).
In definition 5.13b, the criteria \( csi0 \) and \( csi1 \) are combined in a criterion on state lists.

### 5.13 Definition

a. \( csi1 \in ST1 \rightarrow ST1 \rightarrow B \) is defined by:

\[
\text{csi1} \cdot (Q_0, \Gamma_0) \\
\quad \\
= (A \ s : \text{consistent} \cdot \Gamma_0 \cdot Q_0 \ s : (Q_0 \ s = Q_1 \ s) \wedge (TV \cdot \Gamma_0 \ s = TV \cdot \Gamma_1 \ s))
\]

b. \( csiL \in (L \cdot (ST1)) \cup L \cdot (ST1)) \rightarrow B \) is defined by:

\[
\text{csiL} \cdot \Psi = (A \ i : 0 \leq i < k : \Psi : csi0 \cdot \Psi_1, \Psi_{stl} \wedge csi1 \cdot \Psi_1, \Psi_{stl})
\]

Let \( C \in CIR \) and \( \gamma \in NST \).

The correctness criteria for initial behaviour are 'well-matchedness', 'all feasible states are conflict-free', and 'all state transitions between these feasible states are correct (both types)'. That is, the total correctness criterion for initial behaviour is (cf. def. 5.5 (WF5)):

\[
FSL3 \cdot \gamma \subseteq (cgdl \cap FCA0 \cap CFL \cap csiL)
\]

Let \( \Pi \in ST1 \) satisfy \( WM3 \cdot C \cdot \gamma \wedge \text{feasible3} \cdot \gamma \cdot \Pi \). On account of theorem 5.12 state \( \Pi \) satisfies\( stable3 \cdot \gamma \cdot \Pi \wedge cgd0 \cdot \Pi \) (use def. 4.9 (cgdl)). This means that the only successor of \( \Pi \) is \( \Pi \) itself (use def. 5.6 (next)). Since \( destore-\Pi = \gamma_a \) (th. 5.12a) and \( csi0 \cdot \Pi \wedge csi1 \cdot \Pi \) holds (use def. 4.28 (csi0) and 5.13a (csi1)), the additional correctness criterion for initial behaviour, besides \( WM3 \cdot C \cdot \gamma \), is \( CF \cdot \gamma_a \) (use prop. 4.30a). That is:

\[
(FSL3 \cdot \gamma \subseteq (cgdl \cap FCA0 \cap CFL \cap csiL)) = (WM3 \cdot C \cdot \gamma \wedge CF \cdot \gamma_a)
\]

Notice that this result equals the results in chapters 1, 3, and 4 (cf. sections 1.5, 3.1 (end), and 4.1 ('Additional correctness criteria') and use corollary 5.11).

### 5.1 Restricted pass-delays; dynamic behaviour

In the previous section we considered arbitrary pass-delays and the correctness of initial behaviour. In this section we consider restricted pass-delays and the correctness of dynamic behaviour. For similar reasons as for reaction-delays, we need to be able to express restrictions on pass-delays in order to give a useful modelling of dynamic behaviour. Due to the mutual dependence of the succeeding node-state and the succeeding pass-delay counter (cf. sect. 5.0.0), the modelling of restrictions on pass-delays (sect. 5.1.0) is more complicated than the modelling of restrictions on reaction-delays (sect. 3.2). Once restricted pass-delays are modelled, the extension of the model w.r.t. dynamic behaviour (sect. 5.1.1) can be done in a similar way as in section 4.2. In section 5.1.2 we conclude that the two types of delay cannot be modelled by one 'reaction-delay like' notion.
5.1.0 Modelling restrictions on pass-delays

In this subsection the modelling of restrictions on pass-delays and the consequences of such restrictions for well-matchedness are studied.

The following example demonstrates that the functioning even of certain small and simple circuits is not 'well' if the pass-delays are not restricted.

5.14 Example

Consider the circuit depicted in figure 5.1a starting in the top-most state depicted in figure 5.1b. Let the source-connection change to \{(x_1,y_0,0),(y_1,0,z_0,0)\}. The permanent sources depicted in fig. 5.1a remain the same, and, consequently, switches \(s_0\) and \(s_1\) stay conducting. The resulting state lists are schematically depicted in figure 5.1b. They all lead to the bottom-most state in fig. 5.1b, which is stable. In order to avoid the two intermediate states with a (temporary) conflict in node \(z\), the delays must be such that switch \(s_2\) does not become conducting before switch \(s_3\) has become nonconductive. That is, the reaction-delay, say \(rd \in DC\), and the pass-delay, say \(pd \in DC\), must be restricted by: \(pd \cdot s_0 + rd \cdot s_2 \geq pd \cdot s_1 + rd \cdot s_3\), where \(pd\) is used in a similar way as \(rd\) (see sect. 3.2).

figure 5.1a

In section 3.2, restrictions on reaction-delays are modelled by replacing R2.2 in the definition of \(next\) by R2.2" below (cf. section 3.2), where \(rd \in DC\) denotes the reaction-delay.

R2.2" \[(A s : \neg consistent0 \cdot \Gamma_1 \cdot Q_1 \cdot s \land (rc_0 \cdot s = 0) : rc_1 \cdot s = rd \cdot s)

Similarly, we can try to model restrictions on pass-delays by replacing R3.2 (or r3.2, cf. rem. 5.1c) in the definition of \(next\) by R3.2" defined below, where \(pd \in DC\) denotes the intended pass-delay.

R3.2" \[(A s : pc_1 \cdot s > 0 : ((pc_0 \cdot s > 1) \land (pc_1 \cdot s = pc_0 \cdot s - 1))

\lor ((pc_0 \cdot s = 0) \land (\neg Q_0 \cdot s \lor (TV \cdot \Gamma_1 \cdot s \not\subset TV \cdot \Gamma_0 \cdot s)) \land (pc_1 \cdot s = pd \cdot s))\)
Example 5.15a below demonstrates that, although this is a good first step, it is not sufficient to model restrictions on pass-delay. This is due to the following difference between the reaction-delay counter and the pass-delay counter. The reaction-delay counter is forced to count back by restriction R2.1, provided that the state of the gate is stable (the switch-state does not change on account of R0.3). The pass-delay counter is not forced to count back; it can be chosen to be 0 (cf. restrictions R3.1). In the previous section this did not lead to problems, since we considered arbitrary pass-delays.

5.15 Example

a Consider the circuit depicted in fig. 5.2a below. Let the starting state be $\Pi_0$ (fig. 5.2b) and let the source-connection become $\gamma = \{(x_0, H), (x_1, H), (x_2, H), (y, H)\}$. Assume that the pass-delay $pd$ satisfies $(pd_{s_0} = 2) \wedge (pd_{s_1} = 1)$. The intended transitions are $\Pi_1$ to $\Pi_{m+1}$ for all $i$: $0 < i < 4$ and $\Pi_4$ to $\Pi_0$. Using, as suggested above, R3.2 instead of R3.2 in the definition of next3 and no further restrictions on $pc_1$ and $\Gamma_1$, all transitions from $\Pi_i$ to $\Pi_i$ with $((0 < i < 4) \wedge (i = 2 \Rightarrow i = 1)) \vee (i = 4 \wedge i = 4)$ are allowed.

b Now consider this circuit (fig. 5.2a) starting in state $\Pi_0$ (fig. 5.2c). Let the new source-connection be $\gamma$ (ex. 5.15a). Assume that the delays and $pd$ satisfy $(pd_{s_0} = 2) \wedge (pd_{s_1} = 1) \wedge (rd_{s_1} = 0)$. The intended transitions are $\Pi_0 \rightarrow \Pi_1 \rightarrow \Pi_2 \rightarrow \Pi_3 \rightarrow \Pi_4 \rightarrow \Pi_5$. Notice that (with R3.2 as above) a transition from $\Pi_1$ to $\Pi_5$ is also possible.

![Diagram](image)

The replacement mentioned above (of R3.2 by R3.2') restricts the number of states in which the pass-delay is active in a switch $s$ to at most $pd_s$, but it does not require that this number equals, if possible, $pd_s$. For a next switch-state $Q_1$ ($Q_1$ satisfying R0.2 and R0.3), the next-state and pass-delay counter, $\Gamma_1$ and $pc_1$, can be chosen in such a way that $pc_{s_1} = 0$, for $SW_s$, even if it is possible to choose $\Gamma_1$ and $pc_1$ such that $pc_{s_1} \in \{pd_s, pc_{s_0} - 1\}$ (cf. ex. 5.15a). Before showing how to restrict $\Gamma_1$ and $pc_1$ in such a way that only the intended transitions are allowed, we eliminate two options for achieving this that seem reasonable but turn out to be insufficient.

Let the next switch-state be $Q_1$. Obviously, $\Gamma_1$ and $pc_1$ must be chosen such that the pass-delay is active ($pc_{s_1} > 0$) whenever possible. That is, $\Gamma_1$ and $pc_1$ must be chosen such that $pc_1$ is
maximal in the set of possible next pass-delay counters. Although this restricts the number of transitions and allows the intended transitions, it still allows other transitions. For instance, in example 5.15a this allows, besides the intended transitions, the transition $\Pi_0 \rightarrow \Pi_3$. Choosing $pc_1$ to be maximal is a necessary, but insufficient requirement.

Another option that seems reasonable is to require that $\Gamma_1$ is minimal in the set of possible next node-states. In example 5.15a this restriction allows only the intended transitions. However, in example 5.15b this restriction allows, besides the intended transition $\Pi_1 \rightarrow \Pi_2$, also the transition $\Pi_1 \rightarrow \Pi_5$ (viz., the set of possible next node-states of $\Pi_1$, which is \{\Pi_2,\Pi_4,\Pi_5\}, has two minimal elements: $\Pi_2$ and $\Pi_5$).

Since pass-delays are possible only for the source values $\{L\}$ and $\{H\}$ (assumption (1), section 5.0.0), 'requiring that the pass-delays are active in as early a stage of value transportation as possible' means 'requiring that the next node state is minimal w.r.t. these source-values'. That is, for a given next switch-state, $\Gamma_1$ and $pc_1$ must be chosen in such a way that $destore-\Gamma_1$ is minimal in the set $\{destore-\Gamma \mid \Gamma$ is a possible next node state $\}$.

We will prove that this set contains a least element (lemma 5.17b), and define the next node-state $\Gamma_1$ in such a way that $destore-\Gamma_1$ is this least element (def. 5.18).

As an auxiliary notion we define the following subset of $next3$ by replacing $R2.2$ and $R3.2$ in the definition of $next3$ by $R2.2^*$ and $R3.2^*$ respectively. With the help of this notion we can, later on, define the next-state function for restricted delays.

5.16 Definition

For $rd \in DC$ and $pd \in DC$, $\mathcal{N}_{rd, pd} \in NST - (ST \times DC \times DC) \rightarrow (ST \times DC \times DC)$ is defined by:

$$\mathcal{N}_{rd, pd}(\langle (Q_0, \Gamma_0), r_{D0}, p_{D0} \rangle, \langle (Q_1, \Gamma_1), r_{D1}, p_{D1} \rangle) =$$

1. $(A s : (r_{D0} > 0) \wedge \neg CA(\langle Q_0, \Gamma_0 \rangle, s : consistent) \wedge (Q_1, \Gamma_1, r_{D1}, p_{D1}))$

2. $(A s : r_{D0} = 0) \wedge (A s : r_{D1} > 0) \vee CA(\langle Q_0, \Gamma_0 \rangle, s : Q_1 = Q_0 \wedge r_{D1} = 0 \wedge p_{D1} = 0)$

3. $(A s : \neg consistent \wedge (Q_1, \Gamma_1, r_{D1}, p_{D1}) \wedge (Q_1, \Gamma_1, r_{D1}, p_{D1}) = 0)$

4. $(A s : \neg consistent \wedge (Q_1, \Gamma_1, r_{D1}, p_{D1}) \wedge (Q_1, \Gamma_1, r_{D1}, p_{D1}) > 0) : (r_{D1} = r_{D0} - 1)$

5. $(A s : \neg consistent \wedge (Q_1, \Gamma_1, r_{D1}, p_{D1}) \wedge (Q_1, \Gamma_1, r_{D1}, p_{D1}) = 0)$

6. $(A s : \neg consistent \wedge (Q_1, \Gamma_1, r_{D1}, p_{D1}) \wedge (Q_1, \Gamma_1, r_{D1}, p_{D1}) = 0) : (r_{D1} = r_{D0} - 1)$

7. $(A s : \neg consistent \wedge (Q_1, \Gamma_1, r_{D1}, p_{D1}) \wedge (Q_1, \Gamma_1, r_{D1}, p_{D1}) = 0)$

8. $(A s : \neg consistent \wedge (Q_1, \Gamma_1, r_{D1}, p_{D1}) \wedge (Q_1, \Gamma_1, r_{D1}, p_{D1}) = 0) : (r_{D1} = r_{D0} - 1)$
5.17 Lemma

Let $(\Pi_0, \Pi_1) \in ST1$ and $(r_{c_0}, r_{c_1}, r_{c_2}, r_{c_3}, p_{c_1}, p_{c_2}) \leq DC$. Then:
$$\kappa_{ad,pd} \gamma ((\Pi_0, r_{c_0}, p_{c_0}), (\Pi_1, r_{c_1}, p_{c_1})) \land \kappa_{ad,pd} \gamma ((\Pi_0, r_{c_0}, p_{c_0}), (\Pi_2, r_{c_2}, p_{c_2}))$$
$$= (r_{c_1} = r_{c_2}) \land (p_{c_1} = p_{c_2})$$

Let $Q_1$ satisfy R0.2 and R0.3 for current state $((Q_0, \Gamma_0), r_{c_0}, p_{c_0}) \in ST1 \times DC \times DC$. The set
$$(\Gamma, r_{c}, p_{c} : \kappa_{ad,pd} \gamma ((Q_0, \Gamma_0), r_{c_0}, p_{c_0}), ((Q_1, \Gamma), r_{c}, p_{c}) : \text{destore-} \Gamma)$$
has a least element.

Lemma 5.17a follows directly from R2.0, R2.1, R2.2", r3.0 and R3.2". Lemma 5.17b is proven below.

Proof of 5.17b

Let $Q_1 \in SST$ satisfy R0.2 and R0.3. Lemma 5.17b is proven by showing that for any pair of elements of the non-empty finite set defined above a third element of this set exists that is smaller than the other two (use theorem A1, appendix A).

Assume: 
$$\{(Q_0, \Gamma), (r_{c_1}, p_{c_1}), (Q_1, \Gamma_2), (r_{c_2}, p_{c_2})\} \subseteq \kappa_{ad,pd} \gamma ((Q_0, \Gamma_0), r_{c_0}, p_{c_0})$$.

Let $\Gamma, \Gamma_2 \in NSTC$ and $p_{c_1}, p_{c_2}, r_{c_3} \in DC$ be defined by:
$$p_{c_3} = \max \{p_{c_1}, p_{c_2}\} \quad \text{for} \ s \in SW$$
$$\Gamma = R(\gamma \cup \text{store-} \Gamma_0)(Q_1 \land Z-p_{c_3})$$
$$p_{c_3} = \begin{cases} p_{c_3} & \text{if } TV-\Gamma_3 \cap \{L-H\} \neq \emptyset \\ \emptyset & \text{if } TV-\Gamma_3 \cap \{L-H\} = \emptyset \end{cases} \quad \text{for} \ s \in SW$$
$$\Gamma_2 = R(\gamma \cup \text{store-} \Gamma_0)(Q_1 \land Z-p_{c_3})$$

We will prove the following properties:

$P0$ $\Gamma \subseteq \Gamma_1$ $\land$ $\Gamma \subseteq \Gamma_2$

$P1$ destore-$\Gamma_3 = \text{destore-} \Gamma$

$P2$ $((Q_0, \Gamma), (r_{c_1}, p_{c_1}, r_{c_3})$ satisfies R3.0, R3.1, and R3.2"

From $P0$ and $P1$ follows that $(\text{destore-} \Gamma_3 \subseteq \text{destore-} \Gamma_1) \land (\text{destore-} \Gamma_3 \subseteq \text{destore-} \Gamma_2)$.

From $P2$, the definitions of $\Gamma_3$ and $r_{c_3}$, and the assumption on $Q_1$ follows that:
$$\kappa_{ad,pd} \gamma ((Q_0, \Gamma_0), r_{c_0}, p_{c_0}), ((Q_1, \Gamma_2), r_{c_1}, p_{c_1})$$

Consequently, proving $P0$, $P1$, and $P2$ completes the proof.

Proof of $P0$ (only for $\Gamma_1$, for $\Gamma_2$ the proof is similar)

true

= $\{\text{def. } p_{c_3}\}$

= $\{A s \in \text{SW} : s \geq p_{c_1} \land s \geq p_{c_2}\}$

= $\{\text{def. } 5.2 (Z) \text{ and calculus}\}$

= $Q_1 \land Z-p_{c_3} \leq Q_1 \land Z-p_{c_1}$

= $\{\text{monotonicity } R \gamma, \text{ definitions } \Gamma_1, \Gamma\}$

$\Gamma \subseteq \Gamma_1$
Proof of P1

From the definitions of $\Gamma$ and $\Gamma_3$ follows (compare proof P0) that $\Gamma \subseteq \Gamma_3$, and, hence, destore-$\Gamma_3 \subseteq \text{destore-}\Gamma_3$. Below, we proof that destore-$\Gamma_3 \subseteq \text{destore-}\Gamma$.

Let $x \in N$. Then:

\[ H \in \Gamma_3 \cdot x \]
\[ = \{\text{def. } \Gamma_3\} \]
\[ (\exists y : cp \cdot (Q_1 \land Z \cdot pc_3) \cdot [x, y] : H \in y \cdot y) \]
\[ \Rightarrow \{\text{def. } \Gamma\} \]
\[ (\exists y : \text{cp} \cdot (Q_1 \land Z \cdot pc_3) \cdot [x, y] : H \in y \cdot y) \]
\[ \Rightarrow \{\text{induction, using note } 0\} \]
\[ H \in \Gamma \cdot x \]

Note 0

Let $x, y \in N$. Then:

\[ H \in \Gamma \cdot x \land \text{bcp} \cdot (Q_1 \land Z \cdot pc_3) \cdot [x, y] \]
\[ \Rightarrow \{\text{definitions of pc_3 and } \Gamma, \text{ using definitions bcp and TV}\} \]
\[ (H \in \Gamma \cdot x) \land \text{bcp} \cdot (Q_1 \land Z \cdot pc_3) \cdot [x, y] \land (H \in y \cdot y) \]

Similar for L.

Proof of P2

Let $s \in SW$. Assume $pc_3 \cdot s > 0$. On account of note 1, we can, without loss of generality, assume that $pc_3 \cdot s = pc_1 \cdot s$. From notes 1, 2, and 3 follows that:

\[ \text{TV} \cdot \Gamma_3 \cdot s = \text{TV} \cdot \Gamma_1 \cdot s \land \langle A, x, y : \text{p} \cdot s = [x, y] : \Gamma_3 \cdot x \neq \Gamma_3 \cdot y \rangle. \]

Using this and the assumption $pc_3 \cdot s = pc_1 \cdot s$, R3.0, R3.1, and R3.2" for $(Q_1, \Gamma_3, rc_2, pc_3)$ follow directly from R3.0, R3.1, and R3.2" for $(Q_1, \Gamma_1, rc_1, pc_1)$.

Note 1

\[ pc_3 \cdot s > 0 \]
\[ \Rightarrow \{\text{def. } pc_3\} \]
\[ (pc_3 \cdot s = pc_1 \cdot s) \land (\text{TV} \cdot \Gamma_3 \cdot s \cap \{L, H\} \neq \emptyset) \]
\[ \Rightarrow \{\text{P1 and def. } pc\} \]
\[ ((pc_3 \cdot s = pc_1 \cdot s) \lor (pc_3 \cdot s = pc_2 \cdot s)) \land (\text{TV} \cdot \Gamma_3 \cdot s \cap \{L, H\} \neq \emptyset) \]

Note 2

\[ (pc_3 \cdot s > 0) \land (pc_1 \cdot s > 0) \]
\[ \Rightarrow \{\text{note 1 and R3.1 for } ((Q_1, \Gamma_3, rc_2, pc_3))\} \]
\[ (\text{TV} \cdot \Gamma_3 \cdot s \cap \{L, H\} \neq \emptyset) \land (\text{TV} \cdot \Gamma_1 \cdot s \in \{\{L\}, \{H\}\}) \]
\[ \Rightarrow \{\text{from P0 and P1: destore-}\Gamma \subseteq \text{destore-}\Gamma_1, \text{ and calculus}\} \]
\[ \text{TV} \cdot \Gamma_3 \cdot s = \text{TV} \cdot \Gamma_1 \cdot s \]
note 3  
\[(pc_2 \cdot s > 0) \land (pc_3 \cdot s = pc_1 \cdot s)\]  
\[= (R_{3.0} \text{ for } ((Q_3, \Gamma_1), r_{C_1}, pc_1))\]  
\[= (pc_3 \cdot s > 0) \land (pc_3 \cdot s = pc_1 \cdot s) \land (A, x, y : pn \cdot s = [x, y] : \Gamma_1 \cdot x \neq \Gamma_1 \cdot y)\]  
\[= \text{\{from R.3.1 for } ((Q_3, \Gamma_1), r_{C_1}, pc_1) : TV \cdot \Gamma_1 \cdot s \in \{[L], [H]\}\} \]  
\[pc_3 \cdot s > 0 \land pc_3 \cdot s = pc_1 \cdot s \land (B, x, y : pn \cdot s = [x, y] : \Gamma_1 \cdot x \in \{[L], [H]\} \land \text{destore} \cdot \Gamma_1 \cdot y = \emptyset\}\]  
\[\Rightarrow \text{\{note 1 (using the assumption) and } (P_0, P_I) \text{ destore} \cdot \Gamma_3 \subseteq \text{destore} \cdot \Gamma_1\}\]  
\[= (E, x, y : pn \cdot s = [x, y] : (\Gamma_3 \cdot x \in \{[L], [H]\}) \land (\text{destore} \cdot \Gamma_3 \cdot y = \emptyset)\}\]  
\[\Rightarrow \text{\{calculation\}}\]  
\[= (A, x, y : pn \cdot s = [x, y] : \Gamma_3 \cdot x \neq \Gamma_3 \cdot y)\]  
\[\square \text{(end proof 5.17c)}\]

As argued above, the next-state function for restricted delays can be defined as follows.

5.18 Definition

For \(rd \in DC\) and \(pd \in DC\), denoting the reaction-delay and the pass-delay,  
\[\text{next}_{\text{st}, \text{pd}} \in \text{NST} \rightarrow (\text{ST}1 \times \text{DC} \times \text{DC}) \rightarrow K(\text{ST}1 \times \text{DC} \times \text{DC})\]  
\[\text{is defined by:}\]  
\[\text{next}_{\text{st}, \text{pd}} : \gamma_{||(Q_0, \Gamma_0), r_{C_0}, pc_0, ((Q_3, \Gamma_1), r_{C_1}, pc_1)}\]  
\[= K_{\text{st}, \text{pd}} : \gamma_{||(Q_0, \Gamma_0), r_{C_0}, pc_0, ((Q_3, \Gamma_1), r_{C_1}, pc_1)} \land\]  
\[\text{destore} \cdot \Gamma_1 = \min \{\Gamma, r_{C_0}, pc_0 : K_{\text{st}, \text{pd}} : \gamma_{||(Q_0, \Gamma_0), r_{C_0}, pc_0, ((Q_3, \Gamma_1), r_{C_1}, pc_1)} : \text{destore} \cdot \Gamma_1\}\]  

5.19 Remark on restricted pass-delays

In remark 3.15 the possibility to model distinct reaction-delay restrictions for upgoing and for downgoing transitions is discussed (recall: functions \(rd_R\) and \(rd_L\)). Similarly, the restrictions on pass-delays can depend on the cause of the pass-delays, or, more generally, on the previous switch-state and the current node-state. This can be modelled by choosing the pass-delay \(pd \in SST \rightarrow \text{NSTC} \rightarrow DC\). Restriction R3.2" must then be replaced (in def. 5.16 \((K_{\text{st}, \text{pd}})\)) by:

\[R_{3.2'} : (A, s : pc_1 \cdot s > 0 : (pc_0 \cdot s > 1) \land (pc_1 \cdot s = pc_0 \cdot s - 1))\]  
\[\lor ((pc_0 \cdot s = 0) \land (Q_0 \cdot s \land (TV \cdot \Gamma_1 \cdot s \notin TV \cdot \Gamma_1 \cdot s)) \land (pc_1 \cdot s = pd \cdot Q_0 \cdot \Gamma_1 \cdot s))\]  

For instance, for a switch \(s\) with \(pn \cdot s = [x, y]\), it seems reasonable to define the pass-delay \(pd \in \text{NSTC} \rightarrow DC\) in such a way that the sequence \(pd \cdot \Gamma_1 \cdot s, pd \cdot \Gamma_2 \cdot s, pd \cdot \Gamma_3 \cdot s\) is ascending if \((\Gamma_1 \cdot x, \Gamma_1 \cdot y) = ([H], [H]) \land (\Gamma_2 \cdot x, \Gamma_2 \cdot y) = ([H], [I]) \land (\Gamma_3 \cdot x, \Gamma_1 \cdot y) = ([H], [I])\) since the charge difference between \(x\) and \(y\) is ascending in the sequence \(\Gamma_1, \Gamma_2, \Gamma_3\).
In section 5.0.0 (Motivation of assumptions (0a) and (0b)) we discussed the possibility of choosing a zero pass-delay in some cases, in particular in those cases where one of the pass nodes of a conducting switch has a source value and the other has a stored charge of the same type (as in \( \Gamma_1 \) above). Using \( \text{pd} \in \text{NSTC} \rightarrow \text{DC} \) as described above, this can be modelled by requiring, for \( \text{SW-s} \land (\text{pn-s} = [x,y]) \), that \( \text{pd} \) satisfies \( (\text{store-} \Gamma x = \text{store-} \Gamma y) \lor (\text{pd-} \Gamma x = 0) \).

If \( \text{pd} \in \text{NSTC} \rightarrow \text{DC} \) satisfies \( (A \Gamma x, x, y : (\text{pn-s} = [x,y]) \land (\text{Γ-x} \neq \Gamma y) : \text{pd-Γ-x} > 0) \), lemma 5.17c holds and the next-state function can be defined similar as in def. 4.18.

If, however, \( (E \Gamma x, s, x, y : (\text{pn-s} = [x,y]) \land (\text{Γ-x} \neq \Gamma y) : \text{pd-Γ-x} = 0) \), lemma 5.17c need not hold, which is demonstrated in the example below. In that case, destore-\( \Gamma_1 \) must be a minimal element of the set mentioned in def. 5.18 (next\( \Gamma_{n\text{d},\text{pd}} \)) instead of the least element.

**Example**

Consider the circuit depicted in figure 5.3a, with starting state \( \Pi_0 \) (fig. 5.3b) and new source connection \( \{(x, \{H\}), (y_0, \{H\}), (y_1, \{\ell\}), (y_2, \{\ell\}), (y_3, \{\ell\}), (y_4, \{H\})\} \). Assume that \( \text{rd} = 0 \) and that \( \text{pd} \) satisfies
\[
(A \Gamma x, x, y : (\text{pn-s} = [x,y]) \land (\text{store-} \Gamma x = \text{store-} \Gamma y) : \text{pd-Γ-x} = 0)
\]
\[
\land (A \Gamma x, x, y : (\text{pn-s} = [x,y]) \land (\text{store-} \Gamma x \neq \text{store-} \Gamma y) : \text{pd-Γ-x} > 0)
\]

The successor of \( \Pi_0 \) is \( \Pi_1 \). The possible successors (in \( k_{\text{nd},\text{pd}} \)) of \( \Pi_1 \) are \( \Pi_2, \Pi_3, \) and \( \Pi_4 \). Notice that \( \Pi_2, \Pi_3, \) and \( \Pi_4 \) have the same switch-state and that no least node-state exists. Notice that, without this assumption and with positive pass-delays, the successor of \( \Pi_1 \) is \( \{h\} \) in \( y_1, y_2, \) and \( y_3 \).

\[\text{figure 5.3a}\]

\[\text{figure 5.3b}\]

\( \Pi \) (and remark 5.19)

Similar as in section 3.2, the consequences of restrictions on pass-delays for the notion well-matchedness are investigated. It will turn out that this notion does not change if the pass-delays are restricted.

**On the consequences of restrictions on pass-delays for well matchedness**

The modelling of restricted delays for the notions regarding initial behaviour is similar to the one presented in section 3.2.

In the following definition, \( D \in \text{DC} \times \text{DC} \) denotes the set of pairs reaction-delay, pass-delay.
5.20 Definition
For \( D \subseteq DC \times DC \) the following notions have the same type as their counterparts without the subscript \( p_0 \), and are defined by:

a) \( stable_{3D} \gamma \Pi\)
\[
= (E \forall p_d, p_c : D \land (p_d, p_c) \land DC - p_c : next_{3_d,p_0} \gamma (\Pi, rc, pc) \land (\Pi, rc, pc))
\]
b) \( feasible_{3D} \gamma \Pi\)
\[
= (E \forall p_d, p_c : D \land (p_d, p_c) \land DC - p_c : (next_{3_d,p_0} \gamma^+(\Pi, rc, pc) \land (\Pi, rc, pc))
\]
c) \( FSL_{3D} \gamma \Psi\)
\[
= (E \forall p_d, r, pl : D \land (p_d, p_c) \land \Psi^{\*}(DC - r) \land \Psi^{\*}(DC) \land p_l \land (\Psi - r = \Psi - pl) \land (\Psi - pl = \Psi - p_0)
\]
\[= (A \forall i : 0 \leq i \leq \Psi - \Psi \land next_{3_d,p_0} \gamma (\Psi_{i,r,l,p_0} (\Psi_{i,r,l,p_0} (\Psi_{i,r,l,p_0})))
\]
where \( \Psi_{i,r,l,p_0} \) for \( i = \Psi - \Psi \) is defined as \( \Psi_{i,r,l,p_0} \).
d) \( WM_{3D,C} \gamma = (FSL_{3D} \gamma \subseteq (FCA0 \land cgdlL))\)

Due to delay restrictions the behaviour of circuits is restricted (lemma 5.21a,c,d). Since the delays are inactive in stable states (cf. lemma 5.7a), stable states do not change (lemma 5.21b).

The main result of the following lemmas is that restrictions on delays do not influence the notion of well-matchedness (lemma 5.21e).

5.21 Lemma

a) \( (A \gamma \Pi, rd, pc) : NST \gamma \land ST1 \Pi \land (rd, rd, pc) \subseteq DC\)
\[= next_{3_d,p_0} \gamma (\Pi, rc, pc) \subseteq next_{3} \gamma (\Pi, rc, pc)\]
b) \( (A D, \gamma) : (D \subseteq DC \times DC) \land (D \neq \emptyset) \land NST \gamma \land stable_{3D} \gamma = stable_{3} \gamma\)
c) \( (A D, \gamma) : (D \subseteq DC \times DC) \land (D \neq \emptyset) \land NST \gamma \land feasible_{3D} \gamma \subseteq feasible_{3} \gamma\)
d) \( (A D, \gamma) : (D \subseteq DC \times DC) \land (D \neq \emptyset) \land NST \gamma \land FSL_{3D} \gamma \subseteq FSL_{3} \gamma\)
e) \( (A D, C, \gamma) : (D \subseteq DC \times DC) \land (D \neq \emptyset) \land CIR \land NST \gamma \land WM_{3D,C} \gamma = WM_{3} \gamma\)

Lemma 5.21a follows directly from definitions 5.2 (next3) and 5.18 (next3_{d,p_0}). Lemmas 5.21c,d follow directly from lemma 5.21a using the definitions. Lemmas 5.21b,c are proven below.

Proof of 5.21b
Let \( D, \gamma, \Pi \) satisfy \( (D \subseteq DC \times DC) \land (D \neq \emptyset) \land NST \gamma \land ST1 \Pi \). Then:

\( stable_{3D} \gamma \Pi\)
\[= \{ \text{property: proof similar to the proof of lemma 5.7a} \}
\[= (E \forall p_d, p_c : (p_d, p_c) \in D \land next_{3_d,p_0} \gamma (\Pi, 0, 0) \land (\Pi, 0, 0))
\]
\[= \{ \text{definitions 5.2 (next3) and 5.18 (next3_{d,p_0})} \}
\[= \{ \text{lemma 5.7a} \}
\[= \{ \text{lemma 5.7a} \}
\]

\( stable_{3} \gamma \Pi\)
Proof of 5.21c

Let \( D, C, \gamma \) satisfy \((D \subseteq DC \times DC) \land (D \neq \emptyset) \land CIR \land \text{NST} \land \gamma\). Then:

\[ WM3_{D-C} \gamma \]

\( \vdash \) \{property: proof similar to note 0 in proof of th. 4.26\}

\[ \text{stable}_{D} r_{\gamma} \subseteq \text{csgd} \]

\( = \) \{lemma 5.21b\}

\[ \text{stable}_{D} r_{\gamma} \subseteq \text{csgd} \]

\( = \) \{th. 5.10\}

\[ WM3_{C} \gamma \]

From lemma 5.21d and definitions 5.5 \((WM3)\) and 5.20d \((WM3_{D})\) follows:

\[ WM3_{C} \gamma \Rightarrow WM3_{D-C} \gamma \], which completes the proof.

\( \Box \) (end proofs 5.21)

5.1.1 Well-Functioning

In this section, well-functioning is defined in a similar way as in section 4.2. The correctness criteria used in section 4.2 (i–iv) are used again, with an obvious extension of correctness criterion iv. The section concludes with an investigation of the relation between WF2 and WF3.

Lists of successive states starting in \((\Pi, r, c, pc)\) can be defined similar as in ch. 4 (cf. def. 4.32).

5.22 Definition \{resulting state lists 3\}

For \( r, pd \in DC \), \( RSL_{r, pd}^3 \in \text{NST} \land (ST1 \times DC \times DC) \land \omega = (ST1) \land \Omega \) is defined by:

\[ RSL_{r, pd}^3_{\Omega, (\Pi, r, c, pc), \Psi} \]

\( = \) \( \{ r_{pd} : \omega(\Delta C \times r1 \times \Delta C) \land \omega(\Delta r1 \times pd) \land (\psi_{pd} r_{pd} \psi_{pd} = (\Pi, r, c, pc)) \}

\( \cup \{ a : 0 \leq a : \text{next}_{r, pd} r_{pd} \} \}

A circuit is called well-functioning if the correctness criteria formulated in section 4.2 (i–iv) are satisfied. Correctness criterion iv, i.e. 'all state transitions between successive states are correct', needs to be extended. Namely, besides \( cst0 \) for infinite state lists we also need \( cst1 \) for infinite state list. The extended criterion is defined in definition 5.13b \((cstL)\).

We have now argued the following definition for well-functioning 3, where \( D \) denotes the set of pairs reaction-delay, pass-delay.
5.23 Definition (well-functioning 3)
For \( D \subseteq DC \times DC \), \( WF3D \in \text{CIR} \rightarrow \text{NST} \rightarrow (ST1 \times DC \times DC) \rightarrow B \) is defined by:
\[
WF3D \cdot C \cdot \gamma (T, r, p, c) = (A, rd, pd : (rd, pd) \in D : RSL3_{nd, pd} \cdot \gamma (T, r, p, c) \subseteq (c, g, d) \cap CFL \cap FCAI \cap casL) \]

In example 5.24a the inverter from example 4.36a is reconsidered. The example concludes that the restrictions on the delays, in order for the inverter to function correctly, are equal to the restrictions found in example 4.36a. Example 5.24b, which reconsiders the circuit from ex. 4.36c, demonstrates that the conclusion from ex. 5.24a does not mean that the pass-delays in an inverter, or, in general, the pass-delays caused by a change of switch-state (from nonconducting to conducting), can be ignored in our model.

5.24 Example
a Reconsider the circuit from example 4.36a (depicted in figure 4.2a), starting in the topmost state of figure 5.4a. Let the source-connection change to \( \{(x, \{H\}), (x, \{H\})\}. \) The resulting state lists are depicted in figure 5.4a. The difference with the resulting state lists in the previous model, which are depicted in fig. 4.2b (ex. 4.36a), is a result of the possible pass-delay in switch \( s_0 \). In order for the circuit to be well-functioning, the leftmost path in fig. 5.4a – with a conflict in node \( z \) – must be avoided, and, hence, the delays must be restricted by \( rd_{h, s_0} \geq rd_{h, s_1} \). Notice that this is the same restriction as in ex. 4.36a and that the pass-delays need not be restricted.

b Reconsider the circuit from example 4.36c (fig. 4.2c), with starting state \( \Pi \) (fig. 5.4b) and new source-connection \( \delta \) (ex. 4.36c). Assume that the delays satisfy \( \text{rd}_{s_0} = \text{rd}_{s_1} \land (A \land 0 \delta < 3 : (p_{d, s_1} > 0)) \). The resulting state list is depicted in fig. 5.4b. The transitions are drawn at the left side of fig. 5.4b. Notice that, since transitions \( t_0 \) and \( t_1 \) are not correct-0, the circuit is not well-functioning. Ignoring the pass-delays of \( s_0 \) and \( s_1 \), or assuming that \( (p_{d, s_0} = 0) \land (p_{d, s_1} = 0) \), leads to the state transitions depicted at the right side of fig. 5.4b, and, consequently, to the (incorrect) conclusion that the circuit is well-functioning.
The following lemma relates the notions from this section to the notions from section 4.2.

5.25 Lemma
Let \( C, \gamma, \Pi, \rho_c \) satisfy \( CIR \land NST \land ST1 \land DC \land \rho_c \).

a For all \( DC \land (\rho_c = 0) \):
\( a0 \) \( \text{next}_2^{\Delta, \rho_c} \gamma(\Pi, \rho_c, 0) = \text{next}_2 \gamma(\Pi, \rho_c) \)
\( a1 \) \( RSL_2^{\Delta, \rho_c} \gamma(\Pi, \rho_c, 0) = RSL_2 \gamma(\Pi, \rho_c) \)

b Let \( D \subseteq DC \times DC \) and let \( RD = \{ (\rho_d, \rho_c) : (\rho_d, \rho_c) \in D \land \rho_c \} \). Then:
\( b0 \) \( (D = RD\{0\}) \Rightarrow (WF_2^{\Delta, \rho_c} \gamma(\Pi, \rho_c, 0) = WF_2^{\Delta, \rho_c} \gamma(\Pi, \rho_c)) \)
\( b1 \) \( (A \rho_d : RD \land (\rho_d, 0) \in D) \Rightarrow (WF_2^{\Delta, \rho_c} \gamma(\Pi, \rho_c, 0) = WF_2^{\Delta, \rho_c} \gamma(\Pi, \rho_c)) \)

Lemma 5.25a follows from definitions 4.31 (\( \text{next}_2^{\Delta, \rho_c} \)) and 5.18 (\( \text{next}_2^{\Delta, \rho_c} \)), compare with the proof of lemma 5.6b. Lemma 5.25a1 follows directly from 5.25a using definitions 4.32 (\( RSL_2^{\Delta, \rho_c} \)) and 5.22 (\( RSL_2^{\Delta, \rho_c} \)). Lemma 5.25b follows directly from 5.25a1 using definitions 4.34 (\( WF_2^{\Delta, \rho_c} \)) and 5.23 (\( WF_3^{\Delta, \rho_c} \)).

The right-most implication sign in 5.25b cannot be replaced by an equality sign. Namely, consider the circuit depicted in figure 5.5a below (ex. 5.14), starting in state \( \Pi_1 \) (fig. 5.5b) with new source-connection \( \gamma_0 \). The circuit is well-functioning if the delays satisfy (cf. ex. 5.14) \( pd \cdot x_0 + rd \cdot x_2 \geq pd \cdot x_1 + rd \cdot x_3 \). With \( D = \{0\} \times DC \), and, consequently, \( RD = \{0\} \), the following holds:
\( (A \rho_d : RD \land (\rho_d, 0) \in D) \land \neg WF_3^{\Delta, \rho_c} \gamma_0 (\Pi_1, \rho_c, 0) \land WF_2^{\Delta, \rho_c} \gamma_0 (\Pi_1, \rho_c) \).

\[ \begin{array}{cc}
      s_0 & x_0 \\
      H & y_0 \\
      \gamma_0 & y_1 \\
      \Pi_1 & \gamma \ \\
      \gamma_0 & \Pi_1 \\
      x & y_1 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      x & x \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      x & x \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      x & x \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      x & x \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      x & x \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      x & x \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      x & x \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      x & x \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      x & x \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      x & x \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      x & x \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      x & x \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      x & x \\
      \Pi_1 & \Pi_1 \\
      \gamma_0 & \gamma_0 \\
      \Pi_1 & \Pi_1 \\
      s_0 & s_2 \\
\end{array} \)

The following lemma is the counterpart of lemma 4.39 and is given for later use (in ch. 7). Note that lemma 5.26a, which shows that a correct state has a unique next state, only holds for \( \rho_c \in DC \) and not for \( \rho_c \in NST \rightarrow DC \) (cf. remark 5.19). Lemma 5.26b shows that correct resulting state lists have a suffix consisting of a repeated feasible state list. Example 4.40 is still capable of showing that well-functioning circuits do not necessarily end up in a stable state.
5.26 Lemma

a) Let ST1·Π. For all (rd, pd) ∈ D×DC :

cgd1·Π ⊢ (A rc, pc : DC·rc ∧ DC·pc : #·(next3 nd, pd)·τ(Π rc, pc)) = 1 )

b) Let ST1·Π ∧ DC·rc ∧ DC·pc ∧ D(Π ST1)·Ψ. For all (rd, pd) ∈ D×DC :

RSL3 nd, pd · τ(Π rc, pc)·Ψ ∧ cgdL·Ψ

⇒ (E Φ0·Φ1 : Lκ(Π ST1)·Φ0 ∧ FSL3 nd, pd · τ·Φ1 : Ψ = cat·Φ0·σ(Φ1))

The proofs of 5.26a and 5.26b are fairly straightforward (use lemma 5.17a).

5.1.2 Can pass-delays be modelled as part of reaction-delays?

The question above is raised for the following two reasons.

A It would be a simplification of our model:

Positive pass-delays complicate the calculation of state transitions (due to the mutual
dependence of the succeeding node-state and the succeeding pass-delay counter, cf. sections
5.0.0 and 5.1.0) For zero pass-delays the model presented in this chapter equals the model
presented in chapter 4 (cf. lemma 5.25).

B It seems possible:

Intuitively, a signal passing switches influences the behaviour of the circuit when it reaches the
gate of a switch. Following this intuition, it seems that the behaviour of a circuit can be
simulated correctly by adding the pass-delays of switches leading to the gate of a switch s to the
reaction-delay of that switch s.

For instance, consider the circuit depicted in fig. 5.5a above starting in state Π1 (fig. 5.5b) with
new source-connection γ0. The circuit is well-functioning if (cf. ex. 5.14) pd·s0 + rd1·τ2 ≥
pd·s1 + rd1·τ3. The suggestion above is to use, instead of (rd, pd) ∈ D, the delays (rd1, 0)
with rd1·s1 = pd·s1 + rd·s2 (for 0i<2). Now the model from chapter 4 can be used (cf.
lemma 5.25b0). According to that model, the circuit is well-functioning if (cf. ex. 4.36a)
rd1·s1 ≥ rd1·s1 , which equals the result above.

Furthermore, note the correspondence between the criteria cat0 and cat1 in those cases. For
instance, in our example parts of cat1 and cat0 are (pd0·s1 > 0) ⇒ (TV·Γ0·s1 = TV·Γ1·s1) and
(rd0·s2 > 0) ⇒ (Γ0·(g·s2) = Γ1·(g·s2)) , for 0i<2. These requirements equal cat0 for delays
(rd1, 0), which is (rd1·s1 > 0) ⇒ (Γ0·(g·s2) = Γ1·(g·s2)).

Example 5.27 below demonstrates that this transformation does not lead to a correct simulation
of the behaviour of a circuit in all cases. It also shows that other ways to model pass-delays as
part of reaction-delays are, in general, not correct.
5.27 Example

Consider the circuit depicted in fig. 5.6a below. Nodes x and y are input nodes and node z₂ is the output node. For simplicity, we assume that, for all i, j with i ∈ {0, 2, 4} ∧ i ≤ j < 6:

$$(\text{rd}_{-3} = \text{rd}_{-5}) ∧ (\text{pd}_{-3} = 0) ∧ (\text{pd}_{-5} > 0) ∧ (\text{pd}_{-6} > 0).$$

Let the circuit start in state $\Pi₀$ (fig. 5.6b); compare with ex. 4.38b. Input y is changed to $\{H\}$ in order to load the value of x. The feedback path around the two inverters will be interrupted since $\text{s}_1$ will become nonconducting. The state transitions are $\Pi₀ → \Pi₁ → \Pi₂ → \Pi₃ → \Pi₄ → \Pi₅$ (compare with ex. 4.38c). Once the circuit has reached $\Pi₅$, the feedback path can be established and the connection to x interrupted. This is done by changing input y to $\{L\}$. The resulting state transitions are $\Pi₅ → \Pi₆ → \Pi₇$.

The protocol described above is based on the fundamental mode assumption (sect. 0.0): the environment of the circuit will not send the next input change until the circuit has reached a stable state. If the input-output mode assumption (sect. 0.0) is used, which is: ‘the environment will not send the next input change until the expected output change is established’, the protocol can differ due to pass-delays in $\text{s}_0$ and $\text{s}_6$. Namely, the input change in y from $\{H\}$ to $\{L\}$ may arrive when the circuit is in state $\Pi₄$ (instead of in $\Pi₅$ as assumed above). A possible resulting state list then is $\Pi₄ → \Pi₅ → \Pi₉ → \ldots$. Since $\Pi₃ ∈ \text{cgdl}$, the circuit is not well-functioning.

The suggested transformation of pass-delays into reaction-delays (in B above) means adding the pass-delays of $\text{s}_0$ or $\text{s}_6$ to the reaction-delays of $\text{s}_2$ and $\text{s}_3$. In that case the state transitions are $\Pi₀ → \Pi₁ → \Pi₂ → \Pi₃ → \Pi₅$ and $\Pi₃ → \Pi₆ → \Pi₇$ respectively. State $\Pi₄$ — with the (potentially destructive) stored value in $\text{z}_3$ — is not observed as an intermediate state. Note that the circuit would, using the transformation, incorrectly be found to be well-functioning.

Furthermore, note that adding the pass-delay of $\text{s}_6$ to the reaction-delay of either $\text{s}_1$ or $\text{s}_4$ and using a zero pass-delay in $\text{s}_6$ would also ignore $\Pi₉$ as a possible intermediate state, and, hence, the circuit would be classified as well-functioning.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{5.6a}
\caption{figure 5.6a}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{5.6b}
\caption{figure 5.6b}
\end{figure}
In conclusion, the possible charge conflict in $z_0$ can only be observed if a positive pass-delay is used in $s_0$. The possible charge conflict can only be avoided by restricting the pass-delay of $s_0$ relative to the delay of the environment (to change $y$ after $z_2$ has changed).

(End of example 5.27)

Concluding Remark of section 5.1.2

Although not correct in general (cf. ex. 5.27), the transformation described above seems to have the potential to avoid calculation with positive pass-delays in many cases (cf. the running ex. of this section; fig. 5.5a). Investigating in what cases this transformation is correct therefore appears to be an interesting and promising subject for further research (also in view of the correspondence between cte0 and cte1). In this thesis no results in this direction are given.

5.2 Wire-delays

The concept of wire-delay models the time it takes a voltage transition to pass from one end of a wire to the other end of that wire. The simplest way of modelling these delays is to use the correspondence between wire-delays and pass-delays. Although the modelling of wire-delays is not developed in detail, the argumentation below is sufficient to see that the results that can be obtained are similar to the results obtained in the previous sections.

The following example illustrates the notion of wire-delay.

5.28 Example

Consider the C-mos inverter depicted in figure 5.7a, which is studied in example 5.24a. Now assume that the wire-delays between $x$ and the gates of $s_2$ and $s_3$ are not negligible. This situation is depicted in figure 5.7b, where $w0$ and $w1$ are wires. Let the starting state be $\Pi 1$ (fig. 5.7c) and let the new source-connection be $y0$ (fig. 5.7c). The resulting state lists, with arbitrary wire-delay and pass-delay, are depicted in figure 5.1b (ex. 5.14).

Similar as in example 5.14, we can conclude that the circuit is well-functioning if the delays satisfy: $wd-w0+rd-s2 \geq wd-w1+rd-s3$, where $wd$ denotes the wire-delay.
In section 1.0 ('Representation of circuits') we used part of assumption (b) from section 0.3, viz. 'wire-delays are negligible', to define circuits (def. 1.0 (CIR)). In that definition, wires are represented by one node. In order to model non-zero wire-delays we need to include wire-elements explicitly in our circuits. Such circuits, defined below, are called extended circuits. The circuit depicted in figure 5.7b, for instance, is an extended circuit.

5.29 Definition [extended circuits]
An extended circuit is a sextuple \( \langle N, SW, W, t, g, pn \rangle \), where
\[ pn \in (SW \cup W) \rightarrow B2N \] , with \( B2N \) as defined in def. 1.0,
\( W \) is a finite set, called the set of wires, that satisfies \( W \cap (N \cup SW) = \emptyset \), and
\[ \langle N, SW, t, g, pn \downarrow SW \rangle \in CIR \],
with \( pn \downarrow SW \in SW \rightarrow B2N \) defined by \( (pn \downarrow SW) \cdot s = pn \cdot s \).
The set of extended circuits is called \( XCIR \).

Wire-delays are comparable to pass-delays in permanently conducting switches in all aspects; compare, for instance, example 5.28 with example 5.14. Using similar assumptions for wire-delays as used for pass-delays in section 5.0.0 ((0a,b), (1), and (2)), the restrictions on a wire-delay counter are similar to those defined for the pass delay counter (section 5.0.0: R3.i or r3.i). A correctness criterion similar to \( csi \) is required for wire-delays.
Consequently, the simplest way to model wire-delays, that is, with as little extension of the model as possible, is by treating wires as a special kind of (permanently conducting) switches. This way of modelling wire delays is described below. It is not developed in detail, since it is easily seen that the results that can be obtained are similar to those obtained in the previous sections.

In order to regard wires in a similar way as permanently conducting switches, the set of states is extended as follows:
\[ XSST = (SW \cup W) \rightarrow \emptyset \]
\[ ST2 = \{ Q, \Gamma : XSST-Q \land NSTC-\Gamma \land (A \ w : W \wedge Q \wedge w) : (Q, \Gamma) \} \]
Of course, the domains of a number of notions must be extended (from CIR to XCIR, SST to XSST, and from ST1 to ST2). The state transitions in this model must satisfy the restrictions formulated in section 5.0.0 with the following extensions:
add R0.4: \( (A \ w : W \wedge Q \wedge w) \)
extend the domain \( SW \) to \( SW \cup W \) in restrictions R3.i (or, similarly, in r3.i).
This model can now be developed in the same way as the model in sections 5.0 and 5.1.
To illustrate this, notice that the consistency notion for wires, say \( \text{consistent}_2 \in \text{NSTC} - W \rightarrow \mathbb{B} \), and the correct state transition requirement for wire-delays, say \( \text{cs}_2 \in \text{ST}2 - \text{ST}2 \rightarrow \mathbb{B} \), are, according to the model described above, defined as follows.

Let \( \text{ST}2(\mathbb{Q},\Gamma) \land \text{ST}2(\mathbb{Q}_1,\Gamma_1) \land W_w \).

\[
\text{consistent}_2 \cdot \Gamma \cdot w \\
= \{ \text{using the extension described above} \}
\]

\[
\text{consistent}_1 \cdot \Gamma \cdot Q \cdot w
\]

\[
= \{ \text{using } \text{ST}2(\mathbb{Q},\Gamma) \Rightarrow Q \cdot w \text{ and the extension of def. 4.16 (consistent)} \}
\]

\[
(A, x, y : p : w = [x, y] : \Gamma \cdot x = \Gamma \cdot y)
\]

and

\[
\text{cs}_2(\mathbb{Q},\Gamma, (Q_1,\Gamma_1))
\]

\[
= \{ \text{using the extension described above} \}
\]

\[
\text{cs}_1(\mathbb{Q},\Gamma, (Q_1,\Gamma_1))
\]

\[
= \{ \text{using } \text{ST}2(\mathbb{Q},\Gamma) \Rightarrow Q \cdot w \text{, the result above, and the extension of def. 5.13a (cs)} \}
\]

\[
(A, w : \neg \text{consistent}_2 \cdot \Gamma \cdot w : TV \cdot \Gamma \cdot w = TV \cdot \Gamma_1 \cdot w)
\]

It can easily be verified that the results obtained in the previous sections also hold for the extended model described above.

Consequently, the well-matchedness notion does not change. Since all resulting states are stable in case of well-matchedness, and all wires are consistent-1 in stable states (compare with th. 5.10 and prop. 4.17c), the extended circuits (def. 5.29) are not needed for initial behaviour.

The results w.r.t. well-functioning are similar to those given in section 5.1. For instance, by replacing switch \( s_0 \) in example 5.27 (sect. 5.1.2) by a wire-element \( w \), we can conclude (similar to section 5.1.2) that wire-delays cannot be modelled as a part of reaction-delays or as a part of pass-delays of switches.
5.3 Concluding remarks on chapter 5

In this chapter the model is extended in such a way that arbitrary — but non-negative and finite —
pass-delays and wire-delays can be captured and restrictions on them can be expressed. The
consequences for the correctness criteria for initial and dynamic behaviour are investigated.

Assumptions on pass-delays and wire-delays

The formalisation given of pass-delays is based on the following assumptions (cf. sect. 5.0.0):

(0a) A pass-delay can be active in a switch only if the switch is conducting.
(0b) A pass-delay cannot be active in a switch if the pass nodes have the same value.
(1) A pass-delay can be active in a switch only if the value that needs to be transported by
that switch is \{L\} or \{H\}. This means that we assume — in our model — that the
pass-delay is zero if the value that needs to be transported is \{L,H\} or a stored charge.
(2) A pass-delay must have an initial cause. This cause can either be that the switch
becomes conducting, or that the value that needs to be transported has changed. For the
latter case, we also assume that a pass-delay is not possible if the transported value
was \{L,H\} in the preceding state.

In appendix C we proved that assumption (1) does not influence the results regarding initial
behaviour, but can be pessimistic regarding dynamic behaviour.

The assumptions for wire-delays are similar (read 'wire' instead of 'switch'), with the obvious
omission of assumption (0a) and the first cause mentioned in assumption (2).

Formalisation of pass-delays and wire-delays

Pass-delays and wire-delays can be modelled in a similar way (cf. sect. 5.2). The modelling of
these types of delay turned out to be more difficult that the modelling of reaction-delays (cf.
sect. 5.0.0: the mutual dependence of \(\Gamma_i\) and \(p_c\)). As a result of this, the calculation of
the next-state function is complicated if pass-delays and wire-delays are positive. Calculating with
positive pass-delays and wire-delays cannot be avoided in all cases (cf. sect. 5.1.2. and 5.2).
It seems, however, that pass-delays and wire-delays can, in many cases, be modelled as part of
reaction-delays (cf. sect. 5.1.2). In this thesis, no results in that direction are given.

For zero pass-delays and wire-delays the next state function equals the one defined in chapter 4
(cf. lemmas 5.6a and 5.25w).

In section 5.1.0 we have shown how restrictions on these types of delay can be expressed.
Similar as for reaction-delays, these restrictions can depend on the cause of the delay (cf.
remark 5.19).
Initial behaviour
For initial behaviour it is possible to calculate with zero delays. The correctness criteria for initial behaviour are equal to those formulated in the previous chapter (cf. cor. 5.11 and sect. 5.0.3). For initial behaviour, the resulting states in this model equal the resulting states in the model from chapter 4 (cf. lemma 5.7 and th. 5.12).

Dynamic behaviour
The modelling of dynamic behaviour is done in a similar way as in section 4.2. The correctness criterion for dynamic behaviour is expressed in WF3 (def. 5.23). The main extension of this criterion (w.r.t. WF2) is the requirement for stability of the cause for pass-delays and wire-delays when they are active, expressed in csi (def. 5.13a).
Similar as for the next-state function, the calculation of this correctness criterion is complicated for non-zero pass-delays or wire-delays (cf. lemma 5.25).
CHAPTER 6  IMPERFECTNESS OF SWITCHES

The model presented in the previous chapters does not take the 'imperfectness', or 'poorness', of single-transistor switches into account. An abstraction to perfect switches is made (cf. sect. 0.3, restriction (a)). Consequently, this model will classify some circuits as correct whereas the imperfectness of switches causes malfunctioning of these circuits. In this chapter the imperfectness of single-transistor switches is explained and an extension of the model is given which takes this imperfectness into account. The central notions of this extension are called mutilating power and mutilation degree. They are informally introduced in section 6.0 and defined in section 6.1. The mutilation degree of signals is a new notion that enables a concise and elegant description of the effects of the imperfectness of switches, and, consequently, enables a verification whether this imperfectness causes malfunctioning. Besides the familiar correctness criteria from the previous chapters, new correctness criteria are defined using the notions above. The model provides a nice separation of concerns, since the familiar criteria can be calculated using the model from chapter 5, and the new criteria can — but need not — be calculated separately.

The chapter is organised as follows. In the first section the imperfectness of switches — due to threshold voltages — is analysed and an introduction to the central notions mentioned above is given. In the second section these central notions are further analysed and defined. The section is divided into two subsections; the first giving the formalisation for initial behaviour of circuits (the simpler case), the second one for dynamic behaviour. The correctness criteria due to imperfectness of switches are defined in the third section. In the fourth section the formalisation given in the previous sections is discussed, and some generalisations of it are given. As usual, the final section summarizes and discusses the results.
6.0 Introduction to Mutilation

Perfect switches, as used in the previous chapters (cf. sect. 0.3, restriction (a)), are abstractions of transistors. To be able to describe the imperfectness of transistors we consider a more detailed description of their physical behaviour.

Let $V_L$ and $V_H$ be the voltage values of the L-source and the H-source respectively. They are assumed to be constants satisfying $V_L < V_H$. The voltage value at a node $z$ is denoted by $V^z$. Consider an n-mos transistor $nt$ with gate node $gn$ and pass nodes $x$ and $y$, and a p-mos transistor $pt$ with gate node $gp$ and pass nodes $u$ and $v$ (depicted in figure 6.0). They satisfy:

\[
\begin{align*}
t & \text{ is conducting } = (V^x \min V^y < V^{gn} - V_L) \\
p & \text{ is conducting } = (V^u \max V^v > V^{gp} + V_t)
\end{align*}
\]

where $V_t$ is called the threshold voltage and is assumed to be a positive constant that is small compared to $V_H - V_L$. For simplicity, all threshold voltages are assumed to be equal. The general case, with different threshold voltages, is discussed in remark 6.34 (section 6.3).

If a transistor is nonconducting, its pass nodes are not connected by that transistor. If a transistor becomes conducting, this can result in a change of the voltage values of the pass nodes. If one of the pass nodes is connected to a source and the other is not, then the voltage value of the latter is changed towards the voltage value of the former. The resulting voltage value, according to formula 6.0, of this latter pass node is investigated below.

Some consequences of 6.0

In 6.1 below some of the consequences of formula 6.0 for the n-transistor $nt$ are listed. The following assumption is made: \begin{align*}
(A \ z : N^z : V_L \leq V^z \leq V_H) \end{align*}. In 6.1b,c we also assume that $x$ is connected to a source, and that the initial voltage value of $y$ is an intermediate value, say between $V_L + 2V_t$ and $V_H - 2V_t$. The resulting voltage value of a node $z$ is denoted by $V^*z$.

For the p-transistor $pt$ similar consequences hold; they are listed in 6.2 below.

6.1 Remark: Some consequences of 6.0 for $nt$

\[\begin{array}{ll}
a & V^{gn} = V_L \Rightarrow 'nt is nonconducting' \\
& V^{gn} = V_L + V_t \Rightarrow 'nt is nonconducting'
\end{array}\]

\[\begin{array}{ll}
b & (V^{gn} = V_H \land V^x = V_L) \Rightarrow (V^*y = V_L) \\
& (V^{gn} = V_H \land V^x = V_H) \Rightarrow (V^*y = V_H - V_t)
\end{array}\]

\[\begin{array}{ll}
c & (V^{gn} = V_H - V_t \land V^x = V_L) \Rightarrow (V^*y = V_L) \\
& (V^{gn} = V_H - V_t \land V^x = V_H) \Rightarrow (V^*y = V_H - 2V_t)
\end{array}\]
6.2 Remark: Some consequences of 6.0 for \( p_t \)

\( a \)
\[ V_{gp} = V_H \Rightarrow \text{"} p_t \text{" is nonconducting} \]
\[ V_{gp} = V_{H-V_t} \Rightarrow \text{"} p_t \text{" is nonconducting} \]
\( b \)
\[ (V_{gp} = V_L \land V_{u} = V_H) \Rightarrow (V^*_{u} = V_H) \]
\[ (V_{gp} = V_L \land V_{u} = V_L) \Rightarrow (V^*_{u} = V_{L+V_t}) \]
\( c \)
\[ (V_{gp} = V_{L+V_t} \land V_{u} = V_H) \Rightarrow (V^*_{u} = V_H) \]
\[ (V_{gp} = V_{L+V_t} \land V_{u} = V_L) \Rightarrow (V^*_{u} = V_{L+2V_t}) \]

Voltage values like \( V_{H-V_t} \) and \( V_{H-2V_t} \), which are driven by a source but unequal to the voltage value of that source, are called mutilated signals. Formula 6.0 is illustrated further in examples 6.3a,b,c. All given examples are based on n-switches; p-switches behave in a converse fashion.

6.3 Example

\( a \)
Consider circuit C0 (fig. 6.1a). Let \( x_0, x_1 \) and \( x_3 \) be directly connected to an H-source (and, hence, have voltage value \( V_H \)). Let \( x_2 \) and \( x_4 \) initially have intermediate voltage values. This is expressed as source-connection \( \{ (x_0,(H)),(x_1,(H)),(x_2,\emptyset),(x_3,(H)),(x_4,\emptyset) \} \).

According to formula 6.0 the resulting voltage values of \( x_2 \) and \( x_4 \) are (cf. remark 6.1b):
\[ V^*_{x_2} = V_{H-V_t} \quad \text{and} \quad V^*_{x_4} = V_{H-V_t} \]

Notice that the signal is not mutilated further by the rightmost transistor.

\[ x_0 \xrightarrow{x_1} \xrightarrow{x_3} x_4 \]
\[ x_0 \xrightarrow{x_1} \xrightarrow{x_3} x_4 \]

figure 6.1a: C0 (ex. 6.3a) \hspace{1cm} figure 6.1b: C1 (ex. 6.3b,c)

\( b \)
Consider circuit C1 (fig. 6.1b). Let \( x_0, x_1, x_2 \) be directly connected to an H-source, and let \( x_3 \) and \( x_4 \) initially have intermediate voltage values. This is expressed as source-connection \( \{ (x_0,(H)),(x_1,(H)),(x_2,(H)),(x_3,\emptyset),(x_4,\emptyset) \} \).

According to formula 6.0 the resulting voltage values of \( x_3 \) and \( x_4 \) are (cf. rem. 6.1b,c):
\[ V^*_{x_3} = V_{H-V_t} \quad \text{and} \quad V^*_{x_4} = V_{H-2V_t} \]

Notice the difference with the previous example: the signal at \( x_4 \) is more mutilated than the signal at \( x_3 \).

\( c \)
Consider circuit C1 (fig. 6.1b) again. Let \( x_0, x_1, x_2 \) be directly connected to an H-source, an H-source, and an L-source respectively. Let \( x_3 \) and \( x_4 \) initially have intermediate voltage values. This is expressed as source-connection \( \{ (x_0,(H)),(x_1,(H)),(x_2,(L)),(x_3,\emptyset),(x_4,\emptyset) \} \).

According to formula 6.0 the resulting voltage values of \( x_3 \) and \( x_4 \) are (cf. rem. 6.1b,c):
\[ V^*_{x_3} = V_{H-V_t} \quad \text{and} \quad V^*_{x_4} = V_L \]

Notice that, although the signal at \( x_3 \) is mutilated, the signal at \( x_4 \) is not.
Mutilation degree

As demonstrated in example 6.3b, signals can be mutilated more than once. The *mutilation degree* of a signal \( L \) or \( H \) is defined as the number of times that signal is mutilated. That is, signals \( V_H - kV_x \) and \( V_L + kV_x \) (for \( 0 \leq k \leq 1 \)) are said to have mutilation degree \( k \). For a certain number \( i \) (\( 0 \leq i \leq 1 \)), a signal with mutilation degree \( i+1 \) is unable to control a transistor-gate (is no longer recognizable as \{H\} or \{L\}). As a result of this, ‘mutilation degree \( i+2 \)’ is an unrealistic notion. How much mutilation degree can be allowed depends, among other things, on the value of \( V_x \) (relative to the value of \( V_H \) \( V_L \)). The choice of the largest allowable mutilation degree for a signal at a gate (the number \( i \) above) is, therefore, regarded as a design rule. Usually, this number is chosen to be 0 or 1.

In our model we abstract from the fact that large mutilation degrees are unrealistic. We calculate the – abstract – mutilation degrees of signals using a generalisation of the rules given in remarks 6.1 and 6.2. After calculating these abstract mutilation degrees we will verify whether they do not exceed a certain value (given by the design rule mentioned above). This verification will be expressed in a new correctness criterion (section 6.2).

For calculational reasons, which are explained later on (remark 6.31, sect. 6.3), we introduce the mutilation degrees \( m \) and \( \alpha \) (\( m \in \mathbb{N} \), and \( \alpha \) is a fresh symbol). The domain of the mutilation degree of signals, \( \text{MD} \) (def. 6.9), is \( \{ j : 0 \leq j \leq m \} \cup \{ \alpha \} \), with the following interpretation for the elements:

- \( j, 0 \leq j \leq m \): the signal has mutilation degree \( j \)
- \( m \): the signal is maximally mutilated
- \( \alpha \): the signal is absent (no connection to that source type)

6.4 Example

In the resulting states of example 6.3, the pairs (mutilation degree of \( L \), mutilation degree of \( H \)) in nodes \( x_0, x_1, x_2, x_3, \) and \( x_4 \) (in that order) are:

- in 6.3a: \( (\alpha, 0), (\alpha, 0), (\alpha, 1), (\alpha, 0), \) and \( (\alpha, 1) \),
- in 6.3b: \( (\alpha, 0), (\alpha, 0), (\alpha, 0), (\alpha, 1), \) and \( (\alpha, 2) \),
- in 6.3c: \( (\alpha, 0), (\alpha, 0), (0, \alpha), (\alpha, 1), \) and \( (0, \alpha) \).

Towards a formalisation of mutilation degree

Before formalising the notion of mutilation degree, we will analyse the notion further using a generalisation of the remarks 6.1 and 6.2, and the interpretation of the domain of mutilation degree given above. We start with some examples.
6.5 Example
Consider the circuit consisting of one n-switch depicted in figure 6.2a below.
Let $x_1$ be connected to an H-source with mutilation degree $k$ (for H). Suppose that $x_2$ initially has an intermediate value. Now consider the following cases.

a) If $x_0$ is connected to an H-source with mutilation degree $n$ (for H), we conclude (by generalising remark 6.1 using the interpretation of MD given above) that the resulting signal at $x_2$ is $\{H\}$ with mutilation degree $(k+1) \max n$ if $k<n$, and $n$ if $k=n$.

b) If $x_0$ is connected to an L-source with mutilation degree $n$ (for L), we conclude (by generalising remark 6.1 using the interpretation of MD given above) that the resulting signal at $x_2$ is $\{L\}$ with mutilation degree $n$.

figure 6.2a: example 6.5

figure 6.2b: example 6.6

In the previous example we have seen how to generalise remark 6.1 for nodes with a single connection to a source. In the following example we consider a node with multiple connections to sources of the same type. In such a case, a less mutilated signal "overrides" a more mutilated signal of the same type. We are interested in the least mutilated signal only, which means we are interested in the best path to a source.

6.6 Example
Consider the circuit from example 6.3a, for convenience depicted in figure 6.2b above.
Let nodes $x_0, x_1, x_2,$ and $x_4$ be connected to an H-source with mutilation degree $n_0, k_0, k_1$, and $n_1$ respectively, with $k_0<n_0$ and $k_1<n_1$. Suppose node $x_2$ has an intermediate value when the two switches become conducting. The resulting value of $x_3$ then is $\{H\}$ with mutilation degree $(\max (k_0+1) \min n_0, k_1+1) \max n_1$ (compare with ex. 6.5a).

The mutilation degree in state $\Pi = (Q, \Gamma)$ for node $x$ and signal $X \in \{L, H\}$ is denoted by $md_{\Pi}^{x, X}$. With the help of the previous examples, we can now formulate the following (informal) description of mutilation degree.

6.7

$X \notin \Gamma \cdot x \iff md_{\Pi}^{x, X} = \alpha$

and $X \notin \Gamma \cdot x \land md_{\Pi}^{x, X} = k$

$\Rightarrow \text{the best path from } x \text{ to an } X\text{-source, i.e., the path causing the least mutilation for } X\text{-signals, causes mutilation } k \text{ for } X\text{-signals}$
In order to formalise expression 6.7, we will introduce two additional notions.

- The mutilating power of a switch for a signal $X\in \{L,H\}$ is (informally) the amount of mutilation caused by that switch when transporting an $X$-signal ($\alpha$ if the switch is nonconducting).

- The path quality of a bag of nodes $[x,y]$ for a signal $X\in \{L,H\}$ is (informally) the minimal amount of mutilation caused by any path (of switches) between $x$ and $y$ when transporting an $X$-signal ($\alpha$ if no conducting path between $x$ and $y$ exists).

6.8 Example
Consider again the circuit from example 6.5 with node $x_1$ connected to an $H$-source with mutilation degree $k$. The mutilating power of the $n$-switch is, for $H$: $k+1$, and for $L$: 0.

The path quality for $[x_0,x_2]$ is, for $H$: $k+1$, and for $L$: 0.

With the help of the latter notion, the mutilation degree can be formalised. The formalisation of the notions informally introduced in this section takes place in the next section.

6.1 Formalisation of Mutilation Degree

In this section the notions mutilation degree, mutilating power, and path quality, informally introduced in the previous section, are defined. In order to keep a good grip on the formalisation we first consider stable states only (in section 6.1.0), and assume that all input signals have mutilation degree 0. In section 6.1.1 mutilation degree for non-stable states is considered, which will turn out to be a relatively simple extension of the formalisation from section 6.1.0. The case that input signals can have nonzero mutilation degree (which is interesting when studying subcircuits, and, hence, for composability of the model) is discussed in remark 6.32 (sect. 6.3).

6.1.0 Mutilation Degree for Initial Behaviour

An additional correctness criterion for initial behaviour (as described in the previous section) requires that the mutilation degrees of signals at the gates of resulting states do not exceed a certain value (given by a design rule). If the initial behaviour of a circuit with a source connection is correct according to the correctness criteria defined in the previous chapters, all resulting states are stable (cf. corollary 5.11 and theorem 5.12). Consequently, in order to express this additional correctness criterion for initial behaviour, it suffices to define mutilation degree for stable states only.
In this section the notion mutilation degree is formalised for stable states. The additional correctness criterion is defined in section 6.2.
Throughout this section, natural m is used as a constant with the interpretation as given in section 6.0. The following definition of the domain of mutilation degree was already given in section 6.0; the interpretation of the elements of MD was also given in section 6.0.

6.9 Definition
MD, the domain of mutilation degree, and the order $\preceq$ on MD are defined by:

$$\text{MD} = \{ i : (N \downarrow \leq i \leq m) \lor i = \alpha : i \} \quad (\alpha \text{ is a fresh symbol})$$

$$i \preceq j = (i = \alpha) \lor (j = \alpha), \text{ for } \{i,j\} \subseteq \text{MD}$$

In the remainder of this section $C = \langle N, SW, L, g, p_n \rangle$ is a circuit, $\gamma \in \text{NST}$ a source-connection and $\Omega = \langle Q, L \rangle$ a state of $C$. A number of notions are defined for these specific $C, \gamma,$ and $\Omega$.
The dependency of the notions on $C, \gamma,$ and $\Omega$ is not mentioned, except when there is a danger of confusion otherwise (in that case we add $C, \gamma,$ and/or $\Omega$ as subscripts).

6.10 Definition
MUT, the set of mutilation functions (with typical elements: $\rho_0, \rho_1$), is defined by:

$$\text{MUT} = N - \{ L, H \} \cup \text{MD}$$

The order $\preceq$ on MUT is defined by:

$$\rho_0 \preceq \rho_1 = (A x, X : x \in N \land X \{ L, H \} : \rho_0 X \leq X \rho_1), \text{ for } \{\rho_0, \rho_1\} \subseteq \text{MUT}$$

The function mutilation degree for a state $\Omega$, denoted as $md_\Omega$, is a special element of MUT, satisfying the informal requirements given in expression 6.7. In the sequel, we formalise these requirements and prove that exactly one element of MUT satisfies them. First, the notions mutilating power and path quality, informally introduced in the previous section, are defined.

Mutilating power and path quality
The mutilating power of a switch intends to model the minimal degree of mutilation a signal transported by that switch has. To illustrate this, consider a switch $s$ with pass nodes $x$ and $y$, with node $x$ connected to an $X$-source with mutilation degree $n$. The mutilating power of $s$ for $X$, say $p$, must be such that the mutilation degree of $X$ in node $y$ equals $p \max n$ (cf. ex.6.5).

Obviously, if $s$ is nonconducting, node $y$ is not connected to an $X$-source, and, hence, the mutilation degree of $X$ in $y$ is supposed to be $\alpha$. Consequently, for nonconducting $s$ is $p = \alpha$.

If $Q \downarrow s$ and $t \downarrow s \neq \{X\}$, the passing $X$-signal is not mutilated further, and, hence, $p = 0$ (cf. ex.6.5b). If $Q \downarrow s \land t \downarrow s = \{X\} \land \Gamma (g, s) = \{X\}$ and $g \downarrow s$ has mutilation degree $k$ for $X$, then $p = 1 \max k$ (cf. ex.6.5a).
The case that \( Q \cdot s \land t \cdot s = \{X\} \land \Gamma (g \cdot s) \neq \{X\} \) is incorrect for stable states (\( \neg (g \cdot d) \)); we choose \( \nu = m \). This case is further analysed for nonstable states in sect. 6.1.1.

6.11 Definition

a Function \((1@i) \in MD - MD\) is defined by: \(1@i = \begin{cases} 1+i & \text{if } 0 \leq i < m \\ m & \text{if } i \in \{m, a\} \end{cases}\)

For brevity, we write \(1@i\) instead of \((1@i)\) in.

b \(mp \in MLT \rightarrow SW \land \{L, H\} - MD\), the mutilating power of a switch, is defined by:

\[
mp \cdot \rho \cdot s \cdot X = \begin{cases} \alpha, & \text{if } -Q \cdot s \\ 0, & \text{if } Q \cdot s \land t \cdot s \neq \{X\} \\ 1 \land mp \cdot (g \cdot s) \cdot X, & \text{if } Q \cdot s \land t \cdot s = \{X\} \land \Gamma (g \cdot s) = \{X\} \\ m, & \text{otherwise} \end{cases}
\]

where \( \rho \in MUT\), \( SW \cdot s\), and \( X \in \{L, H\}\).

In example 6.6 we have seen the importance of the best connection to a source. The mutilation caused by the best path between two nodes is modelled in the notion path quality.

6.12 Definition

The basic path quality and the path quality, \( bcp, cp \in \text{ MUT } = B2N \rightarrow \{L, H\} \rightarrow \text{ MD}\) are defined by:

\[
bpq \cdot \rho \cdot b \cdot X = (\min s : SW \cdot s \land \rho \cdot s = b : mp \cdot \rho \cdot s \cdot X)
\]

\[
pq \cdot \rho \cdot b \cdot X = (\min n, \{i : 0 \leq i < n : x_i \} \subseteq N \land b = [x_0, x_n]: (\max i : 0 \leq i < n : bpq \cdot \rho \cdot \{x_i, x_{i+1}\} \cdot X))
\]

where \( \rho \in MUT\), \( b \in B2N\), \( X \in \{L, H\}\),

and \( \min \) and \( \max \) are the minimum and maximum w.r.t. \( \leq \).

Notice that, since \( \min \emptyset = \alpha \) and \( \max \emptyset = 0 \), for \( \rho \in MUT\), \( b \in B2N\), \( N \cdot y \) and \( X \in \{L, H\}\):

- \( pq \cdot \rho \cdot [y,y] \cdot X = 0 \)
- \( (A : s : SW \cdot s : \rho \cdot s = b) \rightarrow bpq \cdot \rho \cdot b \cdot X = \alpha \)

6.13 Example (compare the results with those of examples 6.5, 6.6, and 6.8)

a Consider again examples 6.5 and 6.8 (fig. 6.2a) with node \( x_1 \) connected to an H-source with mutilation degree \( k \). That is, \( 1 \cdot x_1 = \{H\}, Q \cdot x_1, p \cdot x_1 \cdot L = \alpha, \) and \( p \cdot x_1 \cdot H = k \).

Then: \( mp \cdot p \cdot s \cdot L = 0, mp \cdot p \cdot s \cdot H = 1 \land k \), \( pq \cdot \{x_0, x_2\} \cdot L = 0 \), and \( pq \cdot \rho \cdot [x_0, x_2] \cdot H = 1 \land k \).

b Consider again the circuit from ex. 6.6 (fig. 6.2b). Assume that both gate nodes are connected to an H-source and that both switches are conducting. Let the mutilation degrees be given by: \( p \cdot x_1 \cdot L = \alpha \land p \cdot x_1 \cdot H = k \land p \cdot x_3 \cdot L = \alpha \land p \cdot x_3 \cdot H = k \).

Then: \( mp \cdot p \cdot s \cdot L = 0, mp \cdot p \cdot s \cdot H = 1 \land k \), and \( mp \cdot p \cdot s \cdot H = 1 \land k \).

Consequently \( pq \cdot \rho \cdot [x_0, x_2] \cdot L = 0 \) and \( pq \cdot \rho \cdot [x_0, x_2] \cdot H = (1 \land k) \max (1 \land k) \).
Mutational degree as a fixpoint of function $h$

In order to define the notion mutational degree, we introduce the following auxiliary function $h$.

6.14 Definition

$h : \text{MUT} \rightarrow \text{MUT}$ is defined by:

$$h : \rho \cdot x \cdot X \equiv (\text{min}_y \cdot N\cdot y \land x : \gamma : m_d \cdot [x,y] \cdot X) , \text{MUT} \cdot \rho \land N\cdot x \land \{L,H\} \cdot X$$

Notice that the expression $$(\text{min}_y \cdot N\cdot y \land x : \gamma : m_d \cdot [x,y] \cdot X) ,$$ where $m_d$ is the mutational degree, can be interpreted as the best path from node $x$ to an X-source. Since we assumed all input signals to have mutational degree 0, this must equal $m_d \cdot x \cdot X$. Consequently, the mutational degree $m_d$ is a fixpoint of $h$. In the sequel we prove that $h$ has exactly one fixpoint. As a result, $m_d$ can be defined by (corollary 6.20):

$$m_d \cdot x \cdot X = (\text{min}_y \cdot N\cdot y \land x : \gamma : m_d \cdot [x,y] \cdot X) , \text{for } N\cdot x \text{ and } X \in \{L,H\}.$$  

Lemmas 6.15 and 6.16, corollary 6.17, and theorem 6.18 deal with the proof that $h$ has only one fixpoint. The mutational degree then is defined in 6.19.

6.15 Lemma

$(\text{MUT}, \leq)$ is a complete lattice.

Proof

The join and meet of $\rho_0$ and $\rho_1$, $(\rho_0 \sqcup \rho_1) \in \text{MUT}$, are defined by:

$$\text{join}(\rho_0 : \rho_1) \cdot y \cdot X = \rho_0 \cdot y \cdot X \land \rho_1 \cdot y \cdot X$$

$$\text{meet}(\rho_0 : \rho_1) \cdot y \cdot X = \rho_0 \cdot y \cdot X \lor \rho_1 \cdot y \cdot X$$

Consequently, $(\text{MUT}, \leq)$ is a lattice. Since MUT is finite, this lattice is complete.

6.16 Lemma

a) $(\text{MUT})$ is monotonic w.r.t. the order $\leq$ on MD.

b) mp is monotonic in its first argument w.r.t. the pointwise order $\leq$, i.e.:

For $\rho_0, \rho_1 \in \text{MUT}$: $\rho_0 \leq \rho_1 \Rightarrow (\text{mp} : \rho_0 \cdot x \cdot X \leq \text{mp} : \rho_1 \cdot x \cdot X)$

c) bpq is monotonic in its first argument w.r.t. the pointwise order $\leq$.

d) pq is monotonic in its first argument w.r.t. the pointwise order $\leq$.

Part a of this lemma follows directly from the definition of $(\text{MUT})$ (def. 6.7a). Parts b, c, and d follow directly from parts a, b, and c respectively, using definitions 6.7 (mp) and 6.8 (bpq).

6.17 Corollary

a) $h$ is monotonic.

b) $h$ has a least and a greatest fixpoint.

Cor. 6.17a follows directly from lemma 6.16d, using def. 6.14 $(h)$. Cor. 6.17b follows directly with the theorem of Knaster-Tarski (th. A2, appendix A) from lemma 6.15 and cor. 6.17a.
6.18 Theorem

\( h \) has exactly one fixpoint.

Proof

Let \( \rho_0 \) and \( \rho_1 \) be the least and the greatest fixpoint of \( h \) (see theorem 3.15).

Let \( P_0 \) and \( P_1 \), both of type \( \{L,H\} \rightarrow MD \rightarrow \mathbb{B} \), be defined by:

\[
\begin{align*}
P_0 \times i &= (A_{x_j : SW \times} \land 0 \leq j \leq i : (mp \cdot p_{0 \times} \cdot X = j) = (mp \cdot p_{1 \times} \cdot X = j)) \\
P_1 \times i &= (A_{x_j : N \times} \land 0 \leq j \leq i : (p_{0 \times} \cdot X = j) = (p_{1 \times} \cdot X = j))
\end{align*}
\]

where \( X \in \{L,H\} \) and \( i \in MD \).

Let \( X \in \{L,H\} \). The proof of \( P_1 \times \alpha \) consists of the following steps:

1. \( P_0 \times 0 \)
2. \( P_1 \times 0 = P_1 \times \alpha \)

In conclusion, notice that \( P_1 \times \alpha = (\rho_0 = \rho_1) \).

In proofs of 0) - 3)

0) Let \( SW \times \)

\[
\begin{align*}
mp \cdot p_{0 \times} \cdot X &= 0 \\
&= [\text{def. 6.11b (mp)}] \\
&= Q \land (m=0 \lor t \neq [X]) \\
&= [\text{def. 6.11b (mp)}] \\
&= mp \cdot p_{1 \times} \cdot X = 0
\end{align*}
\]

Hence \( \text{definition of P0} \) \( P_0 \times 0 \)

1) Let \( i \in 0 \leq i \leq m \). Then:

\[
\begin{align*}
P_0 \times i &= [\text{definition of P0}] \\
&= (A_{x_j : SW \times} \land 0 \leq j \leq i : (mp \cdot p_{0 \times} \cdot X = j) = (mp \cdot p_{1 \times} \cdot X = j)) \\
&= [\text{def. 6.12 (bpq)}] \\
&= (A_{b_j : B2N \times} \land 0 \leq j \leq i : (bpq \cdot p_{0 \times} \cdot B \cdot X = j) = (bpq \cdot p_{1 \times} \cdot B \cdot X = j)) \\
&= [\text{def. 6.12 (pq)}] \\
&= (A_{b_j : B2N \times} \land 0 \leq j \leq i : (pq \cdot p_{0 \times} \cdot B \cdot X = j) = (pq \cdot p_{1 \times} \cdot B \cdot X = j)) \\
&= [\text{def. 6.14 (h)}] \\
&= (A_{x_j : N \times} \land 0 \leq j \leq i : (h \cdot p_{0 \times} \cdot X = j) = (h \cdot p_{1 \times} \cdot X = j)) \\
&= [\rho_0 \text{ and } \rho_1 \text{ are fixpoints of } h] \\
&= (A_{x_j : N \times} \land 0 \leq j \leq i : (p_{0 \times} \cdot X = j) = (p_{1 \times} \cdot X = j)) \\
&= [\text{definition of P1}] \\
P_1 \times i
\end{align*}
\]
2) Let \( 0 \leq i < m \). Then:

\[
P_1 X_i
\]

= \{ \text{definition of } P_1 \}

= \{ 0 \text{ and def. 6.11 (mp and (1\#)),} \}

= \{ 0 \leq (1\# k) < m = (\emptyset \leq k = (1\# k) - 1) \text{ and } (1\# k = m) = (k \in \{ m-1, m, \alpha \}) \}

= \{ \text{definition of } P_0 \}

= \{ \text{definition of } P_1 \}

= \{ 0 \leq j \leq i + 1 \} : (mp p_{0 \# s} X = j) = (mp p_{1 \# s} X = j) \}

= \{ \text{definition of } P_1 \}

= \{ \text{definition of } P_1 \}

= \{ \text{definition of } P_1 \}

= \{ \text{definition of } P_1 \}

= \{ \text{definition of } P_1 \}

\square \text{ (end proof 6.18)}

\textbf{Definition of mutilation degree}

On account of theorem 6.18, the mutilation degree can be defined as follows (cf. the explanation below definition 6.14).

\textbf{6.19 Definition}

For a state \( \Pi \), the \textit{mutilation degree} \( md_{\Pi} \) is defined as the fixpoint of \( h \).

\textbf{6.20 Corollary}

For a state \( \Pi \), the mutilation degree \( md_{\Pi} \) can alternatively be defined by:

\[
md_{\Pi} X = \left( \min y : N \wedge X \wedge y \wedge \neg y \wedge X = \{ x \mid y \in x \} \right)
\]

This is a direct result of definitions 6.14 (h) and 6.19 (md_{\Pi}), and theorem 6.18.

\textbf{6.21 Theorem}

For every \( \Pi \in ST_1 \):

\[
( A \rho : \text{MUT}\_p : (E n : 1 \leq n < \# \text{MUT} : h^p \rho = md_{\Pi} ))
\]

where \#MUT is the number of elements of MUT.

The proof of theorem 6.21 is fairly simple using cor. 6.17a, th. 6.18, and finiteness of MUT.

Theorem 6.21 is interesting, since it allows \( md_{\Pi} \) to be calculated efficiently for every state \( \Pi \).
The following lemma gives an expression for $md_{\Pi}$ for stable states $\Pi$ that is comparable to expression 6.7, and hence, gives a convincing validation of the given formalisation of $md_{\Pi}$.

### 6.22 Lemma

Let $\Pi = (Q, \Gamma)$ satisfy stable3-$\gamma$-$\Pi$. Then, for all $x \in N$, $x \in \{L, H\}$, and $k \in M\Delta\{\alpha\}$:

- $X \in \Gamma \cdot x \iff md_{\Pi \cdot x} X = \alpha$
- $X \in \Gamma \cdot x \wedge md_{\Pi \cdot x} X = k \iff k = \min\{y : N \cdot y \wedge x \in \gamma \cdot y : pq \cdot md_{\Pi \cdot x} [x, y] \cdot X\}$

Lemma 6.22b follows directly from lemma 6.22a and corollary 6.20. The proof of lemma 6.22a is given below.

**Proof of lemma 6.22a**

1. $md_{\Pi \cdot x} X = \alpha$
   
   $$ = \{3.19\} $$
   
   $$(E \ y : N \cdot y \wedge X \in \gamma \cdot y : pq \cdot md_{\Pi \cdot x} [x, y] \cdot X \neq \alpha)$$

   $$ = \{\text{note 0}\} $$

   $$(E \ y : N \cdot y \wedge X \in \gamma \cdot y : cp\cdot Q \cdot [x, y])$$

   $$ = \{\text{definition 4.6 (R), and from stable3-$\gamma$-$\Pi : \Gamma = R \cdot (\gamma \uplus \text{store} \cdot \Gamma) \cdot Q}\} $$

   $X \in \Gamma \cdot x$

**Notes**

0. Let $b \in B2N$. Then:

   $pq \cdot md_{\Pi \cdot b} X \neq \alpha$

   $$ = \{\text{def. 6.12 (pq)}\} $$

   $$(E n \cdot \{i : 0 \leq i \leq s : x_i\} : n \cdot i : 0 \leq i \leq s : x_i \in N \wedge b = [x_0, x_2])$$

   $$ = \{\text{note 1}\} $$

   $$(E n \cdot \{i : 0 \leq i \leq s : x_i\} : n \cdot i : 0 \leq i \leq s : x_i \in N \wedge b = [x_0, x_2])$$

   $$ = \{\text{def. 1.14 (cp)}\} $$

   $cp\cdot Q \cdot b$

1. Let $b \in B2N$. Then:

   $bpq \cdot md_{\Pi \cdot b} X \neq \alpha$

   $$ = \{\text{def. 6.12 (bpq)}\} $$

   $$(E s : SW \cdot s \wedge psr = b : mp \cdot md_{\Pi \cdot b} r \cdot X \neq \alpha)$$

   $$ = \{\text{def. 6.11b (mp)}\} $$

   $$(E s : SW \cdot s \wedge psr = b : Q \cdot s)$$

   $$ = \{\text{def. 1.14 (bcp)}\} $$

   $bcp\cdot Q \cdot b$
6.1.1 Mutilation Degree for Dynamic Behaviour

In order to express the additional correctness criterion for dynamic behaviour as described in section 6.0 (and defined in section 6.2), we need to define the notion of mutilation degree for nonstable states. This definition, given in this section, turns out to be a simple extension of the formalisation given in the previous subsection.

As in the previous section, all notions are defined for a specific circuit \( C \), a source-connection \( \gamma \in \text{NST} \), and an extended state \((\Pi_l, r, p_c)\) of \( C \), with \( \Pi = (Q, \Gamma) \).

The main difference for the formalisation for nonstable states is the definition of mutilating power of switches. The analysis of this notion made in the previous section (above def. 6.11) needs to be reconsidered.

**Mutilating power**

Consider again a switch \( s \) with pass nodes \( x \) and \( y \) where node \( x \) is connected to an X-source with mutilation degree \( n \). Let the mutilating power of \( s \) for \( X \) be equal to \( p \).

Since state \( \Pi \) need not be stable, the pass-delay counter \( pc \) need not be 0. Recall that switch \( s \) is transporting if \((Q \land Z \land pc) \land s \) (cf. ch.5, def. 5.2). Let \( P = (Q \land Z \land pc) \).

As in sect. 6.1.0, if \( \neg P \land s \) then node \( y \) is not connected to the X-source, and, hence, \( p = 0 \).

If \( P \land s \land t \land s \neq (X) \) then the passing X-signal is not mutilated any further, and, hence, \( p = 0 \).

If \( P \land s \land t \land s = (X) \land \Gamma(g \land s) \neq \{X\} \) and \( g \land s \) has mutilation degree \( k \) for \( X \), then \( p = 1 \oplus k \).

The case that \( P \land s \land t \land s = (X) \land \Gamma(g \land s) = \{X\} \) needs to be considered more closely. For a stable state (cf. sect.6.1.0) this case means the state is not correct according to correctness criterion \( \neg \text{gcd} \), and, hence, we are free to choose a value for \( p \) (in 6.1.0: m). For a nonstable state this case does not necessarily imply that the state is incorrect. We distinguish the following subcases of the case \( P \land s \land t \land s = (X) \land \Gamma(g \land s) = \{X\} : \)

- \( c1 \quad \Gamma(g \land s) = \{L, H\} \setminus \{X\} \), in which case the reaction-delay is active in \( s \) (cf. chapter 3)
- \( c2 \quad \Gamma(g \land s) = \{v\} \), in which case \( s \) is controlled by a stored value (cf. chapter 4)
- \( c3 \quad \text{otherwise} \), in which case the state is incorrect since \( \neg \text{gcd}(Q, \Gamma) \land s \) (cf. chapter 4)

where (in \( c2 \) \( v \) is defined by: \( (X = L \land \gamma = l) \land (X = H \land \gamma = h) \).

In cases \( c1 \) and \( c2 \) the state is correct, but the mutilation degree of the gate node of \( s \) is irrelevant for the mutilating power of \( s \). In these two cases the conductance (quality) of \( s \) depends on the previous state(s), and, hence, it seems reasonable to define the mutilating power of \( s \) equal to the mutilating power of \( s \) in the previous state of the circuit.

In case \( c3 \) the state is incorrect and, as before, the mutilating power of \( s \) is defined to be maximal (m).
Let \( p \circ : SW \to \{L,H\} \to MD \). Function \( p \circ \) is interpreted as the mutilating power of switches in the previous state. Note that this function is independent of any mutilation function; we assume it to be part of the given information of the previous state.

6.23 Definition

The mutilating power function \( mp \in \text{MUT} \to SW \to \{L,H\} \to MD \) is defined by:

\[
mp(p \circ \cdot s \cdot X) = \begin{cases} 
\alpha & \text{if } \neg P \cdot s \\
0 & \text{if } P \cdot s \land s \not\in \{X\} \\
1 \circ p \circ (g \cdot s) \cdot X & \text{if } P \cdot s \land s \in \{X\} \land \Gamma \cdot (g \cdot s) = \{X\} \\
p \circ s \cdot X \min m & \text{if } P \cdot s \land s \not\in \{X\} \land \Gamma \cdot (g \cdot s) \in \{\{Y\},\{v\}\} \\
m & \text{if } P \cdot s \land \neg g \circ d \cdot l \cdot (Q, \Gamma) \cdot s 
\end{cases}
\]

where \( p \circ \in \text{MUT} \), \( SW \cdot s \cdot P = (Q \land Z \cdot p c) \), \( \{X, Y\} = \{L, H\} \), \( (X = L \circ v = l) \land (X = H \circ v = h) \), and \( p \circ : SW \to \{L, H\} \to MD \) is the mutilating power of switches in the previous state.

6.24 Remark

Notice that \( mp \) (def.6.23) is a conservative extension of \( mp \) (def.6.11b), since:

\[
\text{stable}3 \cdot y(\text{Q}, \Gamma) \land \text{cgdo} \cdot \Gamma \\
\Rightarrow (pc = 0) \land \text{coco}0 \cdot \Gamma \cdot Q \land \text{cgdo} \cdot \Gamma \\
\Rightarrow (A \cdot s : P \cdot s : \text{gdo} \cdot \Gamma \cdot s \land \Gamma \cdot (g \cdot s) = t \cdot s) \\
\Rightarrow mp \in mp
\]

Mutilation degree

The remaining part of the formalisation of mutilation degree for nonstable states is straightforward: the notions \( b \cdot pq \), \( mp \) \( \Pi \), and the function \( h \) are defined in the same way as in section 6.1.0, using \( mp \) instead of \( mp \) (in the definition of \( bpq \)). The validity of the results from the previous section is easily verified for this case using the existing proofs (sect. 6.1.0). Note that for nonstable states the notions \( b \cdot pq \) and \( md \) \( \Pi \) depend on the previous mutilating power function \( p \circ \). The mutilation degree \( md \) \( \Pi \) for nonstable states \( \Pi \) is, as explained above, defined by (cf. cor. 6.20):

\[
md \cdot s \cdot X = (\min y : N \cdot y \land X \cdot \gamma \cdot y : pq \cdot md \cdot s \cdot \{x, y\} \cdot X ) , \text{ for } N \cdot s \text{ and } X \in \{L, H\}.
\]

Let \( \Pi 0 \) be a state with \( \Pi 1 \) as a successor. Let the mutilating power function of \( \Pi 0 \) (to be used in the definition of \( mp \) for \( \Pi 1 \)) be \( p \circ \). The new mutilating power function (of \( \Pi 1 \)) then equals \( mp \cdot \text{md} \cdot s \).

6.25 Example

Consider the circuit depicted in fig. 4.0a (ex. 4.0, cf. also ex. 4.38), a cascade of two inverters. Starting in a stable state with \( \Gamma \cdot x = \{H\} \), and hence \( \Gamma \cdot y = \{L\} \) and \( \Gamma \cdot z = \{H\} \), input node \( x \) is changed from \( \{H\} \) to \( \{L\} \). If the delays are such that node \( y \) becomes \( \{L\} \) before it becomes \( \{H\} \), then the mutilating power of switch \( x \); while \( \Gamma \cdot y = \{l\} \), remains 0 for \( L \). Consequently, the mutilation degree in \( z \) remains 0 (for \( H \); later for \( L \)).
6.2 Correctness Criteria due to Imperfectness of Switches

In this section the correctness criterion due to imperfectness of switches, described in section 6.0, is formalised. After the correctness criterion is defined for initial behaviour as well as for dynamic behaviour of circuits, the usage of it is illustrated by means of a familiar example.

Let \( \Pi \) be a state with mutilation degree \( \text{md}_{\Pi} \). As explained in section 6.0, for a certain value \( u \) (\( u \) for ultimate, \( 0 \leq u \leq m \)) given by a design rule, signals with mutilation degree larger than \( u \) are not allowed to control a switch. Consequently, the additional correctness criterion due to imperfectness of switches, can, for state \( \Pi \), be expressed as \( \text{ML}(u) \cdot \text{md}_{\Pi} \), where \( \text{ML}(u) \) is defined as follows.

6.26 Definition
For \( u \in \text{ML}(m,\alpha) \) criterion \( \text{ML}(u) \cdot \text{MUT} \rightarrow \mathbb{R} \) is defined as, for \( \text{md} \in \text{MUT} : \)
\[
\text{ML}(u) \cdot \text{md} = (\forall \pi, X : \text{SW}s \land X \in \{ L, H \} : \text{md}(g,s) = \alpha \lor \text{md}(g,s) \leq u ) .
\]
(\( \text{ML} \) stands for Mutilation degree Limit.)

The additional correctness criterion due to imperfectness of switches requires that the mutilation degrees in all resulting states (and, in case of dynamic behaviour, also in all intermediate states) satisfy \( \text{ML}(u) \).

Initial Behaviour
Let \( C \) be a circuit and \( \gamma \) a source-connection of \( C \). Let the correctness criterion for initial behaviour, as formalized in the previous chapters, i.e. \( \text{WM0-C} \cdot \gamma \land \text{CF} \cdot \gamma_s \), hold.

The resulting states are all stable and, abstracted from stored charges, equal to \( \gamma_s \). Since all these states are completely gate defined (\( \text{WM0-C} \cdot \gamma \)), and, hence, equal to \( \gamma_s \) at the gate nodes, the additional correctness criterion for initial behaviour is: \( \text{ML}(u) \cdot \text{md}_{\gamma_s} \). Conclusion:

6.27 The complete correctness criterion for initial behaviour is:
\[
\text{WM0-C} \cdot \gamma \land \text{CF} \cdot \gamma_s \land \text{ML}(u) \cdot \text{md}_{\gamma_s} .
\]

Dynamic Behaviour
The mutilation degree of a nonstable state depends not only on the state itself but also on its pass-delay counter and the mutilating power function of the previous state (cf. def. 6.23). For an extended state \( (\Pi_{ri}, \pi, r) \) with \( r \) as the previous mutilating power function, we therefore define resulting state lists (compare with def. 5.22) as pairs \( (\Psi, M) \in \text{O}^{(ST1)} \times \text{O}^{(MUT)} \) where \( M \) is the mutilation degree of state \( \Psi_{ri} \).
6.28 Definition
For \(rd, pd \in DC\), \(\gamma \in \text{NST}, (\Pi, rc, pc) \in \text{ST1} \times DC \times DC\), and \(p \in \text{SW} - \{L,H\} - \text{MD}\),
the function \(RSLA_{rd,pd} - \gamma((\Pi, rc, pc), p) \in \mathcal{E}(\text{ST1}) \times \mathcal{E}(\text{MUT}) \rightarrow \mathbb{R}\) is defined by:
\[
RSLA_{rd,pd} - \gamma((\Pi, rc, pc), p) : (\Psi, M) = (B \{rl_i,p_i,mpl\} : \mathcal{E}(\text{DC}) - rl \land \mathcal{E}(\text{DC}) - pl \land \mathcal{E}(\text{SW} - \{L,H\} - \text{MD}) - mpl \land (\Psi_0, rl_0, pl_0) = (\Pi, rc, pc) \land mpl_0 = p \land (A i : \text{next3}_{rd,pd} - \gamma((\Psi_i, rl_i, pl_i) - ((\Psi_{i+1}, rl_{i+1}, pl_{i+1})) \land (A i, \Theta : 0i \land \Theta = ((\Psi_i, rl_i, pl_i), mpl_i) : M_i = md_i \land mpl_{i+1} = mpl_0 - M_i))
\]

Remark It is easily verified (using def. 5.22 \(RSL3_{rd,pd}\) and 6.28 \(RSLA_{rd,pd}\)) that:
\[
RSLA_{rd,pd} - \gamma((\Pi, rc, pc), p) : (\Psi, M) = RSL3_{rd,pd} - \gamma((\Pi, rc, pc), \Psi)
\]
\[
RSL3_{rd,pd} - \gamma((\Pi, rc, pc), \Psi) = (E M : RSLA_{rd,pd} - \gamma((\Pi, rc, pc), p) : (\Psi, M))
\]

The additional correctness criterion for dynamic behaviour now requires that for all resulting state lists \((\Psi, M)\) all mutilation degrees \(M_1\) satisfy \(ML(u)\). This requirement is combined with the existing correctness criterion \(WF3_{D, C} - \gamma\) (def. 5.23), and defined below.

6.29 Definition
For \(D \subseteq DC \times DC\), \(WF4_D\) is defined as follows:
\[
WF4_D - \gamma((\Pi, rc, pc), p) = (A r d, pd : (rd, pd) \in D : RSLA_{rd, pd} - \gamma((\Pi, rc, pc), p) \in (cgDL \land CFL \land FCAI \land cstr) \times \text{MLL}(u))
\]

where
\(\text{MLL}(u) \in \mathcal{E}(\text{MUT}) \rightarrow \mathbb{R}\) is defined by: \(\text{MLL}(u) - M = (A i : 0i \leq \text{MLL}(u) - M_i)\)

The following remark shows that the criterion added to \(WF3\) is, as intended, the requirement that the mutilation degrees of all intermediate and resulting states satisfy \(ML(u)\).

Remark It is easily verified (using the previous remark, def. 5.23 \(WF3\) and 6.29 \(WF4\)) that:
\[
WF4_{D, \gamma} - \gamma((\Pi, rc, pc), p) = WF3_{D, \gamma} - \gamma((\Pi, rc, pc), \Psi)
\]
\[
\land (A r d, pd, \Psi, M : (rd, pd) \in D \land RSLA_{rd, pd} - \gamma((\Pi, rc, pc), p) : (\Psi, M) : \text{MLL}(u) - M)
\]

This section concludes with a familiar example (also discussed in examples 2.15, 4.38, 5.27) that illustrates the usage of mutilation degree, and that demonstrates the influence of the design rule — the choice of \(u\) — on the correctness criteria defined in this section.
6.30 Example

Consider the circuit of figure 6.3a. This circuit is discussed in example 4.38, where the state transitions depicted in figure 6.3b are investigated. In figure 6.3b the transitions labelled with ‘i’ denote an input change, and the transitions labelled with ‘+’ denote one or more state transitions resulting from the input change. Here we are not interested in the intermediate state transitions (they are discussed in ex. 4.38), but we investigate the mutilation degrees in the states of figure 6.3b.

The mutilation degrees of the signals at the input nodes x and y are, as usual, supposed to be 0 (or α for absent signals).

Notice that states Π0, Π2, Π3, and Π5 are stable states. We will investigate the mutilation degrees of the nodes. First, notice that the only node that can have mutilated signals (md ϵ {0, α}) is node z0. Even if the signal at node z0 is mutilated, the signal at node z1 is not.

In state Π0, the mutilation degree for L in z0 is 0, since the signal is transported by n-switch s0. Notice that, in state Π1, this mutilation degree remains 0, since the reaction delay is active in s0 (cf. case c/ in the analysis of mutilating power in sect. 6.1.1). In state Π2, the mutilation degree for L in z0 equals 1, since the signal is transported by p-switch s1. Notice that, in states Π3 and Π4, the mutilation degree for L in z0 also is 1, in state Π4 due to retaining of the mutilating power of s1 (similar as before; case c/). In the final state, Π5, the mutilation degree for H in z0 is 1, due to n-switch s0.

\[
\begin{align*}
\text{Figure 6.3a: circuit ex. 6.30} & \quad \text{Figure 6.3b: ex. 6.30} \\
\end{align*}
\]

The correctness of states Π2 – Π5 depends on the chosen design rule that determines the maximal allowable mutilation for signals at gate nodes (the value of α).

If all gate signals are supposed to be non-mutilated, i.e. if α = 0, then the mutilation degrees of these four states do not satisfy \( ML(u) \).

For the design rule allowing mutilation degree 1 for signals at gate nodes, i.e. if α = 1, these four states are correct, i.e., their mutilation degrees satisfy \( ML(u) \). Interestingly in this case is that, although a signal at an internal node (z0) is mutilated, the signal at the output node (z2) is not (we return to this point in chapter 7).

\( \Box \) (end example 6.30)
6.3 Remarks on the Formalisation

In this section three aspects of the given formalisation are discussed. In remark 6.31 the choice of set MD is discussed, in particular the role of its element m (maximally mutilated). Remarks 6.32 and 6.34 discuss the two assumptions made in sections 6.0 and 6.1, namely 'input signals have mutilation degree 0' and 'all threshold voltages are equal'. These assumptions are made to simplify the initial formalisation. However, extensions of the model for the general cases, i.e., allowing nonzero input signals and multiple threshold voltages, turn out to be relatively simple.

6.31 The Role of Element m of MD

The maximal degree of mutilation allowed for a signal that controls a gate, is given by a design rule. The verification of such a rule is expressed in correctness criterion ML(u). Whether the mutilation degree satisfies ML(u) can be verified for all u < m. The choice of m therefore depends on the largest u for which ML(u) needs to be checked (m = 1 + largest u).

Consider the circuit depicted in figure 6.4 with source-connection \( y = \{(x,H),(y,\emptyset)\} \). Notice that the circuit has two stable states, viz. y itself and \( \Gamma = \{(x,H),(y,H)\} \). For the second stable state, \( \Gamma \), the mutilation degree of y satisfies: \( md \cdot y \cdot H = 1 \circ md \cdot y \cdot H \) (cf. 6.11, 6.12, 6.20). The only element k of MD satisfying \( k = 1 \circ k \) is m. This means that in state \( \Gamma \) the signal H in y is maximally mutilated (i.e. worthless - denoting that \( \Gamma \) is an unrealistic stable state). Notice, from this example, that the existence as well as the uniqueness of a solution for the expression for md given in corollary 6.20 depends on the following aspects of the definition of \( (1 \circ) : 1 \circ m = m \) and \( 1 \circ \alpha = m \).

![Figure 6.4](image)

Choosing MD as an infinite set is possible, but the set must contain (in order to guarantee the above mentioned existence and uniqueness) an element playing the same role as m above.

A possible choice is \( \mathbb{N} \cup \{0,\alpha\} \) where \( 1 \circ \) is defined by: \( 1 \circ i = 1 + i \) for \( i \in \mathbb{N} \), and \( 1 \circ \alpha = \omega \) for \( k \in \{0,\alpha\} \).

Choosing MD as a finite set, however, has the calculational advantage that md can be calculated in a finite number of steps, cf. theorem 6.21.

\( \Box \) (end remark 6.31)
6.32 MUTILATED INPUT SIGNALS

In section 6.1 the assumption is made that all input signals have mutilation degree 0. Considering the general case, i.e., allowing nonzero input signals, is interesting for the study of subcircuits, and, hence, for the composability of the model.

In order to allow possible mutilations at input signals, we need to give, instead of a source-connection $\gamma$, a pair $(\gamma, imd) \in NSTxMUT$, where $imd$ is interpreted as the initial mutilation degree of source-connection $\gamma$. Obviously, such a pair must satisfy:

$(0)$ \hspace{1cm} $(A \ y, X : N \cdot y \land X \in \{L, H\} : imd \cdot y \cdot X \neq \alpha) \iff X \in \gamma \cdot y$

Consider again example 6.5a. In this example the pair $(\gamma, imd)$ is given by:

$\gamma = \{(x_0, H), (x_1, H), (x_2, H)\}$, and

$imd \cdot x_0 \cdot H = n$, \hspace{0.5cm} $imd \cdot x_1 \cdot H = k$, \hspace{0.5cm} and \hspace{0.5cm} $imd \cdot x \cdot X = \alpha$ for $(x, X) \in \{(x_0, H), (x_1, H)\}$

In the resulting state, node $x_2$ has value $\{H\}$, with mutilation degree $1 \oplus k$ max $n$ (cf. ex. 6.5a).

For this general case, the mutilation degree function $md'$ can be defined by:

$(1)$ \hspace{1cm} $md'_{p \cdot x} \cdot X = (\min y : N \cdot y \cdot pq \cdot md'_{p \cdot x}, y \cdot X \max imd \cdot y \cdot X), \text{ for } N \cdot x$ and $X \in \{L, H\}$.

The proof of the correctness of this definition is similar to the one given in section 6.1, using:

$(2)$ \hspace{1cm} $h' \cdot p \cdot x \cdot X = (\min y : N \cdot y \cdot pq \cdot p \cdot [x_0] \cdot X \max imd \cdot y \cdot X), \text{ for } N \cdot x$ and $X \in \{L, H\}$.

Notice that the case with non-mutilated input signals (sect. 6.1) is a special case of the general case described here. Assuming non-mutilated input signals means assuming that the pair $(\gamma, imd)$ satisfies, besides $(0)$:

$(3)$ \hspace{1cm} $(A \ y, X : N \cdot y \land X \in \{L, H\} : imd \cdot y \cdot X \in \{0, \alpha\})$.

Then:

$h' \cdot x \cdot X$

$= \{\text{def. (2)}\}$

$(\min y : N \cdot y \cdot pq \cdot p \cdot [x_0] \cdot X \max imd \cdot y \cdot X)$

$= \{\text{range split and assumptions (0) and (3)}\}$

$(\min y : N \cdot y \land X \notin y \cdot y : pq \cdot p \cdot [x_0] \cdot X \max 0)$

$\min \{\min y : N \cdot y \land X \notin \gamma \cdot y : pq \cdot p \cdot [x_0] \cdot X \max \alpha\}$

$= \{\text{calculus, using def. 6.9 (MD)}\}$

$(\min y : N \cdot y \land X \notin \gamma \cdot y : pq \cdot p \cdot [x_0] \cdot X \max 0)$

$= \{\text{def. 6.14 (h)}\}$

$h' \cdot x \cdot X$

And, hence, if $(\gamma, imd)$ satisfies $(0)$ and $(3)$, then: $md' = md$. 

6.33 Example

a Consider the circuit of figure 6.5a, the usual C-mos inverter. For the source-connection\( \gamma = \{s,\{H\}\}, (y,\emptyset) \) and every insd with \( \text{insd} \cdot x \cdot H \leq u \), the circuit is correct, and has resulting state \( \{s,\{H\}\}, (y,\{L\}) \) with \( \text{insd} \cdot y \cdot L = 0 \). This means that the inverter "demutillates", an effect already observed in example 6.30.

b Consider the circuit depicted in figure 6.5b, a two-transistor-switch or pass-transistor. The input assumption for this circuit is that the allowed source-connections \( \gamma \) satisfy \( \gamma \cdot y = \emptyset \) and \( \gamma \cdot z \cdot \bar{y} = \{\{L\},\{H\}\} \). If \( \gamma \cdot z = \{L\} \), the switches are nonconducting in the resulting state, and, hence, the resulting value of \( y \) is \( \emptyset \). If \( \gamma \cdot z = \{H\} \), both switches are conducting in the resulting state, and, as a result, the value and the mutillation degree for \( y \) in the resulting state equal \( \gamma \cdot x \) and the initial mutillation degree for \( x \). Consequently, the circuit can be seen as a perfect switch (provided that \( z \) and \( \bar{z} \) are synchronised).

\[ \begin{aligned}
\text{figure 6.5a: example 6.33a} \\
\text{figure 6.5b: example 6.33b}
\end{aligned} \]

(\text{end remark 6.32})

6.34 Multiple Threshold Voltages

In section 6.0 the assumption is made that all threshold voltages are equal. As a result, the additional mutillation caused by each transistor is uniform (and set to 1; cf. def 6.11b (mp)). Studying different threshold voltages is useful, since n-transistors and p-transistors need not have equal threshold voltages.

Multiple threshold voltages can be modelled by using values in \( \mathbb{Z} \) as the mutillation caused by one switch. In the model this changes the definitions of MD, \( \oplus \) and mp as follows:

\[ \text{MD} = \{i : (\mathbb{Z} \cdot i \wedge 0 \leq \text{SW} \cdot \text{MD}\{\alpha\}) \lor i = \alpha : i \} \]

Let the threshold voltage function be given by \( \text{thr} \in \text{SW} \rightarrow \text{MD}\{\alpha\} \). For a switch \( s \), \( \text{thr}\cdot s \) denotes the (abstract) threshold voltage of \( s \). For \( i \in \text{MD}\{\alpha\} \) function \( i \oplus \) is defined by:

\[ i \oplus \alpha = m \quad \text{and, for } j \in \text{MD}\{\alpha\} : \quad i \oplus j = (i \oplus j) \min m \]

In the definition of the mutillating power (6.11b (mp) and 6.23 (mp1)) the '1\oplus' must be replaced by 'thr-s \oplus'. The remaining part of the formalisation need not be altered.

(\text{end remark 6.34})
6.4 Concluding remarks on chapter 6

Due to the imperfectness of transistors, signals passing a transistor may be mutilated with the threshold voltage of the transistor. One of the design rules used in circuit design gives a maximal amount of mutilation due to threshold voltages that is allowed for a signal in order to be able to control a transistor gate. In the model given in this chapter this amount of mutilation is formalised in the abstract notion of mutilation degree. The design rule then leads to a correctness criterion $ML(u)$ requiring that the mutilation degree at gate nodes is at most $u$.

The model presented provides a general formalisation of the effects of imperfectness of switches in the sense that with one model, choosing an appropriate value for $m$ (the maximal degree of mutilation possible), all correctness criteria $ML(u)$ with $u < m$ can be verified. This is in contrast to existing formalisations that capture imperfectness of switches (like [BrzY]), where for every design rule, i.e. for every $u$, a different model is required. The basic idea of this abstract mutilation degree can, however, also be used in these other models, leading to the obvious improvement of generality.

The formalisation of mutilation degree provides a nice separation of concerns, since the new correctness criterion can (but need not) be calculated separately from the correctness criteria formalised in the previous chapters.

The mutilation degree in a state can be calculated efficiently (cf. theorem 6.21), as a result of which the verification of the new correctness criterion (section 6.2) can also be done efficiently. In section 6.3 we demonstrated that the model can very simply be generalised to allow mutilated input signals and multiple threshold voltages.
CHAPTER 7  THE RELATION BETWEEN
MODEL AND SPECIFICATIONS

In this chapter, the relation between specified behaviour and modelled behaviour is discussed. The main intention is to discuss the kind of correctness criteria that are necessary for circuits in order to meet their specifications, and to demonstrate that these criteria can be expressed in our model. We will not present, or even use, a formal specification language; they can be found elsewhere and would, at this place, distract from the main goals. Specifications of circuits are treated informally using an ad hoc notation presented in the first section. In the second section three new correctness criteria are introduced that relate modelled behaviour to specified behaviour. These criteria are illustrated by means of a few simple examples.

7.0  Specifications of Circuits

In this section an informal notation for specifications is introduced, which should not be regarded as an attempt to introduce a formal specification language. Formal specification languages for circuits can be found in [E1], [Ma1], using a CSP-like notation to specify the communication behaviour, or in [HP], using state transition graphs, or in [HP] and [WE], using Karnaugh maps.

Let \( C = \langle N,SW, t,g,pa \rangle \) be a circuit. In order to describe the specified behaviour of \( C \), we partition node set \( N \) into the (disjoint) subsets \( in \), \( out \), \( pls \), \( phs \), and \( local \), which are interpreted as follows:

\[
\begin{align*}
\text{in} & \quad \text{as the set of input nodes}, \\
\text{out} & \quad \text{as the set of output nodes}, \\
\text{pls} & \quad \text{as the set of nodes that are permanently connected to an L-source}, \\
\text{phs} & \quad \text{as the set of nodes that are permanently connected to an H-source}, \\
\text{local} & \quad \text{as the set of internal nodes that are not directly connected to a source}.
\end{align*}
\]
The set of global states of the circuit is defined as: \[ \text{GLOBAL} = (\text{in} \cup \text{out}) \rightarrow \{\{L\}, \{H\}\} \].

A specification gives a relation between specified inputs and specified outputs. The input is specified by an input function \( \sigma \in \text{in} \rightarrow \{\{L\}, \{H\}\} \).

The source-connection corresponding to an input function \( \sigma \) is called \( \gamma(\sigma) \) and is defined as:

\[
\gamma(\sigma) = \begin{cases} 
\emptyset, & \text{if } x \in \text{out} \cup \text{local} \\
\{L\}, & \text{if } x \in \text{pls} \\
\{H\}, & \text{if } x \in \text{pfs} \\
\sigma \cdot x, & \text{if } x \in \text{in}
\end{cases}
\]

The specified output depends on the specified input and possibly some global state or a history of states. For convenience, we mention only the dependence on a global state (interpreted as the previous state), and an input function (interpreted as the current input). For a global state \( G \) and an input function \( \sigma \), the specified output function is given by \( \text{sof}(G, \sigma) \in \text{out} \rightarrow \{\{L\}, \{H\}\} \).

If \( \sigma \) is an initialising input function, the specified output function depends only on \( \sigma \), in which case it is denoted as \( \text{sof}(\sigma) \).

Usually, only a few input functions are permitted as initialising input functions, and only a few input functions are permitted when the circuit is in a certain global state (giving the allowed input changes). The permitted input functions and the specified output functions are given by global state transition graphs, which are informally introduced in the following example.

### 7.0 Example

In the circuit depicted in figure 7.0a (from ex. 2.15, 4.38, 6.30), the node set is divided as follows: \( \text{in} = \{x, y\} \), \( \text{out} = \{z_2\} \), \( \text{local} = \{z_0, z_1\} \), and \( \text{pls} \) and \( \text{pfs} \) are the sets of nodes labelled with \( L \) and \( H \) respectively.

The global state transition graph is depicted in figure 7.0b. The global states are given by the values of \( x, y \), and \( z_2 \) (in that order). The value of \( x \) and \( y \) gives the permitted input function and the value of \( z_2 \) gives the specified output function. The initialising input functions are denoted by the ingoing fat arrows. Single and double arrows have the obvious meaning. For instance, the top state \( G = (H \ H \ H) \) gives the permitted initialising input function \( \sigma_0 = \{(x, \{H\}), (y, \{H\})\} \) and defines \( \text{sof}(\sigma_0) = \{(z_2, \{H\})\} \).

Input function \( \sigma_0 \) is permitted in the states \((H \ L \ H)\) and \((H \ L \ L)\), and, consequently, gives an input change on \( y \).

\[ \text{figure 7.0a: circuit ex. 7.0} \]

\[ \text{figure 7.0b: state transition graph ex. 7.0} \]
The specification can also include assumptions on the environment about when input changes will be sent to the circuit. The two most important ones for asynchronous circuits are fundamental mode and input-output mode (section 0.0). The difference for the interpretation of the global state transition graph for these two modes is explained in the next section.

7.1 Correctness Criteria w.r.t. Specifications

In this section three new correctness criteria are defined that relate modelled behaviour to specified behaviour. First, initialising input functions are considered, which correspond to initial behaviour. Then, the input functions starting in a global state, which correspond to dynamic behaviour, are considered. The new correctness criteria for initial and dynamic behaviour are combined with the old ones, leading to complete criteria.

Correctness Criteria for Initial Behaviour

Let \( C = (N, SW, t, g, pn) \) be a circuit, where \( N = in \cup out \cup pls \cup phs \cup local \) is the partition of \( N \) described in the previous section. Let \( \sigma \) be an initialising input function, and let the source-connection be \( \gamma = \gamma(\sigma) \). The correctness criterion for initial behaviour we have presented in the previous chapters, with \( u \) as the ultimate mutilation degree given by the design rule described in chapter 6 (section 6.3), is (cf. remark 6.27):

(10) \[ WM0-C \cdot \gamma_\mathcal{C} \land CF \cdot \gamma_\mathcal{A} \land ML(u)-md_\gamma \]

Besides this criterion it is, obviously, necessary to require that the value at the output nodes in the resulting state is as specified by the function sof(\( \sigma \)). This is expressed in the following criterion (where 'OAS' stands for 'output as specified'), using \( \gamma_\mathcal{S} = (Q_\mathcal{S}, \Gamma_\mathcal{S}) \):

(11) \[ OAS(\sigma) \cdot out \cdot \gamma_\mathcal{S} = (A \cdot x : x \in out : \Gamma_\mathcal{S} \cdot x = sof(\sigma) \cdot x) \]

A second criterion requires that the value at the output nodes is not mutilated. This may be useful if the outgoing signals are ingoing signals to other circuitry, possibly using another design rule, and, consequently, using another ultimate mutilation value \( u \). Furthermore, the assumption of non-mutilated inputs does, in view of composability, logically lead to the requirement of non-mutilated outputs. The criterion is defined below and is called \( PO \), which stands for 'perfect outputs'. The criterion may be weakened by replacing '= 0' by '\( \leq k \)' for some \( k \leq u \), and, hence, requiring that the output signals have mutilation degree at most \( k \)

(12) \[ PO \cdot out \cdot md_\gamma = (A \cdot x : x \in out : md_\gamma \cdot x = 0) \]

The initial behaviour of a circuit is correct if (10), (11), and (12) hold for every initialising input function. The correctness criteria are discussed further later on in this section, in combination with the ones for dynamic behaviour. They are illustrated in the following example.
7.1 Example
Consider again the flip-flop depicted in figure 7.0a, with the state transition graph from figure 7.0b. For the initialising input function \( \sigma_0 \) (ex. 7.0), the resulting state \( \gamma(\sigma_0) \) has values \( \{H\}, \{H\}, \{H\}, \{L\}, \{H\} \) in nodes \( x, y, z_p, z_1, z_2 \) (in that order) with mutation degree \( 0, 0, 1, 0, \) and \( 0 \) respectively (cf. example 6.30, state 115). As in example 2.15, we can conclude \( W M 0-C \gamma(\sigma_0) \). Consequently, using \( u = 1 \) and (from fig. 7.0b) \( sof(\sigma_0) - z_2 = \{H\} \), the correctness criteria for initial behaviour for \( \sigma_0 \) are satisfied.

Considering circuit states only (cf. ex. 4.38), nodes \( z_0 \) and \( z_2 \) seem to be equivalent (both good candidates as output nodes). The mutation degree, however, shows that they are not. Notice that the output node, \( z_2 \), is non-mutated, and, consequently, \( P O \cdot md \cdot \{z_2\} \) is satisfied, whereas internal node \( z_0 \) is mutated. The internal mutation (in \( z_0 \)) can be avoided by using pass-transistors instead of \( z_0 \) and \( z_1 \) (cf. example 6.33b).

Correctness Criteria for Dynamic Behaviour
Let \( C = \langle N, SW, t, g, pn \rangle \) be a circuit, where \( N = in \cup out \cup pls \cup rhs \cup local \) is the partition of \( N \) described in the previous section. Let \( \sigma \) be an input function starting in global state \( G \), and let the source connection be \( \gamma = \gamma(\sigma) \). For now, we consider the dynamic behaviour of \( C \) for a starting circuit state \( ((I \ll rc, pc), p) \) (cf. sect. 6.2). Which circuit states can be starting states in a given global state \( G \) and an input function \( \sigma \), is explained later on.

For correctness of dynamic behaviour we have argued that, for \( D \subseteq DC \times DC \) as the set of delay restrictions, and \( u \) as before, \( WF D[0,C \gamma ((I \ll rc, pc), p)] \) must be satisfied (cf. sect. 6.2), which means (cf. def. 6.29):

\[
D(0) \quad (A, rd, pd : (rd, pd) \in D)
\]

\[
\quad : RSL_{A \cdot rd \cdot pd} \cdot \gamma((I \ll rc, pc), p) \subseteq (cghdL \cup CFL \cup FCAI \cup cslL) \times MLL(u)
\]

According to the remark below def. 6.29, using def. 5.23, this equals:

\[
D(1) \quad (A, rd, pd : (rd, pd) \in D)
\]

\[
\quad : RSL_{A \cdot rd \cdot pd} \cdot \gamma((I \ll rc, pc), p) \subseteq (cghdL \cup CFL \cup FCAI \cup cslL)
\]

\[
\wedge (A, \Psi, M : (rd, pd) \in D \wedge RSL_{A \cdot rd \cdot pd} \cdot \gamma((I \ll rc, pc), p) - \cdot \Psi, M : M I I . u - M)
\]

That is, the resulting state lists satisfy criteria \( cghdL, CFL, FCAI, \) and \( cslL, \) and the mutation at gate nodes in the elements of these lists does not exceed \( u \).

According to lemma 5.26b, a resulting state list, say \( \Psi \), has, when criterion (D0) is satisfied, a suffix consisting of a repeated feasible state list, i.e.

\[
(\exists \Phi_0, \Phi_1 : \exists(St1) - \Phi_0 \wedge FSL_{A \cdot rd \cdot pd} \cdot \gamma - \Phi_1 : \Psi = cat \cdot \Phi_0 \cdot \ast (\Phi_1) \).
\]

The elements of these suffixes are considered to be the resulting states. The set of resulting states \( R S_{0, \gamma} ((I \ll rc, pc), \) is, therefore, defined as the smallest set containing them.
Obviously, the output values in these resulting states must be as specified by \( \text{sof}(G,o) \). This is expressed in the criterion (D2) below.

(D2) \[ RSL_{\text{d}}^{\text{\gamma}}(\Pi,\text{rc,pc}) \subseteq \text{OAS}(G,o)-\text{out} \] , where \( \text{OAS} \) is extended by:

\[ \text{OAS}(G,o)-\text{out}(Q_{\Gamma}) = (A \cdot x : x \in \text{out} : \Gamma \cdot x = \text{sof}(G,o) \cdot x) \]

As before, we require that the values at the output nodes are non-mutilated. This is required not only for the resulting states, but also for the intermediate states.

For \( M \in \mathcal{L}(\text{MUT}) \), we define:

(D3) \[ \text{POL-out} \cdot M = (A : i : \text{PO-out} \cdot M_{i}) \] , where \( \text{PO} \) is extended as follows:

\[ \text{PO-out-md} = (A \cdot x : x \in \text{out} : \text{md} \cdot x = 0) \]

The described criterion is expressed by (D4), and can be nicely combined with (D0) (cf. (D7)).

(D4) \[ (A : \text{rd,ps,\Psi}) : M, (\text{rd,ps}) : D \land RSL_{\text{d}}^{\text{\gamma}}(\Pi,\text{rc,pc})\cdot (\Psi, M) : \text{POL-out} \cdot M \]

The next correctness criterion for dynamic behaviour requires that the output values change smoothly to the specified values. This means that, for every output node \( x \), in each resulting state \( Q_{\Gamma} \), the value at the output node starts with the starting value \( \Gamma_{0} \cdot x \), then possibly has, during some steps, the value \( \text{store} \cdot \Gamma_{0} \cdot x \), after which it changes to the final value \( \text{sof}(G,o) \cdot x \). This is also required if the output value does not change, i.e., if \( \Gamma_{0} \cdot x = \text{sof}(G,o) \cdot x \).

Before discussing this criterion further, we define this kind of output behaviour for a state list \( \Psi \in \mathcal{L}(\text{ST1}) \) in the notion \( \text{SOB} \) (D5 below), which stands for 'smooth output behaviour'.

(D5) \[ \text{SOB}(G,o)-\text{out} \cdot \Psi \]

\[ = (A \cdot x \cdot Q_{0}, \Gamma_{0} : x \in \text{out} \land (Q_{0}, \Gamma_{0}) = \Psi_{0} \]

\[ \land (E : j \cdot k : 0 \leq i \leq k : (A \cdot i, Q_{i}, \Gamma_{i} : 0 \leq i \land \Psi_{i} = (Q_{i}, \Gamma_{i}) \land (0 \leq i \leq j) \Rightarrow \Gamma_{i} \cdot x = \Gamma_{0} \cdot x) \]

\[ \land (j < i \leq k \Rightarrow \Gamma_{i} \cdot x = \text{store} \cdot \Gamma_{0} \cdot x) \]

\[ \land (k < i \Rightarrow \Gamma_{i} \cdot x = \text{sof}(G,o) \cdot x)) \]

The described criterion can now be expressed as:

(D6) \[ (A : \text{rd,ps}) : (\text{rd,ps}) : D \land RSL_{\text{d}}^{\text{\gamma}}(\Pi,\text{rc,pc}) \subseteq \text{SOB}(G,o)-\text{out} \]

It is easily verified that the latter criterion (D6) implies the criterion given in (D2) (the proof is left to the reader). Furthermore, criterion (D6) implies the absence of hazardous behaviour ([HP]) at the output nodes, which is particularly important for asynchronous circuits.

The correctness criteria for dynamic behaviour, with starting state \( (\Pi,\text{rc,pc}) \cdot p \), i.e., (D0) (or (D1)), (D2), (D4), and (D6), can now be combined, and are concisely expressed in (D7) below.

(D7) \[ (A : \text{rd,ps}) : (\text{rd,ps}) : D \land RSL_{\text{d}}^{\text{\gamma}}(\Pi,\text{rc,pc}) \cdot p \]

\[ \subseteq (\text{csgdL} \land \text{CFL} \land \text{FCAI} \land \text{cstL} \land \text{SOB}(G,o)-\text{out} \land (\text{MLL}(u) \land \text{POL-out} \}) \]
Starting States for Dynamic Behaviour

The remaining question for dynamic behaviour is which circuit states can be starting states in a given global state and a given input function.

If the previous phase is an initialisation phase with source-connection $\gamma$, the only possible starting state clearly is $((\gamma_a,0,0) , \text{md}_a)$. If the previous phase is a dynamic transition, the possible starting states depend on the timing assumption on the environment. The most important timing assumptions for asynchronous circuits are fundamental mode and input output mode (cf. [BrezB]).

Informally, fundamental mode means that the environment will not send an input change until the circuit has reached internally some sort of stability. In our model, this means that the input change can be sent as soon as the circuit is in one of the states of the suffix of a resulting state list (see above). This means that the circuit states that can be starting states are the elements of $\langle R\phi, \gamma \rangle$ (Ill, lc, pc), where $\gamma$ is the previous source-connection and (Ill, lc, pc) is the previous starting state.

Input output mode means that the environment of the circuit can send a next input change as soon as the specified outputs are observed. In our model this means that an input change can arrive as soon as the elements of the set out are equal to their specified values. Consequently, the possible starting states for the next phase are all elements of resulting state lists that satisfy $OAS(G, o) = \text{out}$, where $G$ and $o$ are the current global state and the current input function.

Obviously, the current delay counters and the mutilation degree (from the definition of resulting state lists type 4) must be part of the starting state.

Notice that if (D7) is satisfied, the set of starting states for fundamental mode is included in the set of starting states for input-output mode (cf. (D2)). This precisely expresses the fact that fundamental mode is a stronger assumption on the environment than input-output mode.

7.2 Example

Consider again the circuit from example 7.0. In example 4.38 we explained that the circuit is correct if pass-delays and wire-delays are zero, and the reaction-delays are restricted as described in ex. 4.38 (these are not the weakest possible delay restrictions).

The case with global state (L L L) and input function $((x,(H)),(y,(L)))$, where the previous global state is (L H L), is described in example 4.38c. Notice that the possible starting states for this case are all states that are depicted in figure 4.3c except for the top one.

However, if the bottom feedback wire in figure 7.0a contains a wire-delay (or, as in example 5.27, a conducting switch with a pass delay), the circuit is not correct under the input-output mode assumption, but it is correct under the fundamental mode assumption (cf. example 5.27).
Summary and discussion of the complexity of the resulting criteria

For correctness of a circuit C with respect to its specification, (I0) \& (I1) \& (I2) must be satisfied for all initialising input functions. Furthermore, for all global states and allowed input changes, and all possible starting states, (D7) must be satisfied.

As observed in chapter 1, the evaluation of (I0) is, in general, rather complex. In chapter 2 we have seen that the evaluation of (I0) can be done efficiently if C is acyclic w.r.t. the initial source-connection. Since (I1) and (I2) can be evaluated efficiently, and, hence, (I0) forms the only problem for initial behaviour, it seems a good design philosophy to take care that the circuit is acyclic w.r.t. all initialising input functions. In example 2.15 we have seen that determining whether a circuit is acyclic w.r.t. a source-connection can be very simple. How to determine this efficiently, using a general procedure, is still an open problem.

The evaluation of the correctness criterion for dynamic behaviour described above seems, in general, quite complex due to the universal quantifiers in the criterion. The quantification over global states and allowed input changes depends totally on the specification. For each single starting state the complexity of (D7) depends on the set of delay restrictions D. The main cause of complexity is the number of starting states. In the examples throughout the thesis, we have seen that the feasible state list S is in the suffix *{S} of the resulting state list usually contains only one element. That means that the circuit ends up in a stable state (which is not a general property: cf. ex. 4.40). In those cases the evaluation of the criterion for dynamic behaviour can be done relatively efficient, particularly for the fundamental mode assumption.

By means of a simple example, the specifications and correctness criteria are illustrated.

7.3 Example

Consider the specification in figure 7.1a, and circuit C in figure 7.1b.

The partition of N is \( \text{in} = \{x\} \), \( \text{out} = \{y\} \), \( \text{local} = \emptyset \), and \( \text{phs} \) and \( \text{pls} \) are the singletons containing the node labelled with H and L respectively.

The two initialising input functions with the corresponding source-connections and the specified output functions are:

\[ \sigma_0 = \{(x,H)\} \quad \text{with} \quad \gamma(\sigma_0) = \{(x,H),(y,\emptyset)\} \quad \text{and} \quad \text{sof}(\sigma_0) = \{(y,L)\} \]

\[ \sigma_1 = \{(x,L)\} \quad \text{with} \quad \gamma(\sigma_1) = \{(x,L),(y,\emptyset)\} \quad \text{and} \quad \text{sof}(\sigma_1) = \{(y,H)\} \]

The depicted circuit is acyclic (AD0-C, cf. ch.2). Consequently, WMO-C-\( \gamma \) follows from \( cgd\gamma \). (th. 2.14). As a result, the criterion for initial correctness is:

\[ (A : i \in \{0,1\} : \gamma(\sigmai)_* \in (cgd\gamma \cap CF \cap OAS(\sigmai) \cap \{y\}) \quad \text{and} \quad \text{md}_{\gamma(\sigmai)} \in (ML(u) \cap PO \{y\}) ) \]

This criterion is easily verified, for all \( u \in \text{MD}_a(\alpha) \), using:

\[ \gamma(\sigma_0)_* = \{(s_0,0),(s_1,0)\}, \{(x,H),(y,L)\} \quad \gamma(\sigma_1)_* = \{(s_0,0),(s_1,1)\}, \{(x,L),(y,H)\} \]

and \( \text{md}_{\gamma(\sigmai)} \colon X = 0 \), for all \( i \in \{0,1\} \), \( \varepsilon \in \text{N} \), and \( X \in (L,H) \).
From example 5.24 we conclude that for global state (H L) the only possible starting circuit state is \((\gamma(\sigma_0), 0, 0), md(\gamma(\sigma_0))_o\) (see figure 5.4a). Similarly, for global state (L H), the only possible starting circuit state is \((\gamma(\sigma_l), 0, 0), md(\gamma(\sigma_l))_o\).

In order for \(WF_{D_C} \gamma(\sigma_i)_o \{(\gamma(\sigma_o), 0, 0), md(\gamma(\sigma_o))_o\} \) to be satisfied, where \(\{i, j\} = \{0, 1\}\), the delay restrictions (expressed in D) must satisfy (cf. ex.5.24): \(rd_{L_i} \leq rd_{H_j}\).

Using figure 5.4a it is easily verified that the other criteria for dynamic behaviour (in D7) are also satisfied.

\[
\begin{array}{c}
\text{H} \\
\downarrow \\
\text{L} \\
\downarrow \\
\text{L}
\end{array}
\quad \begin{array}{c}
\text{H} \\
\downarrow \\
\text{L} \\
\downarrow \\
\text{L}
\end{array}
\]

figure 7.1a: specification

figure 7.1b: circuit example 7.3

7.2 Concluding Remarks on chapter 7

In this chapter, we discussed the kind of correctness criteria for initial behaviour of circuits as well as for dynamic behaviour of circuits that are necessary to relate modelled behaviour to specified behaviour. In order to formalise these criteria, we introduced an informal notation for specifications of circuits. In this notation, specifications are given by a partition of the node set and a global state transition diagram. This diagram defines the permitted initialising input functions, the permitted input changes in global states, and the specified output functions.

With the help of this informal notation for specifications, we demonstrated that the discussed criteria can be expressed in our model. Combined with the familiar criteria, they lead to complete correctness criteria for initial behaviour and for dynamic behaviour, which are defined in section 7.1, and are illustrated with some simple examples. The criteria include absence of hazardous behaviour at output nodes, and the requirement of non mutilated output values. The influence of the timing assumptions on the environment, in particular fundamental mode and input-output mode, on the criteria is discussed, as well as the complexity of the complete criteria.
CHAPTER 8 CONCLUDING REMARKS

In the final sections of chapters 1 to 7, the main results obtained in those chapters, and the formalisation used, are summarised, discussed, and compared to earlier results. At a number of places in the previous chapters, topics for further research are discussed (cf. chapter 0 and sections 4.3, 5.1.2, and 7.1). In this chapter, some global remarks are made regarding the model presented in this thesis (section 8.0), and a number of topics for further research are addressed (section 8.1).

8.0 Remarks on the model

Making the assumptions and abstractions (a) – (e), as formulated in section 0.3, enabled a structured approach with respect to the modelling of circuit behaviour. In the basic model it turned out to be possible, on account of restrictions (a), (d), and (e), to use a simple notion of circuit states. Restrictions (b) and (c) allowed, in this basic model, a simple definition of the next-state function, which is the essential notion for circuit behaviour. While weakening the restrictions and refining the model, every aspect of switch-level circuits could be analysed and modelled in isolation. This approach led to a family of hierarchically ordered switch-level models, with the advantage that the consequences of the various circuit aspects could be investigated by comparing the models. For a summary and discussion of the formalisation of these aspects, and the results obtained for the corresponding models, we refer to the final sections of chapters 1 to 6.

Basically, circuit behaviour is defined with the help of the two consistency notions for switches, viz. consistent (def. 1.12, redefined in 4.5) and consistent (def. 1.15, redefined in 4.16). The correctness criteria for circuit behaviour fall into two categories, which can be called 'internal criteria', consisting of requirements due to aspects of the underlying physics or aspects of the model, and 'external criteria', consisting of the requirements that are necessary to relate modelled behaviour to specified behaviour.
The correctness criteria in the first category are:

- The requirement that in intermediate and resulting states the switches have either a driven low or high voltage value at the gate, or are temporarily controlled by a stored voltage value. This requirement is expressed in \( cgdo, cgdi, PCA0, \) and \( PCA1 \);

- The requirement for absence of conflicts (short circuits), expressed in \( CF \);

- The requirement for correctness of state transitions. This requirement results from the obvious abstractions made in discrete switch level modelling, viz. from a 'switching period' to a 'switching point action', and from 'continuous voltage values' to 'discrete states of nodes' (cf. sections 1.4 and 5.0.3). It is expressed in \( csu0 \) and \( csui \);

- The requirement that the mutilation degree at gates is at most \( u \), where the value \( u \) is given by a design rule (which may be technology dependent, cf. chapter 6). This requirement is expressed in \( Ml(u) \).

These requirements are partly combined in the notions well-matchedness, for initial behaviour of circuits, and well-functioning, for dynamic behaviour of circuits. The main result with respect to initial behaviour of circuits is that the effects of delays and charge storage need not be taken into account for the evaluation of the correctness criterion for initial behaviour. Furthermore, in case of correctness of initial behaviour, there is—abstracted from stored values—only one resulting state, which then is, of course, stable.

The criteria that express the relation between modelled behaviour and specified behaviour are defined in chapter 7. They include the requirement that output values in resulting states are as specified (expressed in \( OAS \)) and perfect, that is, free of mutilation (expressed in \( POL \)), and the requirement that the transitions at output nodes are smooth (expressed in \( SOR \), and including the absence of hazardous behaviour). The latter requirement means that a transition at an output node goes from the starting value, via, possibly, temporarily the stored charge corresponding to this value, directly to the specified value. The total correctness criteria for initial and dynamic behaviour of circuits are discussed in chapter 7.

It is important to note that, since our model is pessimistic with respect to the model reality relation (is conservative), the modelled behaviour should be interpreted as a description of physical behaviour only if this modelled behaviour is classified as correct according to the model.

Since every design (method) differs in the utilization of circuit aspects, it is also important to note that the model has enough expressive power to allow generalizations and variations on the model and on the correctness criteria in a simple way, cf. remark 4.35 (on transient conflicts), remarks 3.15 and 5.19 (on variations in formalising delays), section 6.2 (the design rule mentioned above), section 4.3 (multiple capacitance-strengths), remark 6.32 (on allowing mutilated input signals), and remark 6.34 (on allowing multiple threshold voltages).
8.1 Topics for further research

Throughout the thesis we pointed out subjects for further research. In this section, some of these subjects are addressed, and some additional remarks are made.

In section 5.1.2, we demonstrated that pass-delays cannot, in general, be modelled as part of reaction-delays. The transformation described in this section, from pass-delays to reaction-delays, appears, however, to be correct in many cases. Since calculating with positive pass-delays is much more complicated than calculating with reaction-delays, the investigation of these cases is worthwhile. We have the impression that a partition of the node set as described in chapter 7, with explicit permanent sources, will help to formulate these cases.

In chapter 0, we addressed a number of topics that need further attention:
- The relation between our model and lower-level (continuous) models, for instance w.r.t. the interpretation of the various notions of delays we introduced.
- The translation of switch-level notions to the layout-level, in particular the translation of restrictions that guarantee correctness of behaviour according to the switch-level model, like restrictions on delays and capacitance-strengths, to layout-level parameters.
- The relation to higher-level models (gate-level as well as switch-level) needs to be investigated in order to verify or correct the higher-level models using a more detailed model, like the one presented here, as a semantic domain. A further advantage of a formal comparison of switch-level models is that results that can be proved easily in some models, can possibly, on account of this comparison, be used in others. Finally, a formal comparison of models can lead to an exchange of formal techniques, like
  - The use of countdown functions to describe delays (cf. chapters 3 and 5), thereby avoiding transient cycles as in [BrzS1], and enabling a simple formalisation of restrictions on delays (although using orders of events rather than functions in SW → N might, for calculational purposes, be more efficient);
  - The notion of mutilation degree to describe the effects of imperfectness of switches (cf. chapter 6).
- The development of a design method based on the insights gained by the model. For instance, as discussed in sections 2.2 and 7.1, it seems a good design philosophy to take care that the circuit is acyclic w.r.t. all initialising input functions. The results of the investigation of the cases that pass-delays can be modelled as reaction-delays, as mentioned above, can be taken into account. Depending on the design method and technology used, further investigation of some of the extensions of the model we suggested, like the modelling of resistances described in section 4.3, will be necessary.
- The design of a switch-level simulator based on the model presented in this thesis.
APPENDIX A  Overview Lattice Theory

In this appendix some definitions and results from lattice theory are given. Proofs are not
included. For a more complete overview of lattice theory we refer to [Bl] or [DP].

Closures of relations

Definition  \{(reflexive and) transitive closure\}

Let \( R \) be a binary relation on set \( X \).

The transitive closure of \( R \) is the smallest relation \( R^* \) satisfying:

\( (A \ x \ y : R \times y : R^* \times y ) \)

\( (A \ x , y , z : R^* \times y \wedge R^* \times z : R^* \times z ) \)

The reflexive and transitive closure of \( R \) is the smallest relation \( R^* \) satisfying:

\( (A \ x : x \in X : R^* \times x ) \)

\( (A \ x , y , z : R^* \times y \wedge R^* \times z : R^* \times z ) \)

Notice that \( R^* \) satisfies \( (A \ x , y : R \times y : R^* \times y ) \wedge (A \ x , y , z : R^* \times y \wedge R^* \times z : R^* \times z ) \).

Theorem A0

Let \( R_0 \) and \( R_1 \) be binary relations on set \( X \), let \( R_0^* \) and \( R_1^* \) be the reflexive and
transitive closures of \( R_0 \) and \( R_1 \) respectively, and let \( P \) be a predicate on \( X \) such that

\( (A \ x , y : x \in X \wedge y \in X : P \times x \wedge R_0 \times y \Rightarrow R_1 \times x \times y ) \).

Let \( z \) be an element of \( X \). Then:

\( (A \ y : R_0^* \times y : P \times y ) \wedge (A \ y : R_1^* \times y : R_1^* \times z \times y ) \).

The proof of this theorem is simple using induction to the closure of \( R_0 \).

Orders and posets

Definition  \{partial order, poset\}

A partial order (or order for short) on set \( Y \) is a binary relation \( \leq \) on \( Y \) such that,
for all \( (x,y,z) \subseteq Y \):

\( x \leq x \quad \text{(\( \leq \) is reflexive)} \)

\( x \leq y \wedge y \leq x \Leftrightarrow x = y \quad \text{(\( \leq \) is anti-symmetric)} \)

\( x \leq y \wedge y \leq z \Rightarrow x \leq z \quad \text{(\( \leq \) is transitive)} \)

The pair \( (Y,\leq) \) is called a partially ordered set, or poset.
The relation \( \geq \) (called the dual order of \( \leq \)) is defined for a poset \((Y, \leq)\) by: \((x \geq y) = (y \leq x)\).
The relation \(<\) is defined for a poset \((Y, \leq)\) by: \((x < y) = (x \leq y \land x \neq y)\). Similarly for \(>\).

**Definition** \{linear order, chain\}

Let \((Y, \leq)\) be a poset.
The order \(\leq\) is called linear if \((\forall x, y : x \leq y \land y \leq x)\).
A subset \(X\) of \(Y\) is called a chain if the order \(\leq\) is linear on \(X\).

**Definition** \{Hasse diagram\}

A partial order \(\leq\) on a set \(Y\) can be depicted in a Hasse diagram (or diagram) as follows:
All elements of \(Y\) are depicted in the diagram in such a way that every two elements \(x\) and \(y\) satisfying \((x < y \land \neg(\exists z : z \leq Y : x < z \land z < y))\) are connected by a line, and \(y\) is depicted above \(x\).
Notice that the lines in the diagram give a binary relation on \(X\) of which \(\leq\) is the reflexive and transitive closure. If \(\leq\) is a linear order, the relation given by the lines is the successor relation.

**Definition** \{componentwise order and pointwise order\}

Let \((Y_i, \preceq)\) be posets, for all \(i, 0 \leq i \leq n\). The componentwise order \(\preceq\) on the cartesian product \(Y_0 \times \ldots \times Y_n\) is defined by \((\forall i : 0 \leq i \leq n)(x_i \preceq y_i)\).
Let \((Y, \preceq)\) be a poset and let \(X\) be a set. The pointwise order \(\preceq\) on the set of functions \(X \to Y\) is defined by \((f \preceq g) = (\forall x : x \in X : f(x) \preceq g(x))\).

**Definition** \{monotonicity and anti-monotonicity of functions\}

Let \((X, \preceq)\) and \((Y, \preceq)\) be posets, and let \(f : X \to Y\).
Function \(f\) is called monotonic (or order-preserving) w.r.t. \(\preceq\) if \((\forall x, y : x \preceq y \land y \preceq x)\).
Function \(f\) is called anti-monotonic w.r.t. \(\preceq\) if \((\forall x, y : x \preceq y \land y \preceq x)\).
Notice that function \(f\) is anti-monotonic w.r.t. \(\preceq\) if \(f\) is monotonic w.r.t. \(\preceq\).
Notice also that if \((X, \preceq) = (Y, \preceq)\) and \(f\) is anti-monotonic, then \(f^2\) is monotonic.

**Definition** \{maximal element, minimal element, greatest element, least element\}

Let \((Y, \leq)\) be a poset and let \(X\) be a subset of \(Y\). Then
\(\alpha \in X\) is a maximal element of \(X\) if \((\forall x : x \in X : \alpha \leq x \Rightarrow \alpha = x)\).
\(\alpha \in X\) is the greatest (or maximum) element of \(X\) if \((\forall x : x \in X : \alpha \geq x)\).
A minimal element of \(X\) and the least (or minimum) element of \(X\) are defined dually, that is, by reversing the order.
The greatest element of a poset \((Y, \preceq)\), if it exists, is called the top element (or top) of \(Y\) and is often denoted as \(\top\). The least element of a poset \((Y, \preceq)\), if it exists, is called the bottom element (or bottom) of \(Y\) and is often denoted as \(\bot\).

**Theorem A1**

Let \((Y, \preceq)\) be a non-empty finite poset. The following three propositions are equivalent.

1. \(Y\) has a least element.
2. \((\exists z \in Y : (\forall x \in Y : z \preceq x))\)
3. \((\forall x, y \in Y : x \preceq Y \land y \in Y : (\exists z \in Y : z \preceq x \land z \preceq y))\)

**Definition** (upper bound, lower bound, supremum, infimum, lub, glb)

Let \((Y, \preceq)\) be a poset and let \(X\) be a subset of \(Y\). Then

- \(a \in Y\) is called an upper bound of \(X\) if \((\forall x \in X : x \preceq a)\).
- If the set of upper bounds of \(X\) has a least element, this element is called the least upper bound (or supremum) of \(X\), and is often denoted as \(\operatorname{lub} X\) (or \(\operatorname{sup} X\)).
- A lower bound of \(X\), the greatest lower bound (or infimum) of \(X\), \(\operatorname{glb} X\), and \(\operatorname{inf} X\) are defined dually, that is, by reversing the order.

For the order \(\preceq\), the supremum of a two-element-set \(\{x, y\}\) if often written as \(\operatorname{join}(x, y)\) or \(x \lor y\) instead of \(\operatorname{sup}\{x, y\}\). Similarly, we write \(\operatorname{meet}(x, y)\) or \(x \land y\) instead of \(\operatorname{inf}\{x, y\}\).

**Lattices and fixedpoint theorems**

**Definition** (lattice, complete lattice)

Let \((Y, \preceq)\) be a non-empty poset.

- If \(x \cup y\) and \(x \cap y\) exist for all \(\{x, y\} \subseteq Y\), then \((Y, \preceq)\) is called a lattice.
- If \(\operatorname{sup} X\) and \(\operatorname{inf} X\) exist for all \(X \subseteq Y\), then \((Y, \preceq)\) is called a complete lattice.

Notice that every finite lattice is complete.

**Definition** (fixed point)

Let \(f : X \rightarrow X\). An element \(x \in X\) is called a fixed point (or fixed point) of \(f\) if \(f(x) = x\).

The set of fixed points of \(f\) is denoted as \(\operatorname{fix} f\).

**Definition** (continuous function)

Let \(L_0\) and \(L_1\) be complete lattices. A function \(f : L_0 \rightarrow L_1\) is called continuous if \(f(\operatorname{sup}_{L_0} C) = \operatorname{sup}_{L_1} (f(C))\) for every non-empty chain \(C \subseteq L_0\).

**Property**

Every continuous function is monotonic.
Theorem A2  (the fixpoint theorem of Knaster-Tarski)

Let \((L, \sqsubseteq)\) be a complete lattice and let \(f \in L - L\) be a monotonic function.

Then \((\text{fix } f, \sqsubseteq)\) is a non-empty complete lattice, and the least fixpoint \(\text{lfp } f\) of \(f\) satisfies:
\[ (A \ x : x \in L : f \ x \sqsubseteq x \Rightarrow \text{lfp } f \sqsubseteq x) . \]

Notice that if \(\text{fix } f\) is a complete lattice, it has a least and greatest element, and, consequently, \(f\) has a least fixpoint (denoted as \(\text{lfp } f\)) and a greatest fixpoint (denoted as \(\text{gfp } f\)).

Theorem A3  (the fixpoint theorem for continuous functions)

Let \((L, \sqsubseteq)\) be a complete lattice with least element \(\bot\) and greatest element \(\top\).

Let \(f\) be a continuous function. Then
\[
\text{lfp } f = \sup \{ i : i \in \mathbb{N} : f^i \bot \}
\]
\[
\text{gfp } f = \inf \{ i : i \in \mathbb{N} : f^i \top \}
\]

Notice that, since \(f\) is monotonic (use the property above), for all \(i \in \mathbb{N}\):
\[
f^{i+1} \bot \leq f^{i+1} \bot \text{ and } f^{i+1} \top \supseteq f^{i+1} \top.
\]

Notice also that if \(L\) is finite, \(\text{lfp } f\) and \(\text{gfp } f\) can be calculated in a finite number of steps. The suggested approximation process of \(\text{lfp } f\) (that is \(\bot \leq f \bot \leq f^2 \bot \leq \ldots\)) is called successive approximation from below. Similarly, the process for calculation of \(\text{gfp } f\) is called successive approximation from above.
APPENDIX B  PROOFS

In this appendix, the proofs of theorems 1.28 (appendix B0), 3.8 (appendix B1), and 5.10 (appendix B2) are given. These theorems express one of the main results regarding initial behaviour of circuits, and are formulated as follows, for \( i \in \{0,1,3\} \):

\[
(A \subset \gamma \cap CIR-C \wedge NST-\gamma : WMI-C-\gamma = (stablei-\gamma \subseteq cgd))
\]

The three proofs have a similar construction, with as a result that parts of the first proof(s) are used in the following one(s).

In appendices B1 and B2, two lemmas, B1 and B2, are given, which use the same construction in their proofs. These lemmas are used in the proofs of theorems 3.10 and 5.12 respectively.
APPENDIX B0  Proof of theorem 1.28

In this appendix the proof of theorem 1.28 is given. For completeness, theorem 1.28 is:

\[(A, c, y : CIR-C \land NST-y : WM0-C-y = (stable0-y \subseteq cgdo) )\]

Let CIR-C \land NST-y. On account of lemma 1.27: WM0-C-y = (stable0-y \subseteq cgdo).

Since (see def.1.25 (WM0)) WM0-C-y = (feasible0-y \subseteq cgdo), it suffices to prove:

\[-0. \quad (feasible0-y \subseteq cgdo) \Rightarrow (stable0-y \subseteq cgdo)\]

This will be proved by constructing, given a state \(II\) satisfying feasible0-y-\(\Omega\) \land \neg cgdo-\(\Omega\), a state \(\Omega\) satisfying stable0-y-\(\Omega\) \land \neg cgdo-\(\Omega\).

Let \(i, \{i : 0 < i : (Q_i, \Gamma_i)\}\) satisfy:

\[(1 \leq m) \land \text{feasible0-}\gamma(Q_0, \Gamma_0) \land \neg cgdo-\{Q_0, \Gamma_0\} \land ((Q_m, \Gamma_m) = (Q_0, \Gamma_0))\]

\[\land \quad (A \land : 0 < i : (Q_i, \Gamma_i) \land next \theta-\gamma(Q_i, \Gamma_i) = (Q_{i+1}, \Gamma_{i+1}) )\]

Notice that on account of def. 1.23 (feasible0) for each \((Q_0, \Gamma_0)\) satisfying feasible0-y \((Q_0, \Gamma_0)\)
\land \neg cgdo \((Q_0, \Gamma_0)\) such \(m\) and \(\{i : 0 < i : (Q_i, \Gamma_i)\}\) exist.

We start with a number of definitions.

\[\Gamma_{\text{max}} \in \text{NST}, \{Rt,E,Rb\} \subseteq N \rightarrow B, \{\text{bcpE}, \text{bcpE}\} \subseteq \text{SST} \rightarrow B2N \rightarrow B, \Delta \in \text{NST}, \text{ and } Q/l \in \text{SST}\]

are defined by:

\[\Gamma_{\text{max}} = \sup(i : 0 < i : \text{cm} : \Gamma_i)\]

\[\text{Rt-x} = (E i, y : 0 < i : \text{cm} \land N \cdot y : \Gamma_{\text{max}} \cdot y = (L, H) \land \text{cp-Q}_i(x, y) )\]

\[\text{E-x} = (E i, y : 0 < i : \text{cm} : \Gamma_i \cdot x = \emptyset )\]

\[\text{bcpE-Q-x} = (\text{bcp-Q}_i \cdot x) \land \neg \text{Rt-x} \land \neg \text{E-x} \land \neg \text{Rt-y} \land \neg \text{E-y} \land \text{cpE-Q}_l(x, y) )\]

\[\text{cpE-Q} \text{ is the reflexive and transitive closure of } \text{bcpE-Q}.\]

\[\text{Rb-x} = (\neg \text{Rt-x} \land (A \land : 0 < i : \text{cm} \land N \cdot y : \gamma - y = \emptyset \lor \neg \text{cpE-Q}_l(x, y) )\]

\[\Delta-x = \begin{cases} \{L, H\} & \text{if Rt-x} \\ \{L\} & \text{if } \neg \text{Rt-x} \land \neg \text{Rb-x} \land \Gamma_{\text{max}}-x = \{L\} \\ \{H\} & \text{if } \neg \text{Rt-x} \land \neg \text{Rb-x} \land \Gamma_{\text{max}}-x = \{H\} \\ \emptyset & \text{if Rb-x} \end{cases}\]

\[Q/l = \begin{cases} t \cdot s = \Delta (g \cdot s) & \text{if } gdb-\Delta-x \\ \neg (E x : x \in pn-x : \Delta-x = \emptyset ) & \text{if } \neg gdb-\Delta-x \end{cases}\]
Correctness of the definition of $\Delta$ follows from $(A x :: \neg R^{t} x \lor \neg R^{b} x)$, which follows from the definition of $Rb$, and (1b) below using the definition of $\Gamma_{\#x}$.

In the remainder of this proof the following properties are successively proved:

For all $x, y \in N$ and $s \in SW$:

(1a) \hspace{1em} (A i :: 0 \leq s < cm : y \in \Gamma_i)

(1b) \hspace{1em} \neg R^{t} x \land \neg R^{b} x \Rightarrow (A i :: 0 \leq s < cm : \Gamma_i x = \{L\}) \lor (A i :: 0 \leq s < cm : \Gamma_i x = \{H\})

(1c) \hspace{1em} \Delta x = \{L\} \Rightarrow (A i :: 0 \leq s < cm : \Gamma_i x = \{L\})

\hspace{1em} \Delta x = \{H\} \Rightarrow (A i :: 0 \leq s < cm : \Gamma_i x = \{H\})

(1d) \hspace{1em} gdb\Delta x = (A i :: 0 \leq s < cm : gdb\Gamma_i x \land Q_i x = Q_l x)

(1e) \hspace{1em} \Delta x \neq \emptyset \land \Delta y \neq \emptyset \Rightarrow (E i :: 0 \leq s < cm : bcp\{Q_i x, y\}) \Rightarrow bcp\{Q_l x, y\}

(1f) \hspace{1em} \Delta x \neq \emptyset \land \Delta y \neq \emptyset \land (E i :: 0 \leq s < cm : bcp\{Q_i x, y\}) \Rightarrow bcp\{Q_l x, y\}

(2a) \hspace{1em} (A x :: \Delta x = \emptyset : R^{-\gamma} Q_l x = \emptyset)

(2b) \hspace{1em} (A x :: \Delta x = \{L, H\} : R^{-\gamma} Q_l x = \{L, H\})

(2c) \hspace{1em} (A x :: \Delta x = \{L\} : L \in R^{-\gamma} Q_l x)

(2d) \hspace{1em} (A x :: \Delta x = \{H\} : H \in R^{-\gamma} Q_l x)

(3a) \hspace{1em} ccono\Delta Q_l

(3b) \hspace{1em} (A x :: N x : (\Delta x = \emptyset) = (R^{-\gamma} Q_l x = \emptyset))

(3c) \hspace{1em} \Delta \in R^{-\gamma} Q_l

(4a) \hspace{1em} stable\gamma\{Q_l, R^{-\gamma} Q_l\}

(4b) \hspace{1em} \neg ccono\{Q_l, R^{-\gamma} Q_l\}

Notice that with (4a) and (4b) the proof of $\neg 0$- is complete.

PROOFS
Property (1a) follows directly from the definition of $\Gamma_i (\Gamma_i x = R^{-\gamma} Q_l)$ and property 1.1.8a.

Proof of (1b)
Since $(A i :: \Gamma_i x \in \mathcal{P}(\{L, H\})$ property (1b) can be proved in the following two steps:

(1b.1) \hspace{1em} \neg R^{t} x \Rightarrow \neg (E i :: 0 \leq s < cm : (\Gamma_i x \cup \Gamma_i x) = \{L, H\})

(1b.2) \hspace{1em} \neg R^{b} x \land \neg R^{b} x \Rightarrow \neg (E i :: 0 \leq s < cm : \Gamma_i x = \emptyset)$
Proof of (1b.1)
(B i : 0 ≤ i ≤ m : (Γ_i; x ∪ Γ_{i+1}; x) = {L, H})
⇒ [def. \( \Gamma_{max} \)]
\( \Gamma_{max}; x = \{L, H\} \)
⇒ [reflexivity of cp-Q, for all Q (from def. 1.14 (cp))]
(B i : 0 ≤ i ≤ m : (Γ_{max}; x = \{L, H\}) ∧ cp-Q_i; [x, x])
⇒ [renaming]
(B i, y : 0 ≤ i ≤ m ∧ N·y : (Γ_{max}; y = \{L, H\}) ∧ cp-Q_i; [x, y])
⇒ [def. Rt]
Rt; x

Proof of (1b.2)
Notice that:
(¬Rt · x ∧ ¬Rb · x ⇒ ¬E · x) = (¬Rt · x ∧ E · x ⇒ Rb · x). Then:
¬Rt · x ∧ E · x
⇒ [def. E, def. cpE, and (1a)]
¬Rt · x ∧ γ · x = Ø ∧ (A, Q · y : cpE-Q · [x, y]; y = x)
⇒ [calculus]
¬Rt · x ∧ (A, Q · y : cpE-Q · [x, y]; γ · y = Ø)
⇒ [calculus]
¬Rt · x ∧ (A, i, y : 0 ≤ i ≤ m ∧ N·y : γ · y = Ø ∨ ¬cpE-Q_i · [x, y])
⇒ [def. Rb]
Rb · x

Proof of (1c)
(1c) is a direct result of (1b) and the definiton of \( \Gamma_{max} \).

Proof of (1d)
gd₀ · Δ · s
⇒ [def. 1.11 (gd₀), (1c)]
gd₀ · Δ · s ∧ (A i : 0 ≤ i ≤ m : gd₀ · Γ_{i+1} · s ∧ (Γ_i · (g · s) = Δ · (g · s)))
⇒ [def. Q! · def. 1.12 (consistent₀), and note 1 below]
(A i : 0 ≤ i ≤ m : gd₀ · Γ_{i+1} · s ∧ (Q_i · s = Q! · s))

Note 1
true
⇒ [assumption]
(A i : 0 ≤ i < m : next₀ · f · (Q_i · Γ_i) · (Q_{i+1} · Γ_{i+1}))
⇒ [def. 1.21 (next₀)]
(A i : 0 ≤ i < m : consistent₀ · Γ_i · Q_{i+1})
Appendix B0  Proof of Theorem 1.28

Proof of (1c)

\[ \Delta x \neq \emptyset \land \Delta y \neq \emptyset \land (E_i : 0 \leq i < m : \text{bcp} \cdot Q_i \cdot [x, y]) \]

\[ = \{ \text{def. 1.14 (bcp)} \} \]

\[ \Delta x \neq \emptyset \land \Delta y \neq \emptyset \land (E_{s,i} : 0 \leq i < m \land \text{pn} \cdot s = [x, y] : Q_i \cdot s) \]

\[ \vdash \{ \text{calculus} \} \]

\[ (E_s : \text{pn} \cdot s = [x, y] : (gdb \cdot \Delta x \land (E_i : 0 \leq i < m : Q_i \cdot s))) \]

\[ \lor (\neg gdb \cdot \Delta y \land \neg (E_z : z \in \text{pn} \cdot s : \Delta x = \emptyset )) \]

\[ \vdash \{ (1d) \text{ and def. } QI \} \]

\[ (E_s : \text{pn} \cdot s = [x, y] : QI \cdot s) \]

\[ = \{ \text{def. 1.14 (bcp)} \} \]

\[ \text{bcp} \cdot QI \cdot [x, y] \]

Proof of (1f)

\[ \Delta x \neq \emptyset \land \Delta y \neq \emptyset \land (E_i : 0 \leq i < m : \text{bcpB} \cdot Q_i \cdot [x, y]) \]

\[ \vdash \{ \text{from def. bcpB} : (A_i : 0 \leq i < m : \text{bcpB} \cdot Q_i \cdot [x, y] \Rightarrow \text{bcp} \cdot Q_i \cdot [x, y]) \} \]

\[ \Delta x \neq \emptyset \land \Delta y \neq \emptyset \land (E_i : 0 \leq i < m : \text{bcp} \cdot Q_i \cdot [x, y]) \]

\[ \vdash \{ (1e) \} \]

\[ \text{bcp} \cdot QI \cdot [x, y] \]

Proof of (2a)

We start with another property :

(2a.1) \[ (A, s, x, z : \text{pn} \cdot s = [x, z] \land \Delta x = \emptyset : \text{QI} \cdot s \Rightarrow \Delta z = \emptyset ) \]

Let \( s, x, z \) satisfy : \( \text{pn} \cdot s = [x, z] \land \Delta x = \emptyset \). Then :

\[ \text{QI} \cdot s \]

\[ = \{ \text{def. QI using } (E x : x \in \text{pn} \cdot s : \Delta x = \emptyset ) \} \]

\[ \text{QI} \cdot s \land gdb \cdot \Delta x \]

\[ \vdash \{ (1d) \} \]

\[ \text{QI} \cdot s \land (A_i : 0 \leq i < m : Q_i \cdot s = \text{QI} \cdot s) \]

\[ \vdash \{ \text{calculus} \} \]

\[ \text{QI} \cdot s \land (A_i : 0 \leq i < m : Q_i \cdot s) \]

\[ \vdash \{ \text{def. } \Gamma_1 \} \]

\[ (A_i : 0 \leq i < m : Q_i \cdot s) \land (A_i : 0 \leq i < m : \Gamma_1 \cdot s = \Gamma_1 \cdot z) \]

\[ \vdash \{ \text{def. } \text{E} \} \]

\[ (A_i : 0 \leq i < m : Q_i \cdot s) \land (E x = E z) \]

\[ \vdash \{ \text{pn} \cdot s = [x, z] \text{ and def. 1.14 (bcp)} \} \]

\[ (A_i : 0 \leq i < m : \text{bcp} \cdot Q_i \cdot [x, z]) \land (E x = E z) \]
\[
\begin{align*}
\Delta x &= \emptyset, \text{ note 2A below} \\
\neg Rt \cdot x &\land (A \ i: 0 \leq i < m \land bcp_{Q_1}[x, z]) \land (E \cdot x = E \cdot z) \\
\rightarrow &\{ \text{def. } Rt, \text{ transitivity } cp \cdot Q \text{ (from def. 1.14 (cp))} \}
\neg Rt \cdot x \land \neg Rt \cdot z \land (A \ i: 0 \leq i < m \land bcp_{Q_1}[x, z]) \land (E \cdot x = E \cdot z)
\rightarrow &\{ \text{def. bcpE and calculus} \}
\neg Rt \cdot z \land (A \ i: 0 \leq i < m \land bcpE_{Q_1}[x, z]) = (\neg E \cdot x \land \neg E \cdot z) \land (E \cdot x = E \cdot z)
\rightarrow &\{ \text{calculus} \}
\neg Rt \cdot z \land (A \ i: 0 \leq i < m \land bcpE_{Q_1}[x, z]) = \neg E \cdot z
\rightarrow &\{ \text{calculus} \}
\neg Rt \cdot z \land (A \ i: 0 \leq i < m \land E \cdot z \lor bcpE_{Q_1}[x, z])
\rightarrow &\{ \text{calculus} \}
\neg Rt \cdot z \land (E \cdot z \lor (A \ i: 0 \leq i < m \land bcpE_{Q_1}[x, z]))
\rightarrow &\{ \text{def. cpE, and } \Delta x = \emptyset, \text{ note 2A} \}
\neg Rt \cdot z \land (A \ i, y: 0 \leq i < m \land N \cdot y: \gamma \cdot y = \emptyset \lor \neg cpE_{Q_1}[x, y])
\rightarrow &\{ \text{calculus} \}
\neg Rt \cdot z \land (A \ i, y: 0 \leq i < m \land N \cdot y: \gamma \cdot y = \emptyset \lor \neg cpE_{Q_1}[x, y])
\rightarrow &\{ \text{note 2B below} \}
\neg Rt \cdot z \land (A \ i, y: 0 \leq i < m \land N \cdot y: \gamma \cdot y = \emptyset \lor \neg cpE_{Q_1}[x, y])
\rightarrow &\{ \text{note 2A} \}
\Delta z = \emptyset
\end{align*}
\]

**Note 2A**

\[
\Delta u = \emptyset
\]

\[
\Rightarrow &\{ \text{def. } \Delta \}
\]

\[
\neg Rb \cdot u
\]

\[
\Rightarrow &\{ \text{def. } Rb \}
\neg R\Delta \cdot u \land (A \ i, y: 0 \leq i < m \land N \cdot y: \gamma \cdot y = \emptyset \lor \neg cpE_{Q_1}[u, y])
\]

**Note 2B**

\[
E \cdot z \land \neg Rt \cdot z
\]

\[
\Rightarrow &\{(1a), \text{ def. cpE} \}
\gamma \cdot z = \emptyset \land (A \ i, y: z \neq y: \neg cpE_{Q_1}[z, y])
\]

\[
\Rightarrow &\{ \text{calculus} \}
(A \ i, y: 0 \leq i < m \land N \cdot y: \gamma \cdot y = \emptyset \lor \neg cpE_{Q_1}[x, y])
\]

We are now able to prove (2a):

true

\[
\Rightarrow &\{(2a.1)\}
(A \ x: \Delta x = \emptyset: (A \ s, z: pn \cdot s = [x, z] \land Q \cdot s: \Delta \cdot z = \emptyset))
\]

\[
\Rightarrow &\{ \text{def. 1.14 (bcp)} \}
(A \ x: \Delta x = \emptyset: (A \ z: bcp \cdot Q \cdot l \cdot [x, z]: \Delta \cdot z = \emptyset))
\]
\begin{verbatim}
= \{ induction \}
   \( (A \ x : \Delta x = \emptyset : (A \ z : \text{cp-}\mathcal{Q}^I_{\{x,z\}} : \Delta z = \emptyset) ) \)
\rightarrow \{ from\ reflexivity \ \text{cp-}\mathcal{B}^I \ \text{and\ def.} \ \Delta \ \text{and} \ \text{def.} \ \gamma \ \Rightarrow \ \gamma z = \emptyset \}
   \( (A \ x : \Delta x = \emptyset : (A \ z : \text{cp-}\mathcal{Q}^I_{\{x,z\}} : \gamma z = \emptyset) ) \)
= \{ calculus \}
   \( (A \ x : \Delta x = \emptyset : (\cup z : \text{cp-}\mathcal{Q}^I_{\{x,z\}} : \gamma z ) = \emptyset) ) \)
\rightarrow \{ def. \ R \ \gamma \mathcal{Q}^I \ (1.17) \}
   \( (A \ x : \Delta x = \emptyset : R \ \gamma \mathcal{Q}^I \ x = \emptyset) ) \)
\end{verbatim}

\( \Box \) (end of proof (2a))

Proof of (2b)
\[ \Delta x = \{ L, H \} \]
= \{ def. \ \Delta \}
   \( R \mathcal{T} x \)
= \{ def. \ R \}
   \( (B \ i, y : 0 \leq i < m \ \land \ \Gamma_{\max}^y = \{ L, H \} : \text{cp-}\mathcal{Q}^I_{\{x,y\}} ) \)
\rightarrow \{ def. \ \Gamma_{\max} \ \text{and} \ \text{calculus} \}
   \( (B \ y : \Gamma_{\max}^y = \{ L, H \} : (B \ i : 0 \leq i < m : \text{cp-}\mathcal{Q}^I_{\{x,y\}} ) \)
\land \( (B \ i : 0 \leq i < m : L \in \Gamma_1^y ) \)
\land \( (B \ i : 0 \leq i < m : H \in \Gamma_1^y ) ) \)
\rightarrow \{ def. \ \Gamma_1 \}
   \( (B \ y : \Gamma_{\max}^y = \{ L, H \} : (B \ i : 0 \leq i < m : \text{cp-}\mathcal{Q}^I_{\{x,y\}} ) \)
\land \( (B \ z_0 : L \in \gamma_{z_0} \ \land \ (B \ i : 0 \leq i < m : \text{cp-}\mathcal{Q}^I_{\{z_0,y\}} ) ) \)
\land \( (B \ z_1 : H \in \gamma_{z_1} \ : (B \ i : 0 \leq i < m : \text{cp-}\mathcal{Q}^I_{\{z_1,y\}} ) ) \)
= \{ calculus \}
   \( (B \ y, z_0, z_1 : \Gamma_{\max}^y = \{ L, H \} \ \land \ \Lambda \in \gamma_{z_0} \ \land \ \Lambda \in \gamma_{z_1} : (B \ i : \text{cp-}\mathcal{Q}^I_{\{x,y\}} ) \land (B \ i : \text{cp-}\mathcal{Q}^I_{\{z_0,y\}} ) \land (B \ i : \text{cp-}\mathcal{Q}^I_{\{z_1,y\}} ) ) \)
\rightarrow \{ note \ 2C \ below \}
   \( (B \ y, z_0, z_1 : L \in \gamma_{z_0} \ \land \ \Lambda \in \gamma_{z_1} : \text{cp-}\mathcal{Q}^I_{\{x,y\}} \land \text{cp-}\mathcal{Q}^I_{\{z_0,y\}} \land \text{cp-}\mathcal{Q}^I_{\{z_1,y\}} ) \)
\rightarrow \{ transitivity \ of \ \text{cp-}\mathcal{Q}^I \ (from \ def. \ 1.14 \ (cp)) \}
   \( (B \ z_0, z_1 : L \in \gamma_{z_0} \ \land \ \Lambda \in \gamma_{z_1} : \text{cp-}\mathcal{Q}^I_{\{x,y\}} \land \text{cp-}\mathcal{Q}^I_{\{z_0,y\}} ) \)
= \{ def. \ R \ \gamma \mathcal{Q}^I \ (1.17) \}
   \( R \ \gamma \mathcal{Q}^I \ x = \{ L, H \} \)
\end{verbatim}
Note 2C
\[ \Gamma_{\max} y = \{ L, H \} \]
\[ \vdash \{ \text{def. } R_t \text{ and calculus} \} \]
\[ (A x : (B i : 0 \leq i < m : \text{cp } Q_t[x,y]) : \text{Rt} \cdot x) \]
\[ \vdash \{ \text{def. } \Delta \} \]
\[ (A x : (B i : 0 \leq i < m : \text{cp } Q_t[x,y]) : \Delta \cdot x \neq \emptyset \) \]
\[ \vdash \{ \text{theorem A0 (appendix A), using (1e)} \} \]
\[ (A x : (B i : 0 \leq i < m : \text{cp } Q_t[x,y]) : \text{cp } Q^I \cdot [x,y]) \]

Proof of (2c)
\[ \Delta \cdot x = \{ L \} \]
\[ = \{ \text{def. } \Delta \} \]
\[ \Gamma_{\max} x = \{ L \} \land \neg R_t \cdot x \land \neg R_b \cdot x \]
\[ \vdash \{ \text{def. } R_b \text{ and calculus} \} \]
\[ \Gamma_{\max} x = \{ L \} \land (B i, y : 0 \leq i < m \land N_i \cdot y : \gamma \cdot y = \emptyset \land \text{cp } E \cdot Q_t[x,y]) \]
\[ = \{ \text{calculus} \} \]
\[ \Gamma_{\max} x = \{ L \} \land (B y : \gamma \cdot y = \emptyset \land (B i : 0 \leq i < m \land \text{cp } E \cdot Q_t[x,y])) \]
\[ \vdash \{ \text{definitions } \Gamma_t \text{ and } \Gamma_{\max}, \text{using } \text{cp } E \cdot Q_t \subseteq \text{cp } Q \text{ for all Q} \} \]
\[ (B y : \gamma \cdot y = \{ L \} \land (B i : 0 \leq i < m \land \text{cp } E \cdot Q_t[x,y])) \]
\[ \vdash \{ \text{note 2D below} \} \]
\[ (B y : \gamma \cdot y = \{ L \} : \text{cp } Q^I \cdot [x,y]) \]
\[ = \{ \text{def. } R \cdot \gamma \cdot Q^I (1.17) \} \]
\[ L \in R \cdot \gamma \cdot Q^I \cdot x \]

Note 2D
\[ \gamma \cdot y = \{ L \} \]
\[ \vdash \{ \text{calculus, def. } R_b \} \]
\[ (A i, x : \text{cp } E \cdot Q_t[x,y] : \neg R_b \cdot x) \]
\[ \vdash \{ \text{def. } \Delta \text{ and calculus} \} \]
\[ (A x : (B i : 0 \leq i < m : \text{cp } E \cdot Q_t[x,y]) : \Delta \cdot x \neq \emptyset \) \]
\[ \vdash \{ \text{theorem A0 (appendix A), using (1f)} \} \]
\[ (A x : (B i : 0 \leq i < m : \text{cp } E \cdot Q_t[x,y]) : \text{cp } Q^I \cdot [x,y]) \]

Proof of (2d)
The proof of (2d) is similar to the proof of (2c).
Proofs of (3a,b,c)
(3a) follows directly from the definitions of coco0 (1.12) and QI ;
(3b) follows directly from (2a,b,c,d) ; and (3c) follows directly from (2b,c,d) .

Proof of (4a)
true
= [(3b,c)]
\((A \, x : R \cdot \gamma \cdot QI \cdot x \in \{ [L], [H] \} : R \cdot \gamma \cdot QI \cdot x = \Delta \cdot x)\)
\Rightarrow [\text{calculus}]
\((A \, s : R \cdot \gamma \cdot QI \cdot (g \cdot s) \in \{ [L], [H] \} : R \cdot \gamma \cdot QI \cdot (g \cdot s) = \Delta \cdot (g \cdot s))\)
\Rightarrow [\text{def. 1.12 (coco0) and (3a)}]
coco0 \cdot (R \cdot \gamma \cdot QI) \cdot QI
= [\text{lemma 1.22c}]
stable0 \cdot \gamma \cdot (QI, R \cdot \gamma \cdot QI)

Proof of (4b)
true
= [assumption]
\neg \gamma \cdot d0 \cdot (Q0, \Gamma_0)
= [\text{def. 1.11 (cfd0)}]
\((A \, s : SW \cdot s : \neg \gamma \cdot d0 \cdot \Gamma_0 \cdot s)\)
\Rightarrow [(1d)]
\((E \, s : SW \cdot s : \neg \gamma \cdot d0 \cdot \Delta \cdot s)\)
\Rightarrow [\text{def. 1.11 (gfd0) and (2a,b)}]
\((E \, s : SW \cdot s : \neg \gamma \cdot d0 \cdot (R \cdot \gamma \cdot QI) \cdot s)\)
= [def. 1.11 (cgfd0)]
\neg \gamma \cdot d0 \cdot (QI, R \cdot \gamma \cdot QI)

END OF PROOF
APPENDIX B1 Proof of theorem 3.8

In this appendix the proof of theorem 3.8 is given. For completeness, theorem 3.8 is:

\[(A, C, \gamma : \text{CIR-C} \land \text{NST-}\gamma : \text{WM1-C-}\gamma = (\text{stable1-}\gamma \subseteq \text{cgdo})\)\]

Let \(\text{CIR-C} \land \text{NST-}\gamma\). On account of prop. 3.7 we have: \(\text{WM1-C-}\gamma = (\text{stable1-}\gamma \subseteq \text{cgdo})\).

Since (see def. 3.3 (WM1)) \(\text{WM1-C-}\gamma = (\text{feasible1-}\gamma \subseteq \text{cgdo})\), it suffices to prove:

1. \((\text{feasible1-}\gamma \subseteq \text{cgdo}) \Rightarrow (\text{stable1-}\gamma \subseteq \text{cgdo})\)

This will be proved by constructing, given a state \(\Pi\) satisfying \(\text{feasible1-}\gamma \cdot \Pi \land \neg \text{cgdo} \cdot \Pi\), a state \(\Omega\) satisfying \(\text{stable1-}\gamma \cdot \Omega \land \neg \text{cgdo} \cdot \Omega\).

The structure of this proof is very similar to the one in appendix B0, but in order to have more benefit from the proof, some little changes are made. First of all, these changes are the omission of the premise \(\neg \text{cgdo} \cdot (Q_0, \Gamma_0)\), which must — in order to be able to prove 1 — result in a change of property (4b). At the end of this appendix we will show what we gained.

Let \(m, \{i : 0 \leq i \leq m : ((Q_0 \cdot \Gamma), r c)\}\) satisfy:

\[\begin{align*}
1 \leq m \land \text{feasible1-}\gamma \cdot (Q_0, \Gamma_0) \land ((Q_m \cdot \Gamma_m), r c_m) = ((Q_0, \Gamma_0), r c_0) \\
\land (A, i : 0 \leq i \leq m : \text{nexti-}\gamma ((Q_i \cdot \Gamma_i), r c_i)) = ((Q_{i+1} \cdot \Gamma_{i+1}), r c_{i+1})
\end{align*}\]

Notice that on account of def. 3.3 (feasible) for each \((Q_0, \Gamma_0)\) satisfying \(\text{feasible1-}\gamma \cdot (Q_0, \Gamma_0)\) such \(m\) and \(\{i : 0 \leq i \leq m : ((Q_0 \cdot \Gamma_i), r c_i)\}\) exist.

The definitions and properties from appendix B0 can be used again (with a minor change in (4b)). The proofs of properties (1a-f), (2a-d), and (3a-c) require a change only of Note 1 (belonging to the proof of (1d)). The proofs of (4a,b) also require very little change. For convenience the definitions and properties are listed below. For the proofs of (1a-c), (1e,f), (2a-d), and (3a-c) we refer to appendix B0. The proofs of properties (1d) and (4a,b) are given below.
The definitions

\[ \Gamma_{\text{max}} \in \text{NST}, \ \{R,t,E,Rb\} \subseteq N \rightarrow B, \ \{\text{bcpE},\text{cpE}\} \subseteq \text{SST} \rightarrow B \rightarrow B, \ \Delta \in \text{NST}, \ \text{and} \ Q_I \in \text{SST} \]

are defined by:

\[
\begin{align*}
\Gamma_{\text{max}} &= \sup(i : 0\leq i < m : \Gamma_i) \\
R i x &= (E i y : 0\leq i < m \land N i y : \Gamma_{\text{max}} y = \{L,H\} \land \text{cpQ}_1[x,y]) \\
E i x &= (E i : 0\leq i < m : \Gamma_i x = \emptyset) \\
\text{bcpE} Q [x,y] &= (\text{bcpQ}_1[x,y] \land (\neg E x \lor R t x) \land (\neg E y \lor R t y)) \\
\text{cpE} Q &= \text{the reflexive and transitive closure of bcpE} Q, \\
R b x &= (\neg R t x \land (A i y : 0\leq i < m \land N i y : \gamma y = \emptyset \lor \neg \text{cpE} Q[x,y])) \\
\Delta x &= \begin{cases} 
[L,H] & \text{if } R t x \\
[L] & \text{if } \neg R t x \land \neg R b x \land \Gamma_{\text{max}} x = \{L\} \\
[H] & \text{if } \neg R t x \land \neg R b x \land \Gamma_{\text{max}} x = \{H\} \\
\emptyset & \text{if } R b x 
\end{cases} \\
\text{Q}_I x &= \begin{cases} 
t s = \Delta x (g s) & \text{if } g d o \Delta s \\
(\neg (E x : x \in \text{po} s : \Delta x = \emptyset) & \text{if } \neg g d o \Delta s 
\end{cases}
\end{align*}
\]

Correctness of the definition of \( \Delta \) follows from \( \{A x : \neg R t x \lor \neg R b x\} \), which follows from the definition of \( R b \) and (1b) using the definition of \( \Gamma_{\text{max}} \) below.

The properties

For all \( x,y \in N \) and \( s \in \text{SW} \):

1a. \( \{A i : 0\leq i < m : \gamma y \subseteq \Gamma_i\} \)

1b. \( \neg R t x \land \neg R b x \Rightarrow (A i : 0\leq i < m : \Gamma_i x = \{L\}) \lor (A i : 0\leq i < m : \Gamma_i x = \{H\}) \)

1c. \( \Delta x = \{L\} \Rightarrow (A i : 0\leq i < m : \Gamma_i x = \{L\}) \)

1d. \( \Delta x = \{H\} \Rightarrow (A i : 0\leq i < m : \Gamma_i x = \{H\}) \)

1e. \( g d o \Delta s \Rightarrow (A i : 0\leq i < m : g d o \Gamma_i s \land Q_I s = Q_I s) \)

1f. \( \Delta x \neq \emptyset \land \Delta y \neq \emptyset \land (E i : 0\leq i < m : \text{bcpQ}_1[x,y]) \Rightarrow \text{bcpQ}_I[x,y] \)

2a. \( \{A x : \Delta x = \emptyset : R_I Q_I x = \emptyset\} \)

2b. \( \{A x : \Delta x = \{L,H\} : R_I Q_I x = \{L,H\}\} \)

2c. \( \{A x : \Delta x = \{L\} : L \in R_I Q_I x\} \)

2d. \( \{A x : \Delta x = \{H\} : H \in R_I Q_I x\} \)

3a. \( \text{cocone}\Delta Q_I \)

3b. \( \{A x : N x : (\Delta x = \emptyset) = (R_I Q_I x = \emptyset)\} \)

3c. \( \Delta \subseteq R_I Q_I \)
\[(4a) \quad \text{stable}_1 \cdot (Q_1, R \cdot (Q_1))\]
\[(4b) \quad \neg g \cdot (Q_0, \Gamma_0) \implies \neg g \cdot (Q_1, R \cdot (Q_1))\]

Notice that with (4a) and (4b) the proof of -1 is complete.

PROOFS

As mentioned above we refer to appendix B0 for the proofs of (1a-c), (1e-f), (2a-d), and (3a-c).

Proof of (1d)

Let \( s \in SW \). Then:
\[ gd \cdot \Delta \cdot s \]
\[ = \{ \text{def. 1.11 (gd\(\cdot\)), (1c)} \} \]
\[ \quad gd \cdot \Delta \cdot s \land \ (A \ i : 0 \leq i < m : gd \cdot \Gamma_1 \cdot s \land (G_{i}(g \cdot s) = \Delta(g \cdot s)) \} \]
\[ \implies \{ \text{def. } Q1 , \text{ def. 1.12 (consistent), and note 1 below} \} \]
\[ \quad (A \ i : 0 \leq i < m : gd \cdot \Gamma_1 \cdot s \land (Q_i \cdot s = Q_1 \cdot s)) \]

Note 1

Assume: \( (A \ i : 0 \leq i < m : gd \cdot \Gamma_1 \cdot s \land (G_{i}(g \cdot s) = \Delta(g \cdot s)) \) . Then:
\[ \text{true} \]
\[ = \{ \text{note 1.0} \} \]
\[ \quad (E \ i : 0 \leq i < m : rc_i \cdot s = 0) \]
\[ \implies \{ \text{induction of note 1.1, using } ((Q_m, \Gamma_m), rc_m) = ((Q_0, \Gamma_0), rc_0) \} \]
\[ \quad (A \ i : 0 \leq i < m : \text{consistent}_0 \cdot \Gamma_1 \cdot Q_i \cdot s) \]

Note 1.0

\[ (A \ i : 0 \leq i < m : rc_1 \cdot s > 0) \]
\[ \implies \{ rc_0 = rc_m \} \]
\[ (A \ i : 0 \leq i < m : rc_1 \cdot s > 0) \]
\[ \implies \{ \text{R2.0, R2.1 from ch.3} \} \]
\[ (A \ i : 0 \leq i < m : rc_{i+1} \cdot s = rc_i \cdot s - 1) \]
\[ \implies \{ \text{induction} \} \]
\[ (A \ i : 0 \leq i < m : rc_i \cdot s = rc_0 \cdot s - i) \]
\[ \implies \{ rc_0 = rc_m \} \]
\[ rc_0 \cdot s = rc_0 \cdot s - m \]
\[ \implies \{ 1 \leq m \} \]
false
APPENDIX B1 PROOF OF THEOREM 3.8

**Note 1.1**

Let \( \mathcal{O}_i \subset \mathcal{M} \). Assume : \((A_j : O \mathcal{O}_j \subset \mathcal{M} : g \mathcal{G} \mathcal{O}_j \Gamma_j \mathcal{S} \wedge (\Gamma_j \mathcal{G} \mathcal{S}) = \Delta \mathcal{S})\). Then :

\[ r_j = 0 \]

\[ \Rightarrow \{ \text{R}0.0 \text{ from ch.3} \} \]

\[ \text{consistent} \mathcal{G} \mathcal{O}_j \Gamma_j \mathcal{S} \]

\[ = \{ \text{from assumption} : \Gamma_j \mathcal{G} \mathcal{S} = \Gamma_j \mathcal{G} \mathcal{S} \}, \text{and def. 1.12 (consistent)} \]

\[ \text{consistent} \mathcal{G} \mathcal{O}_j \Gamma_j \mathcal{S} \]

\[ \Rightarrow \{ \text{R}2.0 \text{ from ch.3} \} \]

\[ \text{consistent} \mathcal{G} \mathcal{O}_j \Gamma_j \mathcal{S} \wedge (r_{i+1} \mathcal{S} = 0) \]

\( \square \) (end of proof (1d))

**Proof of (4a)**

true

\[ = \{(3b,c)\} \]

\[ (A \mathcal{x} : R \gamma \mathcal{Q} \mathcal{I} \mathcal{x} \in \{ [L],[H] \} : R \gamma \mathcal{Q} \mathcal{I} \mathcal{x} = \Delta \mathcal{x}) \]

\[ \Rightarrow \{ \text{calculus} \} \]

\[ (A \mathcal{x} : R \gamma \mathcal{Q} \mathcal{I} \mathcal{x} \in \{ [L],[H] \} : R \gamma \mathcal{Q} \mathcal{I} \mathcal{x} = \Delta \mathcal{x}) \]

\[ \Rightarrow \{ \text{def. 1.15 (coco00) and (3a)} \} \]

\[ \text{coco00} \mathcal{R} \gamma \mathcal{Q} \mathcal{I} \mathcal{L} \]

\[ = \{ \text{prop. 1.22c} \} \]

\[ \text{stable0} \gamma \mathcal{Q} \mathcal{I} \gamma \mathcal{Q} \mathcal{I} \]

\[ = \{ \text{lemma 3.5} \} \]

\[ \text{stable1} \gamma \mathcal{Q} \mathcal{I} \gamma \mathcal{Q} \mathcal{I} \]

**Proof of (4b)**

\[ \neg \neg \neg \mathcal{G} \mathcal{O}_0 \mathcal{G} \mathcal{O}_0 \]

\[ = \{ \text{def. 1.11 (cgdo)} \} \]

\[ (B \mathcal{x} : \neg \neg \neg \mathcal{G} \mathcal{O}_0 \mathcal{S}) \]

\[ \Rightarrow \{ (1d) \} \]

\[ (B \mathcal{x} : \neg \neg \neg \mathcal{G} \mathcal{O}_0 \Delta \mathcal{S}) \]

\[ \Rightarrow \{ \text{def. 1.11 (cgdo)} \text{ and (2a,b)} \} \]

\[ (B \mathcal{x} : \neg \neg \neg \mathcal{G} \mathcal{O}_0 \mathcal{S}) \]

\[ = \{ \text{def. 1.11 (cgdo)} \} \]

\[ \neg \neg \neg \mathcal{G} \mathcal{O}_0 \mathcal{G} \mathcal{Q} \mathcal{I} \mathcal{R} \gamma \mathcal{Q} \mathcal{I} \]

\( \square \) (end of proofs of properties)
Using the construction above it is possible to prove the following lemma, which can be used to prove theorem 3.10b.

\textbf{B1 Lemma}

\[(\text{stable} \cdot \gamma = \{\gamma_\star\} \land cgd0 \cdot \gamma_\star) \Rightarrow (\text{feasible} \cdot \gamma = \{\gamma_\star\})\]

\textbf{Proof}

Let \((Q_0, \Gamma_0)\) satisfy \(\text{feasible} \cdot \gamma(Q_0, \Gamma_0)\). The construction and properties above are used in the remaining of this proof.

\((\text{stable} \cdot \gamma = \{\gamma_\star\}) \land cgd0 \cdot \gamma_\star\)

\[\Rightarrow \{(4a)\}\]

\[= (Q_1, R \cdot \gamma \cdot Q_1) = \gamma_\star \land cgd0 \cdot \gamma_\star\]

\[= (\text{def. 1.11 (cgdo, gd0)})\]

\[= (Q_1, R \cdot \gamma \cdot Q_1) = \gamma_\star \land (A \cdot s : SW \cdot s : R \cdot \gamma \cdot Q_1 \cdot (g \cdot s) \in \{L\}, \{H\}) \quad (3b)\]

\[= (\text{def. 1.11 (gd0)})\]

\[= (Q_1, R \cdot \gamma \cdot Q_1) = \gamma_\star \land (A \cdot s : SW \cdot s : gd0 \cdot \Delta \cdot s) \quad (1d)\]

\[= (\text{calculus \text{ (Leibniz)} })\]

\[= (Q_1, R \cdot \gamma \cdot Q_1) = \gamma_\star \land (A \cdot i \cdot s : SW \cdot s : (0 \leq i < m) \cdot (Q_i = Q_1 \cdot s) \quad \text{from assumption of the construction: } \Gamma_0 = R \cdot \gamma \cdot Q_{m-1} \quad (Q_0, \Gamma_0) = \gamma_\star\]

\[\square\]
APPENDIX B2 Proof of theorem 5.10

In this appendix the proof of theorem 5.10 is given. For completeness, theorem 5.10 is:

\((A, C, \gamma; \text{CIR} \land \text{NST} \land \gamma; W M 3 \land \gamma = (\text{stable} \land \gamma \subseteq \text{cgdo})\)

Let \(\text{CIR} \land \text{NST} \land \gamma\). On account of lemma 5.9b: \(W M 3 \land \gamma = (\text{stable} \land \gamma \subseteq \text{cgdo})\).

Since \(W M 3 \land \gamma = (F S L 3 \land \gamma \subseteq (F C A 0 \land \text{cgdo})\), it suffices to prove:

\(-2: (F S L 3 \land \gamma \subseteq (F C A 0 \land \text{cgdo})\) \land \neg (\text{stable} \land \gamma \subseteq \text{cgdo})

This will be proved by constructing, given a state list \(\Psi\) satisfying \(F S L 3 \land \gamma \Psi \land \neg (F C A 0 \land \Psi \land \text{cgdo})\), a state \(\Omega\) satisfying \(\text{stable} \land \Omega \land \neg \text{cgdo} \land \Omega\).

Let \(F S L 3 \land \gamma \Psi\) and \(m = \# \Psi\). Let \(\{i : 0 \leq i \leq m : ((Q_i, A_i), r_{c_i}, p_{c_i})\}\) satisfy:

\((A, i : 0 \leq i \leq m : ((Q_i, A_i) = \Psi_i) \land \text{next} \land \gamma ((Q_i, A_i), r_{c_i}, p_{c_i}) = ((Q_{i+1}, A_{i+1}), r_{c_{i+1}}, p_{c_{i+1}}))\)

\(\land (((Q_0, A_0), r_{c_0}, p_{c_0}) = ((Q_0, A_0), r_{c_0}, p_{c_0}))\)

Notice that on account of def. 5.4 \((F S L 3)\) for each \(\Psi\) satisfying \(F S L 3 \land \gamma \Psi\) this \(m\) satisfies \(1 \leq m\), and such set \(\{i : 0 \leq i \leq m : ((Q_i, A_i), r_{c_i}, p_{c_i})\}\) exists.

Define, for \(0 \leq i \leq m\): \(\Gamma_i = \text{destore} \land \Lambda_i\).

Notice that as a result of def. 5.2 \((\text{next})\), lemma 4.20, and prop. 4.18c.d: \(\Gamma_1 = R \land (Q_1 \land Z \land p_c)\)

The structure of the proof given here is similar to the ones in appendices B0 and B1. As in appendix B1 an additional lemma (B2; corresponding to B1) is given which enables us to prove theorem 5.12b (corresponding to 3.10b). The definitions and properties from appendix B0 can be used again (with a minor change in (4b)); for convenience they are listed below. The proofs of properties (1b,c,e,f), (2b), and (3a,c) require no change; for them we refer to appendix B0. The proofs of (1a,d), (2a,c,d), and (4a,b) are given below. Some of these proofs are essentially more difficult than their counterparts in the previous appendices, e.g. the proofs of (2a,c). The proofs of (1a,d) and (4a) are similar to the corresponding proofs in appendix B1.
The definitions
\[ \Gamma_{\text{max}} \in \text{NST}, \ (Rt,E,Rb) \in N - \mathbb{B}, \ \{bcpE,cpE\} \subseteq \text{SST} - \mathbb{B} \cap \mathbb{B}, \ \Delta \in \text{NST}, \ \text{and} \ QI \in \text{SST} \]
are defined by:
\[ \Gamma_{\text{max}} = \sup(i : 0 \leq i \leq m : \Gamma_i) \]
\[ Rt \cdot x = (E \cdot i : y : 0 \leq i \leq m \wedge N \cdot y : \Gamma_{\text{max}} \cdot y = \{L,H\} \wedge cpE \cdot \{x,y\}) \]
\[ E \cdot x = (E \cdot i : 0 \leq i \leq m : \Gamma_i \cdot x = \emptyset) \]
\[ bcpE \cdot Q \cdot \{x,y\} = (bcpE \cdot \{x,y\} \wedge (\neg E \cdot x \vee Rt \cdot x) \wedge (\neg E \cdot y \vee Rt \cdot y)) \]
\[ cpE \cdot Q \text{ is the reflexive and transitive closure of } bcpE \cdot Q. \]
\[ Rb \cdot x = (\neg Rt \cdot x \wedge (A \cdot i : 0 \leq i \leq m \wedge N \cdot y : \forall \ y : \neg \emptyset \vee \neg cpE \cdot \{x,y\})) \]
\[ \Delta \cdot x = \begin{cases} \{L,H\}, & \text{if } Rt \cdot x \\ \{L\}, & \text{if } \neg Rt \cdot x \wedge \neg Rb \cdot x \wedge \Gamma_{\text{max}} \cdot x = \{L\} \\ \{H\}, & \text{if } \neg Rt \cdot x \wedge \neg Rb \cdot x \wedge \Gamma_{\text{max}} \cdot x = \{H\} \\ \emptyset, & \text{if } Rb \cdot x \end{cases} \]
\[ QI \cdot s = \begin{cases} t \cdot s = \Delta \cdot (g \cdot s), & \text{if } g \cdot t \cdot s \Delta \cdot s \\ \neg(t \cdot x \in \text{pn} \cdot s : \Delta \cdot s = \emptyset), & \text{if } \neg g \cdot t \cdot s \Delta \cdot s \end{cases} \]

Correctness of the definition of \( \Delta \) follows from \((A \cdot x : \neg Rt \cdot x \vee \neg Rb \cdot x)\), which follows from the definition of \( Rb \), and (1b) below using the definition of \( \Gamma_{\text{max}} \).

The properties
For all \( x,y \in N \) and \( s \in \text{SW} \):
\[ \text{(1a) } (A \cdot i : 0 \leq i \leq m : y \subseteq \Gamma_i) \]
\[ \text{(1b) } \neg Rt \cdot x \wedge \neg Rb \cdot x \Rightarrow (A \cdot i : 0 \leq i \leq m : \Gamma_i \cdot x = \{L\}) \vee (A \cdot i : 0 \leq i \leq m : \Gamma_i \cdot x = \{H\}) \]
\[ \text{(1c) } \Delta \cdot x = \{L\} \Rightarrow (A \cdot i : 0 \leq i \leq m : \Gamma_i \cdot x = \{L\}) \\ \Delta \cdot x = \{L\} \Rightarrow (A \cdot i : 0 \leq i \leq m : \Gamma_i \cdot x = \{H\}) \]
\[ \text{(1d) } g \cdot t \cdot s \Delta \cdot s \Rightarrow (A \cdot i : 0 \leq i \leq m : g \cdot t \cdot s \wedge Q \cdot s = QI \cdot s) \]
\[ \text{(1e) } \Delta \cdot x \neq \emptyset \wedge \Delta \cdot y \neq \emptyset \wedge (E \cdot i : 0 \leq i \leq m : bcpE \cdot \{x,y\}) \Rightarrow bcpE \cdot QI \cdot \{x,y\} \]
\[ \text{(1f) } \Delta \cdot x \neq \emptyset \wedge \Delta \cdot y \neq \emptyset \wedge (E \cdot i : 0 \leq i \leq m : \text{bcpE} \cdot \{x,y\}) \Rightarrow \text{bcpE} \cdot QI \cdot \{x,y\} \]
\[ \text{(2a) } (A \cdot x : \Delta \cdot x = \emptyset : R \cdot y \cdot QI \cdot x = \emptyset) \]
\[ \text{(2b) } (A \cdot x : \Delta \cdot x = \{L,H\} : R \cdot y \cdot QI \cdot x = \{L,H\}) \]
\[ \text{(2c) } (A \cdot x : \Delta \cdot x = \{L\} : L \in R \cdot y \cdot QI \cdot x) \]
\[ \text{(2d) } (A \cdot x : \Delta \cdot x = \{H\} : H \in R \cdot y \cdot QI \cdot x) \]
\[ \text{(3a) } co \cdot co \cdot \Delta \cdot QI \]
\[ \text{(3b) } (A \cdot x : N \cdot x : (\Delta \cdot x = \emptyset) = (R \cdot y \cdot QI \cdot x = \emptyset)) \]
\[ \text{(3c) } \Delta \subseteq R \cdot y \cdot QI \]
APPENDIX B2 PROOF OF THEOREM 5.10

(4a) \(\text{stable}3\cdot\gamma(Q_i, R\cdot\gamma Q_i)\)

(4b) \(\neg(\text{FCA0} \cdot \Psi \land \text{gdf} \cdot L \cdot \Psi) \Rightarrow \neg \text{gdf} \cdot (Q_i, R\cdot\gamma Q_i)\)

Notice that with (4a) and (4b) the proof of -2- is complete.

PROOFS
For the proofs of properties (1b,c,e,f), (2b), and (3a-c) we refer to appendix B0.

Property (1a) follows immediately from \(\Gamma_i = R\cdot\gamma(Q_i \land Z \cdot F_\alpha)\) and prop. 1.18a.

Proof of (1d)
Let \(s \in SW\). Then:

\[ gd0 \cdot \Delta \cdot s \]

\[ \Rightarrow \{\text{def. 1.11 (gd0) and (1c)}\} \]

\[ gd0 \cdot \Delta \cdot s \land (A i : 0 \leq i < cm : gd0 \cdot \Gamma_i \cdot s \land (\Gamma_i \cdot (g \cdot s) = \Delta \cdot (g \cdot s)))\]

\[ \Rightarrow \{\text{def. QI, def. 1.12 (consistent0), and Note 1 below}\} \]

\[ (A i : 0 \leq i < cm : gd0 \cdot \Gamma_i \cdot s \land (Q_i \cdot s = QI \cdot s))\]

Note 1

Assume: \((A i : 0 \leq i < cm : gd0 \cdot \Gamma_i \cdot s \land (\Gamma_i \cdot (g \cdot s) = \Delta \cdot (g \cdot s)))\). Then:

true

\[ \Rightarrow \{\text{Note 1.0}\} \]

\[ (B i : 0 \leq i < cm : rc_i \cdot s = 0) \]

\[ \Rightarrow \{\text{induction of Note 1.1, using } ((Q_{G0}, \Gamma_{G0}), rc_{G0}) = ((Q_{G0}, \Gamma_{G0}), rc_{G0})\} \]

\[ (A i : 0 \leq i < cm : \text{consistent}0 \cdot \Gamma \cdot Q_i \cdot s)\]

Note 1.0

\[ (A i : 0 \leq i < cm : rc_i \cdot s > 0) \]

\[ \Rightarrow \{rc_{G0} = rc_{G0}\} \]

\[ (A i : 0 \leq i < cm : rc_i \cdot s > 0) \]

\[ \Rightarrow \{\text{R2.0, R2.1 from sect. 5.0.0 (with } (Q_{G1}, \Lambda_{G1})\})\]
Note 1.1

Let $0 \forall i < m$. Assume: $(A \forall j : 0 \forall j < m : g_{dib\forall j} \cdot s \land (\Gamma_j \cdot (g \cdot s) = \Delta \cdot (g \cdot s))$. Then:

\[ r_{\forall i} \cdot s = 0 \]

$\Rightarrow \left\{ \begin{array}{l}
(R.0 (\text{sect. 5.0.0}), \text{using the assumption above and (with def. 4.9 (g_{dib})), and prop. 4.19a): \ (g_{dib\forall j} \cdot \Gamma_j \cdot s = g_{dib\forall j} \cdot \Lambda_j \cdot s \land (g_{dib\forall j} \cdot \Lambda_j \cdot s \equiv \neg CA \cdot (Q_j \cdot \Lambda_j \cdot s) ) \}
\end{array} \right.$

\[ \text{consistent} \cdot \Lambda_j \cdot Q_{\forall i \forall j \forall s} \]

$\Rightarrow \{ \text{def. 4.5 (consistent0) and def. } \Gamma_j \}
\text{consistent0} \cdot \Gamma_j \cdot Q_{\forall i \forall j \forall s}$

$\Rightarrow \{ \text{from assumption: } \Gamma_j \cdot (g \cdot s) = \Gamma_{\forall j} \cdot (g \cdot s), \text{and def. 1.12 (consistent0)} \}
\text{consistent0} \cdot \Gamma_{\forall j} \cdot Q_{\forall i \forall j \forall s}$

$\Rightarrow \{ \text{def. 4.5 (consistent0)} \}
\text{consistent0} \cdot \Lambda_{\forall j} \cdot Q_{\forall i \forall j \forall s}$

$\Rightarrow \{ \text{R2.0 (sect. 5.0.0), def. 4.5 (consistent0)} \}
\text{consistent0} \cdot \Gamma_{\forall j} \cdot Q_{\forall i \forall j \forall s} \land (r_{\forall i} \cdot s = 0)$

$\Box$ (end of proof (1d))

Proof of (2a)

The main part of the proof of (2a) is the same as in appendix B0. The only steps that need additional argumentation are the 4th and 5th step of the proof of (2a,1), which are (in app.B0):

\[ Q_{\forall i \forall j \forall s} \land (A \forall i : 0 \forall i < m : Q_{\forall i \forall j \forall s}) \]

$\Rightarrow \{ \text{def. } \Gamma_j \}
(A \forall i : 0 \forall i < m : Q_{\forall i \forall j \forall s}) \land (A \forall i : 0 \forall i < m : \Gamma_j \cdot s = \Gamma_{\forall i} \cdot s)$

$\Rightarrow \{ \text{def. } E \}
(A \forall i : 0 \forall i < m : Q_{\forall i \forall j \forall s}) \land (E \cdot \Gamma_j \cdot s = E \cdot s)$

These steps must now be replaced by the following step:

\[ Q_{\forall i \forall j \forall s} \land (A \forall i : 0 \forall i < m : Q_{\forall i \forall j \forall s}) \]

$\Rightarrow \{ \text{note 3} \}
(A \forall i : 0 \forall i < m : Q_{\forall i \forall j \forall s}) \land (E \cdot \Gamma_j \cdot s = E \cdot s)$

under the assumption (see proof of (2a,1) in appendix B0): $(p_{\forall i \forall j} = [x, z]) \land (\Delta \cdot x = \emptyset)$.

Note 3

Assume $(p_{\forall i \forall j} = [x, z]) \land (\Delta \cdot x = \emptyset) \land Q_{\forall i \forall j \forall s} \land (A \forall i : 0 \forall i < m : Q_{\forall i \forall j \forall s})$.

true

$\Rightarrow \{ \text{use R3.2 (sect. 5.0.0); compare with note 1.0} \}
(E \forall i : 0 \forall i < m : p_{\forall i \forall j \forall s} = 0)$

$\Rightarrow \{ \text{induction of note 3.0} \}
(E \cdot x \land E \cdot s) \lor (A \forall i : 0 \forall i < m : p_{\forall i \forall j \forall s} = 0)$
\[
\begin{align*}
\Rightarrow \{ \text{assumption} \} \\
(B \cdot x \land E \cdot z) \lor (A \cdot i : 0 \leq i < m : (Q_i \land Z \cdot \text{pc}_i \cdot s)) \\
\Rightarrow \{ \Gamma_1 = R \cdot \gamma(Q_i \land Z \cdot \text{pc}_i) \} \\
(B \cdot x \land E \cdot z) \lor (A \cdot i : 0 \leq i < m : \Gamma_1 \cdot x = \Gamma_1 \cdot z) \\
\Rightarrow \{ \text{def. } E \} \\
E \cdot x = E \cdot z
\end{align*}
\]

**Note 3.0**

Let \( 0 \leq i < m \) and assume \( (p \cdot s = [x,z]) \land (\Delta \cdot x = \emptyset) \land Q \cdot s \land Q \cdot z \). Then:

\[
\begin{align*}
\text{pc}_{i+1} \cdot s &= 0 \\
\Rightarrow \{ \Gamma_1 = R \cdot \gamma(Q_i \land Z \cdot \text{pc}_i), \text{ and } Q_i \cdot s \} \\
(p \cdot c_i \cdot x = 0) \land (\Gamma_1 \cdot x = \Gamma_1 \cdot z) \\
\Rightarrow \{ \text{note 3.1} \} \\
((\Gamma_1 \cdot x = \emptyset) \land (\Gamma_1 \cdot z = \emptyset)) \lor (p \cdot c_{i+1} \cdot s = 0) \\
\Rightarrow \{ \text{def. } E \} \\
(E \cdot x \land E \cdot z) \lor (p \cdot c_{i+1} \cdot x = 0)
\end{align*}
\]

**Note 3.1**

Let \( 0 \leq i < m \).

Assume \( (p \cdot s = [x,z]) \land (\Delta \cdot x = \emptyset) \land Q \cdot s \land Q \cdot x \land (p \cdot c_i \cdot s = 0) \land (\Gamma_1 \cdot x = \Gamma_1 \cdot z) \land (\Gamma_1 \cdot x = \emptyset) \). Then:

\[
\begin{align*}
\text{pc}_{i+1} \cdot s &= 0 \\
\Rightarrow \{ R.3.2 (\text{sect. 5.0.0}) \text{ and } p \cdot c_i \cdot s = 0 \land Q_i \cdot s, \text{ hence also } A_1 \cdot x = A_1 \cdot z \} \\
(A_{i+1} \cdot x \notin A_i \cdot x) \lor (A_{i+1} \cdot z \notin A_i \cdot z) \\
\Rightarrow \{ \text{from assumption: } (\Gamma_1 \cdot x = \emptyset) \land (\Gamma_1 \cdot z = \emptyset), \text{ and def. } \Gamma_1 \} \\
(\Gamma_{i+1} \cdot x \notin \Gamma_i \cdot x) \lor (\Gamma_{i+1} \cdot z \notin \Gamma_i \cdot z) \\
\Rightarrow \{ \text{from assumption: } (\Gamma_1 \cdot x = \emptyset) \land (\Gamma_1 \cdot z = \emptyset), \text{ and def. } \Gamma_{\text{max}} \} \\
(\Gamma_{\text{max}} \cdot x = \{L, M\}) \lor (\Gamma_{\text{max}} \cdot z = \{L, H\}) \\
\Rightarrow \{ \text{def. } Rt, \text{reflexivity cp-Q}_i \text{ and from assumption: } cp-Q_i \cdot [x, z] \} \\
Rt \cdot x \\
\Rightarrow \{ \text{def. } \Delta \} \\
\Delta \cdot x = \{L, H\} \\
\Rightarrow \{ \text{from assumption: } \Delta \cdot x = \emptyset \} \\
\text{false}
\end{align*}
\]

\( \square \) (end of proof (2a))
Proof of (7c)
\[ \Delta \cdot x = \{ L \} \]
= [def. \( \Delta \)]
\[ (\Delta \cdot x = \{ L \}) \land (\Gamma_{\max} \cdot x = \{ L \}) \land \neg Rt \cdot x \land \neg Rb \cdot x \]
[def. Rb and calculus]
\[ (\Delta \cdot x = \{ L \}) \land (\Gamma_{\max} \cdot x = \{ L \}) \land (E_{i,y} : 0 \leq i \leq m \land \exists y : \gamma \cdot y \neq \emptyset \land cpE \cdot Q_i \cdot [x,y]) \]
= [calculus]
\[ (\Delta \cdot x = \{ L \}) \land (\Gamma_{\max} \cdot x = \{ L \}) \land (E_{i,y} : \gamma \cdot y \neq \emptyset \land (E_{i,j} : 0 \leq i \leq m : cpE \cdot Q_i \cdot [x,y])) \]
= [note 2E]
\[ L \in R \cdot \gamma \cdot Q \cdot I \cdot x \]

Note 2E
Assume \( (\Delta \cdot x = \{ L \}) \land (\Gamma_{\max} \cdot x = \{ L \}) \land (\gamma \cdot y \neq \emptyset) \land (0 \leq i \leq m) \land cpE \cdot Q_i \cdot [x,y] \).
On account of \( cpE \cdot Q_i \cdot [x,y] \) a natural \( n \) and a set \( \{ j : 0 \leq j \leq n \} \) exist satisfying:

P2E.0 \[ (x = x_0) \land (x_n = y) \land (A_{j,j} : 0 \leq j \leq n : bcpE \cdot Q_i \cdot [x_j,y]) \]
On account of (1f) and Note 2F below they also satisfy:

P2E.1 \[ (A_{j,j} : 0 \leq j \leq n : \Delta \cdot x_j \neq \emptyset) \land (A_{j,j} : 0 \leq j \leq n : bcp-\gamma \cdot Q \cdot I \cdot [x_j,y]) \]
Then:
true
\[ \Rightarrow (P2E.1) \]
\[ (A_{j,j} : 0 \leq j \leq n : \Delta \cdot x_j \neq \emptyset) \]
[calculus]
\[ (E_{i,j} : 0 \leq j \leq n : \Delta \cdot x_j = \{ L,H \}) \lor (A_{j,j} : 0 \leq j \leq n : \emptyset \subseteq \Delta \cdot x_j \subseteq \{ L,H \}) \]
\[ \Rightarrow \{ notes 2E.0 and 2E.1, using the assumption \}
\[ L \in R \cdot \gamma \cdot Q \cdot I \cdot x \]

Note 2F
\[ \gamma \cdot y \neq \emptyset \]
\[ \Rightarrow \{ calculus, def. Rb \}
\[ (A_{i,x} : cpE \cdot Q_i \cdot [x,y] : \neg Rb \cdot x) \]
\[ \Rightarrow \{ def. \Delta \ and \ calculus \}
\[ (A \cdot x : (E_{i,j} : 0 \leq i \leq m : cpE \cdot Q_i \cdot [x,y]) : \Delta \cdot x \neq \emptyset) \]

Note 2E.0
Assume \( (\Delta \cdot x = \{ L \}) \land (\Gamma_{\max} \cdot x = \{ L \}) \land (\gamma \cdot y \neq \emptyset) \land (0 \leq i \leq m) \land cpE \cdot Q_i \cdot [x,y] \).

\[ (E_{i,j} : 0 \leq j \leq n : \Delta \cdot x_j = \{ L,H \}) \]
\[ \Rightarrow (2b) \]
\[ (E_{i,j} : 0 \leq j \leq n : R \cdot \gamma \cdot Q \cdot I \cdot x_j = \{ L,H \}) \]
\[ \Rightarrow \{ from P2E-1 : \gamma \cdot Q \cdot I \cdot [x_j] \ for \ all \ j, \ and \ def. R \cdot \gamma \}
\[ R \cdot \gamma \cdot Q \cdot I \cdot x = \{ L,H \} \]
Note 2E.1

Assume \( (\Delta \cdot x = \{ L \}) \land (\Gamma_{\max} \cdot x = \{ L \}) \land (\gamma \gamma \neq \emptyset) \land (0 \leq i < m) \land \text{cpE} \cdot Q_{1} \cdot [x, y] \).

\((A : j ; 0 \leq j \leq n : \emptyset \subseteq \Delta \cdot x_{j} \subseteq \{ L, H \})\)  
\(\Rightarrow \)  
\((1c)\), and from assumption, \( \Delta \cdot x = \{ L \} \)

\((A : j ; 0 \leq j \leq n : \emptyset \subseteq \Gamma_{j} \cdot x_{j} \subseteq \{ L, H \}) \land (\Gamma_{1} \cdot x = \{ L \})\)  
\(\Rightarrow \)  
\((R3.1 \text{ and } R3.0 \text{ (sect. S.0.0)}, \text{using}) \)

\((\emptyset \subseteq \Gamma_{1} \cdot x) \land (\Gamma_{1} \cdot x = \Lambda_{1} \cdot z) \land (A j ; 0 \leq j \leq n : (E s : SW \cdot s : pu \cdot s = [x_{j}, x_{j+1}])})\)

\((A : j ; 0 \leq j \leq n : \Gamma_{j} \cdot x_{j} = \{ L \})\)  
\(\Rightarrow \)  
\(\{ \gamma \gamma \neq \emptyset \text{ (assumption), } (1a), \text{ and } x_{n} = y \}\)  
\(\gamma \gamma = \{ L \}\)

\(\Rightarrow \)  
\{from P2E.1 : \text{cpE} \cdot Q_{1} \cdot [x, y] \}, and def. \( R \cdot \gamma \)

\(\) \(L \in R \cdot \gamma \cdot Q_{1} \cdot x\)

\(\) \(\square \) \(\) (end of proof (2c))

Proof of (2d)

The proof of (2d) is similar to the proof of (2c).

Proof of (4a)

\(\) \(\text{true}\)

\(= \)  
\((3b,c)\)

\((A x : R \cdot \gamma \cdot Q_{1} \cdot x \in \{ \{ L \}, \{ H \} \} : R \cdot \gamma \cdot Q_{1} \cdot x = \Delta \cdot x)\)

\(\Rightarrow \)  
\{calculus\}

\((A x : R \cdot \gamma \cdot Q_{1} \cdot (g \cdot x) \in \{ \{ L \}, \{ H \} \} : R \cdot \gamma \cdot Q_{1} \cdot (g \cdot x) = \Delta \cdot (g \cdot x))\)

\(= \)  
\{def. 1.12 (coco0) and (3a)\}

\(\) \(\text{coco0} : (R \cdot \gamma \cdot Q_{1}) \cdot Q_{1}\)

\(= \)  
\{prop. 1.22c\}

\(\) \(\text{stable0} : \gamma(Q_{1}, R \cdot \gamma \cdot Q_{1})\)

\(\Rightarrow \)  
\{lemmas 3.5, 4.22b, and 5.7c\}

\(\) \(\text{stable3} : \gamma(Q_{1}, R \cdot \gamma \cdot Q_{1})\)

\(\) \(\square \) \(\) (end proof (4a))
Proof of (4b)

\[ \neg(FCA_0 \land cgdil \land \Psi) \]
\[ = \text{[calculus]} \]
\[ \neg FCA_0 \lor \neg \neg cgdil \lor \Psi \]
\[ \Rightarrow \text{[note 4A and note 4B]} \]
\[ (B : i,s : (0 \leq i < m) \land SW \cdot s : \neg gdo \cdot A_{i,s}) \]
\[ = \text{[def. 4.9 (gd0) and def. 11]} \]
\[ (B : i,s : (0 \leq i < m) \land SW \cdot s : \neg gdo \cdot \Gamma_{i,s}) \]
\[ \Rightarrow \text{[1a]} \]
\[ (B : s : SW \cdot s : \neg \neg gdo \cdot A_{i,s}) \]
\[ \Rightarrow \text{[def. 1.11 (gd0) and (2a,b)]} \]
\[ (B : s : SW \cdot s : \neg gdo \cdot (R \cdot \Pi \cdot QI) \cdot s) \]
\[ = \text{[def. 1.11 (cgd0)]} \]
\[ \neg gdo \cdot (QI, R \cdot \Pi \cdot QI) \]

Note 4A

\[ \neg FCA_0 \land \Psi \]
\[ = \text{[def. 4.13 (FCA0)]} \]
\[ \neg (A : s : SW \cdot s : (B : i : 0 \leq i < m : \neg CA \cdot \Psi_{i,s})) \]
\[ = \text{[calculus]} \]
\[ (B : s : SW \cdot s : (A : i : 0 \leq i < m : CA \cdot \Psi_{i,s})) \]
\[ \Rightarrow \text{[prop. 4.19a]} \]
\[ (B : s : SW \cdot s : (A : i : 0 \leq i < m : \neg gdo \cdot A_{i,s})) \]

Note 4B

\[ \neg cgdil \land \Psi \]
\[ = \text{[def. 4.14 (cgdil)]} \]
\[ \neg (A : i : 0 \leq i < m : cgdil \land \Psi_{i}) \]
\[ = \text{[def. 4.10 (cgdil)]} \]
\[ \neg (A : i : 0 \leq i < m : (A : s : SW \cdot s : gdo \cdot A_{i,s} \land \neg CA \cdot \Psi_{i,s})) \]
\[ = \text{[calculus]} \]
\[ (B : i,s : (0 \leq i < m) \land SW \cdot s : \neg gdo \cdot A_{i,s} \land \neg CA \cdot \Psi_{i,s}) \]
\[ \Box \text{[end proof (4b)]} \]

\[ \Box \text{[end of proofs of property]} \]
Similar as in appendix B1, it is possible, using the construction above, to prove an additional lemma that can be used to prove theorem 5.12b.

**B2 Lemma**

\[
\text{cgd0} \cdot \gamma_s \land (\text{stables} \cdot \gamma = \{Q, A : (\text{destore}(Q, A) = \gamma_s) \land \text{cocoI} \cdot A \cdot Q : (Q, A)\})
\]

\[
\Rightarrow (\text{feasible} \cdot \gamma = \{Q, A : (\text{destore}(Q, A) = \gamma_s) \land \text{cocoI} \cdot A \cdot Q : (Q, A)\})
\]

**Proof**

Let \( \Pi \) satisfy \( \text{feasible} \cdot \gamma \cdot \Pi \). On account of lemma 5.8b a state list \( \Psi \) exist satisfying \( \text{FSL3} \cdot \Psi \land (E i : 0 < i \# \Psi : \Psi_i = \Pi) \). The construction and properties above are used in the remaining of this proof.

\[
\text{cgdo} \cdot \gamma_s \land (\text{stables} \cdot \gamma = \{Q, A : (\text{destore}(Q, A) = \gamma_s) \land \text{cocoI} \cdot A \cdot Q : (Q, A)\})
\]

\[
\Rightarrow \{(4a) \text{and from def. 4.4, using } R \cdot \gamma \cdot QI \in \text{NST} : \text{destore}(R \cdot \gamma \cdot QI) = R \cdot \gamma \cdot QI\}
\]

\[
\text{cgdo} \cdot \gamma_s \land (\text{<QI}_1, R \cdot \gamma \cdot QI) = \gamma_s
\]

\[
\Rightarrow \{(3b)\text{e and def. 1.11 (cgdo)}\}
\]

\[
((QI_1, R \cdot \gamma \cdot QI) = \gamma_s) \land (A i : \text{SW} \cdot s : \text{gdob} \cdot \Delta \cdot s)
\]

\[
\Rightarrow \{(1d)\}
\]

\[
((QI_1, R \cdot \gamma \cdot QI) = \gamma_s) \land (A i, s : \text{SW} \cdot s \land (0 < i < m) : Q_i, s = QI_1 \cdot s)
\]

\[
\Rightarrow \{(\Gamma_1 \in R \cdot \gamma \cdot (Q_i \land Z \cdot pc_i), (Q_i \land Z \cdot pc_i) \land \text{Q}_i, \text{and monotonicity } R \cdot \gamma\}
\]

\[
((QI_1, R \cdot \gamma \cdot QI) = \gamma_s) \land (A i : 0 < i < m : Q_i = QI_1) \land (\Gamma_1 \in R \cdot \gamma \cdot QI)
\]

\[
\Rightarrow \{\text{Notes B2.0, B2.1, and B2.2}\}
\]

\[
(A i : 0 < i < m : (Q_i, \Gamma_1) = \gamma_s)
\]

\[
\Rightarrow \{\text{Note B2.5}\}
\]

\[
(A i : 0 < i < m : ((Q_i, \Gamma_1) = \gamma_s) \land \text{cocoI} \cdot \Delta_I \cdot Q_i)
\]

\[
\Rightarrow \{(A i : 0 < i < m : (Q_i, \Gamma_1) = \text{destore}(Q_i, A_i)) \land (E i : 0 < i < m : (Q_i, A_i) = \Pi)\}
\]

\[
\Pi \in \{Q, A : (\text{destore}(Q, A) = \gamma_s) \land \text{cocoI} \cdot A \cdot Q : (Q, A)\}
\]

**Note B2.0**

Let \( N \cdot x \). Assume \( A i : 0 < i < m : \Gamma_1 \in R \cdot \gamma \cdot QI \). Then:

\[
R \cdot \gamma \cdot QI \cdot x = \emptyset
\]

\[
\Rightarrow \{\text{assumption}\}
\]

\[
(A i : 0 < i < m : \Gamma_1 \cdot x = \emptyset)
\]

**Note B2.1**

Let \( N \cdot x \). Then:

\[
R \cdot \gamma \cdot QI \cdot x \in \{(L)\}
\]

\[
\Rightarrow \{(3b,e)\}
\]

\[
(\Delta x \in \{(L)\}) \land (\Delta x = R \cdot \gamma \cdot QI \cdot x)
\]

\[
\Rightarrow \{(1c)\}
\]

\[
(A i : 0 < i < m : \Gamma_1 \cdot x = R \cdot \gamma \cdot QI \cdot x)
\]
Note B.2.2

Let $N$-x. Assume $(A_i : 0 \leq i < m : (Q_i = Q_i))$. Then:

$$R \cdot \gamma \cdot Q \cdot I \cdot x = [L,H]$$

\[\vdash\] {def. $R$}

$$(E_{z_0} : (L \in \gamma \cdot z_0) \land (H \in \gamma \cdot z_0) : cp - Q \cdot I \cdot [x,z_0] \land cp - Q \cdot I \cdot [x,z_1])$$

\[\vdash\] {1(a)}

$$(E_{z_0} : (A_i : 0 \leq i < m : L \in \Gamma_1 \cdot z_0) : cp - Q \cdot I \cdot [x,z_0])$$

$$\land (E_{z_1} : (A_i : 0 \leq i < m : H \in \Gamma_1 \cdot z_1) : cp - Q \cdot I \cdot [x,z_1])$$

\[\vdash\] {induction using Note B.2.3}

$$(A_i : 0 \leq i < m : (L \in \Gamma_1 \cdot x) \land (H \in \Gamma_1 \cdot x))$$

\[\vdash\] {calculus}

$$(A_i : 0 \leq i < m : \Gamma_1 \cdot x = [L,H])$$

Note B.2.3

Assume $(A_i : 0 \leq i < m : (Q_i = Q_i))$. Let $\{u,v\} \subseteq N$. We will prove:

1. $(A_i : 0 \leq i < m : L \in \Gamma_1 \cdot u) \land bcp \cdot Q \cdot I \cdot [u,v] \Rightarrow (A_i : 0 \leq i < m : L \in \Gamma_1 \cdot v)$

2. $(A_i : 0 \leq i < m : H \in \Gamma_1 \cdot u) \land bcp \cdot Q \cdot I \cdot [u,v] \Rightarrow (A_i : 0 \leq i < m : H \in \Gamma_1 \cdot v)$

Since the proofs of 1 and 2 are similar, we only give the proof of 1.

Assume $(A_i : 0 \leq i < m : L \in \Gamma_1 \cdot u)$. Then:

$$bcp \cdot Q \cdot I \cdot [u,v]$$

\[\vdash\] {def. $bcp$}

$$(E_s : pn \cdot s = [u,v] : Q \cdot I \cdot s)$$

\[\vdash\] {assumption: $(A_i : 0 \leq i < m : (Q_i = Q_i))$}

$$(E_s : pn \cdot s = [u,v] : (A_i : 0 \leq i < m : Q_i \cdot s))$$

\[\vdash\] {use R(i) 5.0.0; compare with note 1.0 in the proof of (1d) above}

$$(E_s : (pn \cdot s = [u,v]) \land (A_i : 0 \leq i < m : Q_i \cdot s) \land (E_j : 0 \leq j < m : pc_{j \cdot v} = 0))$$

\[\vdash\] {induction using Note B.2.4 and $pc_{m} = pc_{0}$}

$$(E_s : (pn \cdot s = [u,v]) \land (A_i : 0 \leq i < m : Q_i \cdot s) \land (A_j : 0 \leq j < m : pc_{j \cdot v} = 0))$$

\[\vdash\] {calculus}

$$(E_s : pn \cdot s = [u,v] : (A_i : 0 \leq i < m : (Q_i \land Z \cdot pc_{i \cdot v}) \cdot s))$$

\[\vdash\] {1. $\Gamma_i = R \cdot \gamma \cdot (Q_i \land Z \cdot pc_{i})$, def. $R$, calculus}

$$(A_i : 0 \leq i < m : \Gamma_i \cdot u = \Gamma_i \cdot v)$$

\[\vdash\] {assumption: $(A_i : 0 \leq i < m : L \in \Gamma_1 \cdot u)$}

$$(A_i : 0 \leq i < m : L \in \Gamma_1 \cdot v)$$
Note B2.4

Let \( SW \rightarrow \). Assume \((p \cup s = [u,v]) \land (A \cup i : 0 \leq i \leq m : Q_i \uplus s) \land (A \cup i : 0 \leq i \leq m : L \in \Gamma_i \cup u)\).

Let \( 0 < j < m \). Assume \( p_{c_j} \uplus s = 0 \). Then:

true

\[\{\text{assumption}\}\]
\[(L \in \Gamma_{j+1} \cup u \lor \Gamma_{j+1} \cup v) \land (L \in \Gamma_{j} \cup u \lor \Gamma_{j} \cup v) \land (p_{c_j} \uplus s = 0)\]

\[\{\text{def. } \Gamma_j, \Gamma_{j+1}\}\]
\[(L \in A_{j+1} \cup u \lor A_{j+1} \cup v) \land (L \in A_j \cup u \lor A_j \cup v) \land (p_{c_j} \uplus s = 0)\]

\[\{\text{calculus}\}\]
\[((A_{j+1} \cup u \lor A_{j+1} \cup v = [L_u,H]) \lor (A_{j+1} \cup u \lor A_{j+1} \cup v \leq A_j \cup u \lor A_j \cup v)) \land (p_{c_j} \uplus s = 0)\]

\[\{\text{R3.1 and R3.2 (sect. 5.0.0), using from the assumption: } Q_i \uplus s\}\]
\[p_{c_{j+1}} \uplus s = 0\]

Note B2.5

Let \( 0 \leq i < m \). Then:

\((Q_i, \Gamma_i) = \gamma_{s_i}\)

\[\{\text{prop. 1.34a}\}\]

\[\text{stable-}\gamma(Q_i, \Gamma_i)\]

\[\{\text{prop. 1.22c, def. } \Gamma_i\}\]

\[\text{coco} \cdot \Gamma_i; Q_i\]

\[\{\text{def. 1.15 (coco)}\}\]

\((A \cup s, x, y : Q_i \uplus s \land (p \cup s = [x,y]) : \Gamma_i \uplus x = \Gamma_i \uplus y)\)

\[\{\text{calculus}\}\]

\[(A \cup s, x, y : Q_i \uplus s \land (p \cup s = [x,y]) \land (\Gamma_i \uplus x \cup \Gamma_i \uplus y \neq \emptyset) : \Gamma_i \uplus x = \Gamma_i \uplus y) \land \]

\[(A \cup s, x, y : Q_i \uplus s \land (p \cup s = [x,y]) \land (\Gamma_i \uplus x \cup \Gamma_i \uplus y = \emptyset) : \Gamma_i \uplus x = \Gamma_i \uplus y)\]

\[\{\text{def. } \Gamma_i\}\]

\[(A \cup s, x, y : Q_i \uplus s \land (p \cup s = [x,y]) \land (\Gamma_i \uplus x \cup \Gamma_i \uplus y \neq \emptyset) : A_i \uplus x = A_i \uplus y) \land \]

\[(A \cup s, x, y : Q_i \uplus s \land (p \cup s = [x,y]) \land (\Gamma_i \uplus x \cup \Gamma_i \uplus y = \emptyset) : A_i \uplus x \lor A_i \uplus y \notin \{[L_u],[H]\})\]

\[\{\text{R3.1 (sect. 5.0.0)}\}\]

\[(A \cup s, x, y : Q_i \uplus s \land (p \cup s = [x,y]) \land (\Gamma_i \uplus x \cup \Gamma_i \uplus y \neq \emptyset) : A_i \uplus x = A_i \uplus y) \land \]

\[(A \cup s, x, y : Q_i \uplus s \land (p \cup s = [x,y]) \land (\Gamma_i \uplus x \cup \Gamma_i \uplus y = \emptyset) : p_{c_i} \uplus s = 0)\]

\[\{\text{R3.0 (sect. 5.0.0)}\}\]

\[(A \cup s, x, y : Q_i \uplus s \land (p \cup s = [x,y]) \land (\Gamma_i \uplus x \cup \Gamma_i \uplus y \neq \emptyset) : A_i \uplus x \uplus A_i \uplus y) \land \]

\[(A \cup s, x, y : Q_i \uplus s \land (p \cup s = [x,y]) \land (\Gamma_i \uplus x \cup \Gamma_i \uplus y = \emptyset) : A_i \uplus x = A_i \uplus y)\]

\[\{\text{calculus and def. 1.15 (coco)}\}\]

\[\text{coco} \cdot A_i; Q_i\]

\[\bowtie (\text{end of proof B2})\]
APPENDIX C  On the pessimism caused by assumption (1) on pass−delays

In section 5.0.0 the assumption, called (1), is made that the pass-delay in a switch is zero if the value that needs to be transported is \{L,H\} or a stored value. The motivation for this assumption is that it can only lead, in some cases, to a pessimistic modelling of circuit behaviour. In this appendix we prove that the inclusion of assumption (1) does not change the correctness criterion for initial behaviour; it does not change the resulting states for initial behaviour if this initial behaviour is correct; and it does not weaken the correctness criterion for dynamic behaviour.

In the sequel, the (imaginary) model without assumption (1) will be called the alternative model. In the alternative model, the counterparts of the notions stable3, next3, FSL3, RSL3, and WF3 from chapter 5 are called stable3A, next3A, FSL3A, RSL3A, and WF3A respectively.

Let \( C \in \text{CIR} \) and \( \gamma \in \text{NST} \).

The correctness criterion for initial behaviour in the model from ch. 5 is (cf. section 5.0.3):

\[
\text{FSL3}\cdot \gamma \models (cgdl_l \land \text{FCA}0 \land \text{CFL} \land \text{cst}l) ,
\]

which equals (cf. sect. 5.0):

\[
\text{WM3}\cdot C \cdot \gamma \land \text{CF} \cdot \gamma_0 .
\]

Similarly, the correctness criterion for initial behaviour in the alternative model is:

\[
\text{FSL3A} \cdot \gamma \models (cgdl_l \land \text{FCA}0 \land \text{CFL} \land \text{cst}l) .
\]

The correctness criterion for dynamic behaviour is called well-functioning. We will prove:

In part a: \((\text{WM3}\cdot C \cdot \gamma \land \text{CF} \cdot \gamma_0) \Rightarrow (\text{FSL3A} \cdot \gamma \models (cgdl_l \land \text{FCA}0 \land \text{CFL} \land \text{cst}l))) .

In part b: \((\text{WM3}\cdot C \cdot \gamma \land \text{CF} \cdot \gamma_0) \Rightarrow (\text{FSL3A} \cdot \gamma \models (cgdl_l \land \text{FCA}0 \land \text{CFL} \land \text{cst}l))) .

In part c: If the initial behaviour is correct, then the resulting states are equal in both models.

In part d we will explain that modelling restricted pass-delays in the alternative model cannot be done in a similar way as in chapter 5. Furthermore, we will—in informally—argue that

\[
\text{WF3D}\cdot C \cdot \gamma (\text{P1},\text{p}c) \Rightarrow \text{WF3A} \cdot \text{D} \cdot C \cdot \gamma (\text{P1},\text{p}c) \quad (\text{for all D and (P1},\text{p}c)).
\]

In order to give these proofs we assume that a state transition in the alternative model, say from \((\text{Q}_0,\text{\Gamma}_0),\text{p}c\text{p}_0,\text{p}_0)\) to \((\text{Q}_1,\text{\Gamma}_1),\text{p}c\text{p}_1,\text{p}_1)\), satisfies R0, R0, R1, R2, R2 (0\(\leq\)i) from sect. 5.0.0, and R0, R1, and R2 below (which express assumptions (0a,b) and (2); cf. section 5.0.0).

RR0 \((A \cdot s,x,y : (\text{p}_n \cdot s = [x,y]) \land (\text{p}_c \cdot s > 0) : \text{Q}_1 \cdot s \land (\text{\Gamma}_1 \cdot x \neq \text{\Gamma}_1 \cdot y)) \)

RR1 \((A \cdot s : (\text{p}_c \cdot s > 0) \land (\text{p}_c \cdot s > 0) : \text{p}_c \cdot s = \text{p}_c \cdot s - 1) \)

RR2 \((A \cdot s : (\text{p}_c \cdot s > 0) \land (\text{p}_c \cdot s = 0) : -\text{Q}_0 \cdot s \lor ((\text{TV} \cdot \text{\Gamma}_1 \cdot s \neq \text{TV} \cdot \text{\Gamma}_1 \cdot s) \land (\text{TV} \cdot \text{\Gamma}_0 \cdot s \neq \{L,H\})) \)

In section 5.0.0 the assumption, called (1), is made that the pass-delay in a switch is zero if the value that needs to be transported is \{L,H\} or a stored value. The motivation for this assumption is that it can only lead, in some cases, to a pessimistic modelling of circuit behaviour. In this appendix we prove that the inclusion of assumption (1) does not change the correctness criterion for initial behaviour; it does not change the resulting states for initial behaviour if this initial behaviour is correct; and it does not weaken the correctness criterion for dynamic behaviour.

In the sequel, the (imaginary) model without assumption (1) will be called the alternative model. In the alternative model, the counterparts of the notions stable3, next3, FSL3, RSL3, and WF3 from chapter 5 are called stable3A, next3A, FSL3A, RSL3A, and WF3A respectively.

Let \( C \in \text{CIR} \) and \( \gamma \in \text{NST} \).

The correctness criterion for initial behaviour in the model from ch. 5 is (cf. section 5.0.3):

\[
\text{FSL3}\cdot \gamma \models (cgdl_l \land \text{FCA}0 \land \text{CFL} \land \text{cst}l) ,
\]

which equals (cf. sect. 5.0):

\[
\text{WM3}\cdot C \cdot \gamma \land \text{CF} \cdot \gamma_0 .
\]

Similarly, the correctness criterion for initial behaviour in the alternative model is:

\[
\text{FSL3A} \cdot \gamma \models (cgdl_l \land \text{FCA}0 \land \text{CFL} \land \text{cst}l) .
\]

The correctness criterion for dynamic behaviour is called well-functioning. We will prove:

In part a: \((\text{WM3}\cdot C \cdot \gamma \land \text{CF} \cdot \gamma_0) \Rightarrow (\text{FSL3A} \cdot \gamma \models (cgdl_l \land \text{FCA}0 \land \text{CFL} \land \text{cst}l))) .

In part b: \((\text{WM3}\cdot C \cdot \gamma \land \text{CF} \cdot \gamma_0) \Rightarrow (\text{FSL3A} \cdot \gamma \models (cgdl_l \land \text{FCA}0 \land \text{CFL} \land \text{cst}l))) .

In part c: If the initial behaviour is correct, then the resulting states are equal in both models.

In part d we will explain that modelling restricted pass-delays in the alternative model cannot be done in a similar way as in chapter 5. Furthermore, we will—in informally—argue that

\[
\text{WF3D}\cdot C \cdot \gamma (\text{P1},\text{p}c) \Rightarrow \text{WF3A} \cdot \text{D} \cdot C \cdot \gamma (\text{P1},\text{p}c) \quad (\text{for all D and (P1},\text{p}c)).
\]

In order to give these proofs we assume that a state transition in the alternative model, say from \((\text{Q}_0,\text{\Gamma}_0),\text{p}c\text{p}_0,\text{p}_0)\) to \((\text{Q}_1,\text{\Gamma}_1),\text{p}c\text{p}_1,\text{p}_1)\), satisfies R0, R0, R1, R2, R2 (0\(\leq\)i) from sect. 5.0.0, and R0, R1, and R2 below (which express assumptions (0a,b) and (2); cf. section 5.0.0).

RR0 \((A \cdot s,x,y : (\text{p}_n \cdot s = [x,y]) \land (\text{p}_c \cdot s > 0) : \text{Q}_1 \cdot s \land (\text{\Gamma}_1 \cdot x \neq \text{\Gamma}_1 \cdot y)) \)

RR1 \((A \cdot s : (\text{p}_c \cdot s > 0) \land (\text{p}_c \cdot s > 0) : \text{p}_c \cdot s = \text{p}_c \cdot s - 1) \)

RR2 \((A \cdot s : (\text{p}_c \cdot s > 0) \land (\text{p}_c \cdot s = 0) : -\text{Q}_0 \cdot s \lor ((\text{TV} \cdot \text{\Gamma}_1 \cdot s \neq \text{TV} \cdot \text{\Gamma}_1 \cdot s) \land (\text{TV} \cdot \text{\Gamma}_0 \cdot s \neq \{L,H\})) \)
PART a
Let C and \( \gamma \) satisfy \( \text{CIR-C} \land \text{NST-}\gamma \).
If pass-delays are arbitrary, the pass-delays in the alternative model can be chosen to be zero in case the value that needs to be transported is \([L,H]\) or a stored value. Consequently, for all extended states \((\Pi_{rc,pc})\):
\[
\text{next}^3A\cdot\gamma(\Pi_{rc,pc}) \subseteq \text{next}^3A\cdot\gamma(\Pi_{rc,pc})
\]
As a direct consequence (compare with lemma 5.8a):
\[
\text{FSL}3\cdot\gamma \subseteq \text{FSL}3A\cdot\gamma
\]
which implies (compare with lemma 5.9a):
\[
(\text{WM3-C} \land GF\cdot\gamma) \Rightarrow (\text{FSL}3A\cdot\gamma \subseteq \text{cgdl} \land FCA0 \land CFL \land \text{csl})
\]

PART b
Let C, \( \gamma \), and \( \Phi \) satisfy \( \text{CIR-C} \land \text{NST-}\gamma \land (\Phi \in \mathcal{L}^*(\text{ST})) \land \text{FSL}3A\cdot\gamma \Phi \). We will prove that
\[
(\exists \Omega : \text{FSL}3\cdot\gamma \Omega : (\Omega \in (\text{CFL} \land \text{cgdl} \land FCA0 \land \text{csl})) \Rightarrow (\Phi \in (\text{CFL} \land \text{cgdl} \land FCA0 \land \text{csl})))
\]
which directly implies:
\[
(\text{WM3-C} \land GF\cdot\gamma) \Rightarrow (\text{FSL}3A\cdot\gamma \subseteq \text{cgdl} \land FCA0 \land \text{CFL} \land \text{csl})
\]
This proof is given in two steps. Step 0 transforms \( \Phi \) into \( \Psi \in \text{FSL}3A\cdot\gamma \) such that pass delays are active in states \( \Psi \), if the value that needs to be transported is \([L,H]\), and \( \Psi \) satisfies:
\[
(\Psi \in (\text{CFL} \land \text{cgdl} \land FCA0 \land \text{csl})) \Rightarrow (\Psi \in (\text{CFL} \land \text{cgdl} \land FCA0 \land \text{csl}))
\]
Step 1 transforms \( \Psi \) into \( \Omega \in \text{FSL}3\cdot\gamma \) satisfying:
\[
(\Omega \in (\text{CFL} \land \text{cgdl} \land FCA0 \land \text{csl})) \Rightarrow (\Psi \in (\text{CFL} \land \text{cgdl} \land FCA0 \land \text{csl}))
\]

STEP 0
Let \( \Phi_i = (Q_i, \Gamma_i) \) and let \( ri, pi \in \mathbb{L}(\mathbb{D}) \) be s.t. \( \text{next}^3A\cdot\gamma((Q_0, \Gamma_0), r_i, p_i; ((Q_{\omega}, \Gamma_{\omega}), r_i, p_i)) \) for all \( i = 0 \leq i < \# \Phi \), with \( + \) defined modulo \( \# \Phi \).
Let \( Q \in \mathbb{S}T \) and \( C \in \mathbb{N} - \mathbb{B} \) be defined by:
\[
Q \cdot s = (B_i : 0 \leq i < \# \Phi : Q_i \cdot s)
\]
\[
C \cdot x = (B_{\Delta x} : \text{cp} \cdot Q \cdot \{x \cdot \Delta x\} \land \text{cp} \cdot Q \cdot \{x \cdot \Delta x\} : (L \in \gamma \cdot x) \land (H \in \gamma \cdot x))
\]
For \( i = 0 \leq i < \# \Phi \) the extended states \((q_i, \Lambda_i, r_i, p_i)\) are defined by:
\[
\Lambda_i \cdot x = \begin{cases} 
(L_i, H_i) & \text{if } C \cdot x \\
\Gamma_i \cdot x & \text{if } \neg C \cdot x
\end{cases}
\]
\[
q_i \cdot s = (Q_i \cdot s \lor (Q \cdot s \land (TV \cdot A_i \cdot s = [L,H])))
\]
\[
p_i \cdot s = \begin{cases} 
0 & \text{if } TV \cdot A_i \cdot s = [L,H] \\
p_i \cdot s & \text{if } TV \cdot A_i \cdot s = [L,H]
\end{cases}
\]
\[
r_i \cdot s = \begin{cases} 
0 & \text{if } \text{consistent} \cdot A_i \cdot Q_i \cdot s \\
ri \cdot s & \text{if } \neg \text{consistent} \cdot A_i \cdot Q_i \cdot s \land (A_i : q_i \cdot s = Q_i \cdot s) \\

\# \Phi & \text{if } \neg \text{consistent} \cdot A_i \cdot Q_i \cdot s \land (B_i : q_i \cdot s \# Q_i \cdot s) \land \text{consistent} \cdot A_i \cdot Q_i \cdot s \\
r_i \cdot s - 1 & \text{if } \neg \text{consistent} \cdot A_i \cdot Q_i \cdot s \land (B_i : q_i \cdot s \# Q_i \cdot s) \land \neg \text{consistent} \cdot A_i \cdot Q_i \cdot s
\end{cases}
\]
From 0.0 below follows that \( r_i \) is correctly defined.
From the above follows that:

0.0 \( (E \ j:: q_j \neq Q_j \ L ) \)

\[ \Rightarrow \{ \text{definitions of } q, \Lambda, \text{ and } Q \} \]

\( (A \ j:: q_j \ L ) \land (E \ j:: \neg Q_j \ L ) \land (E \ j:: Q_j \ L ) \)

\[ \Rightarrow \{ \text{calculus} \} \]

\( (E \ j:: \neg Q_j \ L \land Q_j \ L \land q_{j \ L} \ L ) \)

\[ \Rightarrow \{ \text{def. } A_j \} \]

\( (E \ j:: \text{consistent} \ L \Gamma_{j \ L} \land Q_j \ L \land q_{j \ L} \ L ) \)

\[ \Rightarrow \{ \text{def. } 4.5 \text{ (consistent)} \} \]

\( (E \ j:: \text{consistent} \ L \Gamma_{j \ L} \land q_{j \ L} \ L ) \)

0.1 Let \( 0 \leq j < \# \Phi \). Then:

\( \text{consistent} \ L \Gamma_{j \ L} \land Q_j \ L \land q_{j \ L} \ L \)

\[ \Rightarrow \{ \text{def. of } \Lambda \text{ and def. } 4.5 \text{ (consistent)} \} \]

\( \text{consistent} \ L \Gamma_{j \ L} \land q_{j \ L} \ L \)

0.2 Let \( 0 \leq j < \# \Phi \). Then:

\( \text{consistent} \ L \Gamma_{j \ L} \land q_{j \ L} \ L \land q_{j \ L} \ L \)

\[ \Rightarrow \{ \text{def. } 4.5 \text{ (consistent)} \} \]

\( \text{consistent} \ L \Gamma_{j \ L} \land q_{j \ L} \ L \)

0.3 Let \( x \in N \). Then:

\( (E \ i:: A \ L x = \{ L,H \} ) \)

\[ \Rightarrow \{ \text{definitions of } Q, C, \text{ and } A \} \]

\( (A \ i:: A \ L x = \{ L,H \} ) \)

\[ \Rightarrow \{ \text{definitions of } q \text{ and } p \} \]

\( (E \ z_0 \ z_1:: (A \ i:: cp-(q_i \land z_0 \land z_1 \land z_2) \land cp-(q_i \land z \land z_0 \land z_1 \land z_2) \land (y \in y_{z_0}) \land (H \in y_{z_1}) ) \)

0.4 Let \( x \in N \) and \( 0 \leq i < \# \Phi \). Then:

\( A \ L x = \not \{ L,H \} \)

\[ \Rightarrow \{ \text{definitions of } \Lambda, q, \text{ and } p \} \]

\( \Gamma \ L x = A \ L x \land (A \ y:: N \ y : cp-(q_i \land z \land z_0 \land z \land z_0 \land z_1 \land z_2) \land (y \in y_{z_0}) \land (H \in y_{z_1}) ) \)

0.5 \( (A \ i:: 0 \leq i < \# \Phi : \text{CF}-(q_i, A_i) ) \)

\[ = \{ \text{def. } 4.29 \text{ (CF)} \} \]

\( (A \ i:: 0 \leq i < \# \Phi : \text{CF}-(q_i, A_i) ) \land (A \ x:: A \ L x = \{ L,H \} ) \)

\[ \Rightarrow \{ \text{definitions of } \Lambda \text{ and } x \} \]

\( (A \ i:: 0 \leq i < \# \Phi : (A_i = \Gamma_i \land (q_i = Q_i) ) \)

\[ \Rightarrow \{ \text{definitions of } r \text{ and } p, \text{ using } R2.0 \text{ for } (\Phi \ L r, p) \} \]

\( (A \ i:: 0 \leq i < \# \Phi : ((q_i, A_i), r_0, p_0) = (\Phi \ L r, p_0) ) \)

\[ = \{ \text{def. } 4.29 \text{ (CF)} \} \]

\( (A \ i:: 0 \leq i < \# \Phi : \text{CF}-(q_i, A_i) ) \land (A \ x:: A \ L x = \{ L,H \} ) \)

\[ \Rightarrow \{ \text{definitions of } \Lambda \text{ and } x \} \]

\( (A \ i:: 0 \leq i < \# \Phi : (A_i = \Gamma_i \land (q_i = Q_i) ) \)

\[ \Rightarrow \{ \text{definitions of } r \text{ and } p, \text{ using } R2.0 \text{ for } (\Phi \ L r, p) \} \]

\( (A \ i:: 0 \leq i < \# \Phi : ((q_i, A_i), r_0, p_0) = (\Phi \ L r, p_0) ) \)
Let $\Psi$ be defined by

\[(\#_i^\Psi = \#_i^\Phi) \wedge (A_i \colon 0 \leq i < \#_i^\Psi : \Psi_i = (q_i, A_i)) .\]

We will now argue that

\[(A_i \colon 0 \leq i < \#_i^\Psi : \text{next} 3 A \gamma (\Psi_i, s_i, p_i) \cdot (\Psi_{i+1}, s_{i+1}, p_{i+1})) .\]

From R0.2 and R0.3 for $(\Phi, r, p)$, the definitions of $r$ and $q$, and 0.1 and 0.2 above follows that $(\Psi, r, p)$ satisfies R0.2 and R0.3.

From R1.2 for $(\Phi, r, p)$, def. 4.6 (K), and 0.3 and 0.4 above follows that $(\Psi, r, p)$ satisfies R1.2.

From R2.1 (0Si<s) for $(\Phi, r, p)$ and the definition of $r$ follows that $(\Psi, r, p)$ satisfies R2.1.

From the restrictions on the pass-delay counter $p_l$ and {from the definitions of $p$, $\Lambda$, and $q$}:

\[(p_{1}, s > 0) \Rightarrow (p_{1}, s = p_{l}, s) \wedge (A_{1} : x \in \text{p}_{n} s : A_{1} x \text{p}_{n} s = \Gamma_{1} s) \wedge (q_{1} = Q_{1}, s))\]

follows that $(\Psi, r, p)$ also satisfies these restrictions.

The final results for $\Psi$ can now be achieved easily.

From $(A_i : 0 \leq i < \#_i^\Psi : \text{next} 3 A \gamma (\Psi_i, s_i, p_i) \cdot (\Psi_{i+1}, s_{i+1}, p_{i+1}))$ follows:

\[FSLA \gamma^\Psi .\]

From the definition of $p$ immediately follows:

\[(A_i : 0 \leq i < \#_i^\Psi : (p_{i}, s > 0) \Rightarrow (TV : A_{i}, s \neq (L, H)) .\]

From 0.5 $(CFL , \Psi , (\Psi , r , p) = (\Phi , r , p))$ follows directly:

\[\Psi \in (CFL \cap cgdTL \cap FCAO \cap csL) \Rightarrow (\Phi \in (CFL \cap cgdTL \cap FCAO \cap csL)) .\]

**STEP 1**

For $0 \leq n$ and $0 \leq i < \#_i^\Psi$ we define $p_{c} \in \text{DC}$ and $\Delta(n), i \in \text{NSTC}$ by:

\[p_{c} : s = \begin{cases} 0 & \text{if TV : A} : s \in \mathcal{R}([I, H]) \\ p_{i} : s & \text{if TV : A} : s \notin \mathcal{R}([I, H]) \end{cases} .\]

\[\Delta(0) : x = \begin{cases} A_{1} : x & \text{if } A_{1} x \neq \emptyset \\ \{ I, H \} & \text{if } A_{1} x = \emptyset \end{cases} .\]

\[\Delta(n+1) = R (\gamma \text{store} : \Delta(n)) \cdot (q_{i} \wedge Z : p_{c}) .\]

The following properties hold for all $n$ and $i$ satisfying $0 \leq n$ and $0 \leq i < \#_i^\Psi$:

1.0 \text{ destore} : \Delta(n) = \text{destore} : \Delta ;

1.1a \quad \Lambda_{1} \subseteq \Delta(0);

1.1b \quad \Delta(n), i \subseteq \Delta(n+1);

Property 1.0 for $n=0$ and property 1.1a follow directly from the definition of $\Delta(0)$.

Properties 1.0 for $n>0$ and 1.1b are proven below.

**Proof of 1.0**

Let $n \geq 0$. Then:

\[\text{destore} : \Delta(n+1) ;\]

\[= (\text{def. } \Delta(n+1), i) \]

\[\text{destore} : (R (\gamma \text{store} : \Delta(n)) \cdot (q_{i} \wedge Z : p_{c})) .\]
\[ \{ \text{lemma 4.20a, prop. 4.18a,b} \} \\
R \cdot \gamma(q_i \land Z \cdot pc_1) \]
\[ = \{ \text{notes 0 and 1} \} \\
R \cdot \gamma(q_i \land Z \cdot p_1) \]
\[ = \{ \text{lemma 4.20a, prop. 4.18a,b} \} \\
destore \cdot \langle R \cdot (\gamma \uplus \text{store} \cdot \Delta_i) \rangle \cdot (q_i \land Z \cdot p_1) \]
\[ = \{ \text{R1.2 for } (\Psi, x, p) \} \\
destore \cdot \Delta_i \]

**note 0**

true

\[ = \{ \text{def. of pc}_1 \} \\
Z \cdot p_1 \preceq Z \cdot pc_1 \]

\[ \Rightarrow \{ \text{monotonicity } R \cdot \gamma \} \\
R \cdot \gamma(q_i \land Z \cdot pc_1) \subseteq R \cdot \gamma(q_i \land Z \cdot p_1) \]

**note 1**

Let \( x \in \mathbb{N} \). Then:

\[ R \cdot \gamma(q_i \land Z \cdot pc_1) \cdot x \subseteq R \cdot \gamma(q_i \land Z \cdot p_1) \cdot x \]

\[ \Rightarrow \{ \text{definition 1.17 (R \cdot \gamma)} \} \\
(\exists y : y \neq \emptyset : \text{cp}(q_i \land Z \cdot p_1) \cdot \{ x, y \} \land \lnot \text{cp}(q_i \land Z \cdot p_1) \cdot \{ x, y \} ) \]

\[ \Rightarrow \{ \text{definition 1.14 (cp) and calculus} \} \\
(\exists y, y', x : y \neq \emptyset \\
: \text{cp}(q_i \land Z \cdot p_1) \cdot \{ x, y \} \land \text{cp}(q_i \land Z \cdot p_1) \cdot \{ x, y', x \} \land (q_i \land Z \cdot pc_1) \cdot y \land \lnot (q_i \land Z \cdot p_1) \cdot y' ) \]

\[ \Rightarrow \{ \text{from above: destore} \cdot \Delta_i = R \cdot \gamma(q_i \land Z \cdot p_1) \} \\
(\exists y, z : z \in \text{pan} \cdot s : (\Delta_i \cdot z \in \mathcal{P}((l, h))) \land Z \cdot pc_1 \cdot s \land \lnot Z \cdot p_1 \cdot s ) \]

\[ \Rightarrow \{ \text{definition pc}_1 \} \\
\text{false} \]

\[ \Box \text{ (end proof 1.0)} \]

**Proof of 1.1b**

The proof uses induction to \( n \).

Basis of the induction:

true

\[ \Rightarrow \{ \text{definition of pc}_1 \} \\
Z \cdot p_1 \preceq Z \cdot pc_1 \]

\[ \Rightarrow \{ \text{monotonicity of } R \} \\
R \cdot \gamma \uplus \text{store} \cdot \Delta(0)_i \cdot (q_i \land Z \cdot p_1) \subseteq R \cdot (\gamma \uplus \text{store} \cdot \Delta(0)_i) \cdot (q_i \land Z \cdot pc_1) \]

\[ = \{ \text{note 2 below and definition of } \Delta(1)_i \} \\
\Delta(0)_i \subseteq \Delta(1)_i \]
Step of the induction:
\[ \Delta(n)_{=1} \subseteq \Delta(n+1)_{=1} \]
\[ \Rightarrow \{ \text{def. 4.4 (store), using (from 1.0): destore.\(\Delta(n)_{=1} = \text{destore.} \Delta(n+1)_{=1} \) \}
\]
\[ \text{store.}\Delta(n)_{=1} \subseteq \text{store.}\Delta(n+1)_{=1} \]
\[ \Rightarrow \{ \text{def. 4.6 } (R) \}
\]
\[ R \langle \gamma \equiv \text{store.}\Delta(n+1)_{=1} \cdot \langle q_1 \wedge Z \cdot p_0 \rangle \rangle \subseteq R \langle \gamma \equiv \text{store.}\Delta(n+1)_{=1} \cdot \langle q_1 \wedge Z \cdot p_0 \rangle \rangle \]
\[ \Rightarrow \{ \text{definitions of } \Delta(n+1)_{=1} \text{ and } \Delta(n+2)_{=1} \}
\]
\[ \Delta(n+1)_{=1} \subseteq \Delta(n+2)_{=1} \]

**note 2**

true

\[ \Rightarrow \{ \text{R1.2 for } (\Psi,x,p) \text{, note 3, and def. of } \Delta(0)_{=1} \}
\]
\[ (A_1 = R \langle \gamma \equiv \text{store.}\Lambda_{=1} \cdot \langle q_1 \wedge Z \cdot p_1 \rangle \rangle)
\]
\[ \wedge (A \cdot x : A_1 \cdot x = \Phi : \text{store.}\Delta(0)_{=1} \cdot x = \{i,j\} \)
\]
\[ \wedge (A \cdot x : A_1 \cdot x = \Phi : \text{store.}\Delta(0)_{=1} \cdot x = \text{store.}\Lambda_{=1} \cdot x) \]
\[ \Rightarrow \{ \text{def. of } \Delta(0)_{=1} \text{, def. 4.6 (R), and calculus} \}
\]
\[ \Delta(0)_{=1} = R \langle \gamma \equiv \text{store.}\Delta(0)_{=1} \cdot \langle q_1 \wedge Z \cdot p_1 \rangle \rangle \]

**note 3**

true

\[ \Rightarrow \{ \text{R1.2 for } (\Psi,x,p) \}
\]
\[ (A_1 = 0 \leq i < \# \cdot \Psi \cdot A_1 = R \langle \gamma \equiv \text{store.}\Lambda_{=1} \cdot \langle q_1 \wedge Z \cdot p_1 \rangle \rangle) \]
\[ \Rightarrow \{ \text{def. } R \}
\]
\[ (A_1 = 0 \leq i < \# \cdot \Psi \cdot \text{store.}\Lambda_{=1} \subseteq A_1) \]
\[ \Rightarrow \{ \text{def. 4.4 (store)} \}
\]
\[ (A \cdot i,x : (0 \leq i < \# \cdot \Psi) \wedge N \cdot x : (A_1 \cdot x = \Phi) \Rightarrow (A_1 \cdot x = \Phi) \)
\]
\[ \Rightarrow \{ \text{calculus} \}
\]
\[ (A \cdot i,x : (0 \leq i < \# \cdot \Psi) \wedge (0 \leq j < \# \cdot \Psi) \wedge N \cdot x : (A_1 \cdot x = \Phi) \Rightarrow (A_1 \cdot x = \Phi) \)
\]
\[ \square \text{ (end proof 1.1b) } \]

From 1.1b and the finiteness of NSTC follows that a natural, say M, exists such that:
\[ (A_1 = 0 \leq i < \# \cdot \Psi : \Delta(M)_{=1} = \Delta(M+1)_{=1} ) \]

From 1.1a,b follows:

1.1c \quad (A_1 = 0 \leq i < \# \cdot \Psi : A_1 \subseteq \Delta(M)_{=1} ) .

Let \( \Omega \in L^* (ST_1) \) be defined by:
\[ (\# \cdot \Omega = \# \cdot \Psi) \wedge (A_1 = 0 \leq i < \# \cdot \Omega : \Omega = (q_1 \cdot \Delta(M)_{=1} ) ) .
\]
For $0 < \# : \Omega$ the following holds.

1.2 Let $x \in N$. Then:
   \[ \Lambda_1 : x = \emptyset \]
   \[ \vdash \{ \text{def. } \Delta(0) \text{ and 1.0} \} \]
   \[ (\Delta(0)_1 : x = \{l, h\}) \land (\Delta(M)_1 : x \notin \mathcal{P}([l, h])) \]
   \[ \vdash \{1.1b\} \]
   \[ \Delta(M)_1 : x = \{l, h\} \]

1.3 consistent0-$\Lambda_1$
   \[ = \{ \text{def. 4.5 (consistent0)} \} \]
   consistent0-$($destore-$\Lambda_1$)$
   \[ = \{1.0\} \]
   consistent0-$($destore-$\Delta(M)_3$)$
   \[ = \{ \text{def. 4.5 (consistent0)} \} \]
   consistent0-$\Delta(M)_1$

1.4 Let $s \in SW$. Then:
   \[ \neg \mathcal{A}_1 : s \]
   \[ \vdash \{ \text{def. 4.3 (CA), 1.0, 1.1c, and 1.2} \} \]
   \[ \neg \mathcal{A}_p : s \]

1.5 Let $s \in SW$. Then:
   \[ \mathcal{A}_1 : s \]
   \[ \vdash \{ \text{prop. 4.19a} \} \]
   \[ \neg \mathcal{A}_p : s \]
   \[ \vdash \{ \text{prop. 4.19b} \} \]
   consistent0-$\Lambda_1 : q_{a+1} : s$
   \[ = \{1.3\} \]
   consistent0-$\Delta(M)_1 : q_{a+1} : s$

1.6 Let $s \in SW$. Then:
   \[ p_0 : s > 0 \]
   \[ \vdash \{ \text{def. } p_0 \} \]
   \[ (p_1 : s > 0) \land (TV : \Lambda_1 : s \notin \mathcal{P}([l, h])) \]
   \[ \vdash \{ \text{result part a} \} \]
   \[ (TV : \Lambda_1 : s \neq \{L, H\}) \land (TV : \Lambda_1 : s \notin \mathcal{P}([l, h])) \]
   \[ \vdash \{1.0\} \]
   \[ (TV : \Delta(M)_1 : s \neq \{L, H\}) \land (TV : \Delta(M)_1 : s \notin \mathcal{P}([l, h])) \]
   \[ = \{ \text{calculus} \} \]
   \[ TV : \Delta(M)_1 : s \in \{\{L\}, \{H\}\} \]
1.7 Let $s \in SW$. Then:

\[ pc_i,s > 0 \]

$\rightarrow$ (1.6 and def. RR0 (in introduction this appendix))

\[ \{ E \ x : x \in pan-s : E(M) \ x \in \{ x,y \} \} \land \{ A(x,y) : pan-s = [x,y] : A(x \neq A,y) \} \]

$\rightarrow$ (1.0)

\[ \{ A(x,y) : pan-s = [x,y] : E(M) \ x \neq E(M)y \} \]

1.8 Let $s \in SW$. Then:

\[ pc_i,s > 0 \land pc_{i-1},s > 0 \]

$\rightarrow$ (def. of pc)

\[ pc_i,s = p_{1},s \land pc_{i-1},s = p_{1-1},s \land (p_{1},s > 0) \land (p_{1-1},s > 0) \]

$\rightarrow$ (def. RR1 (in introduction this appendix))

\[ pc_i,s = pc_{i-1},s - 1 \]

1.9 Let $s \in SW$. Then:

\[ pc_i,s > 0 \land (pc_{i-1},s = 0) \]

$\rightarrow$ (def. of pc)

\[ pc_i,s = p_{1},s \land (p_{1},s > 0) \land (p_{1-1},s > 0) \lor (TV \cdot A_i \cdot y \in R(i,\text{is})) \]

$\rightarrow$ (def. RR2 (in introduction this appendix), 1.6, and 1.0)

\[ \neg q_{i-1},s \lor (TV \cdot A_i \cdot y \notin TV \cdot A_i \cdot z,s) \]

Using that (from part a) \((A : i : 0s < \# \Psi : next3\alpha \gamma(\Psi_{i},t_{i},p_{i}) \cdot (\Psi_{i+1},t_{i+1},p_{i+1}) )\), we will now prove that \((A : i : 0s < \# \Omega : next3 \alpha \gamma(\Omega_{i},t_{i},p_{i}) \cdot (\Omega_{i+1},t_{i+1},p_{i+1}) )\).

From 1.3, 1.4, 1.5, and RR0 for \((\Psi,t,p)\) follows that \((\Omega,t,p)\) satisfies R0.i.

From the definitions of \(E(M)\) and M follows directly that \((\Omega,t,p)\) satisfies R1.2.

From 1.3 above and RR0 for \((\Psi,t,p)\) follows that \((\Omega,t,p)\) satisfies R2.i.

From RR0, RR1, and RR2 (defined in the introduction this appendix) for \((\Psi,t,p)\) and 1.6, 1.7, 1.8, and 1.9 follows that \((\Omega,t,p)\) satisfies R3.1, R3.0 and R3.2.

The final results for \(\Omega\) can now be achieved easily.

From \((A : i : 0s < \# \Omega : next3 \alpha \gamma(\Omega_{i},t_{i},p_{i}) \cdot (\Omega_{i+1},t_{i+1},p_{i+1}) )\) follows:

\[ FSL_{\alpha} \gamma \Omega \]

From 1.0, 1.3, and the definition of \(pc_{i}\) follows directly:

\[ (\Omega \in (CFL \cap cgdl \cap FCA0 \cap csl) ) \land (\Psi \in (CFL \cap cgdl \cap FCA0 \cap csl) ) \]

$\Box$ (end part b)
PART c
In parts a and b equality of the correctness criterion for initial behaviour in the model of ch. 5 and that in the alternative model are proven. Let the initial behaviour be correct. In section 5.0 we have proven that the resulting states in the model from ch. 5 all are — abstracted from stored values — equal to γ s (ch. 5.12). Using the constructions from part b, 2.0 below shows that this also holds for the resulting states in the alternative model. It is easily seen that this means that all resulting states are stable. Evidently, stable3A·γ = stable3.

2.0 Assume WM3·Cγ ∧ CF·γ s ∧ FSL3A·γ·Φ. Let Ψ and Ω be as defined in part b. 

true
⇒ {assumption and FSL3·γ·Ω, using th. 5.12}
(A i : 0 ≤ t · #Ω : destore·Ωi = γ s ) ∧ CFL·Ω
⇒ {definition of Ω and 1.0}
(A i : 0 ≤ t · #Ψ : destore·Ψi = γ s ) ∧ CFL·Ψ
⇒ {definition of Ψ and 0.5}
(A i : 0 ≤ t · #Φ : destore·Φi = γ s ) ∧ CFL·Φ

PART d
In part d we will explain why the straightforward definition of next3A α,pd (i.e., a similar extension of next3A as in chapter 5 for defining next3A pd ) does not capture the notion of restricted pass-delays we want to model. Furthermore, we will argue — informally — that 3.0 below holds. The argument given can be formalised in a similar way as the proof in part b.

3.0 (A C γ D (Π,rc,pc) : = W F 3 P·Cγ·(Π,rc,pc) ⇒ W F 3 A P·C γ·(Π,rc,pc) )

Consider the circuit depicted in figure C0 below. Let the starting state be the top-most state in fig. C1 and let the new source-connection be \{(x0, L1), (x1, δ), (x2, δ), (y, H)\} . Let the delays be restricted by: (rd s1 = 0) ∧ (pd s0 = 2) ∧ (pd s1 = 1). The resulting state list in the model from ch.5 is given in fig. C1 (where for each switch the switch-state (left-most number) and the pass-delay counter are given). Using the straightforward extension for next3A α,pd , the resulting state list in the alternative model is the one given in figure C2. In this list the pass-delay for the low source value \{L\} is 0 in switch s1. Note that it is intended to be 1.
In order to model such pass-delays properly, the restrictions RR1 and RR2 must be adjusted. Note that such an adjustment is not required in the previous parts since the pass delays considered there are assumed to be arbitrary.

Let $C$, $\gamma$, $D$, and $(Tl,r_p,pc)$ be such that $WF3_{A_{II}}:C:1\gamma(Tl,r_p,pc)$ . Let $(rd,pd) \in D$ and $\Phi$, $\Psi$ be such that $RSL3_{A_{II}}:pdp:1\gamma(Tl,r_p,pc) : \Phi \land RSL3_{A_{II}}:pdp:1\gamma(Tl,r_p,pc) : \Psi$ . Note that from lemma 5.26a follows that $\Psi$ is in the model from ch.5, the only resulting state list for $(rd,pd)$.

Let $\Phi_1 = (Q_1,\Gamma_1)$ and $\Psi_1 = (Q_1,\Delta_1)$ . Then:

1. $cgdtd$ \quad $\Pi$(def. 4.10) 
   $\Rightarrow$ \quad $\Pi = (Q_0,\Gamma_0)$ , and $\Pi = (Q_0,\Delta_0)$ )
   
   $(A s : SW \cdot s : (\Gamma_0(g,s) \neq \emptyset) \land (\Delta_0(g,s) \neq \emptyset))$
   
   $\Rightarrow$ \quad (R1.2 for $\Phi$ and $\Psi$ )
   
   $(A i_s : SW \cdot s : (\Gamma_1(g,s) \neq \emptyset) \land (\Delta_1(g,s) \neq \emptyset))$

Consequently, disallowing pass-delays in cases where $TV : \Gamma_1 x \in ([L,H] \cup \tau(I,L,H))$ can change the next node-state in two ways:

- $\alpha$ a source-conflict occurs, i.e. $(E x : (\Delta_1 x = [L,H]) \land (\Gamma_1 x \neq [L,H]))$
- $\beta$ a charge-conflict occurs, i.e. $(E x : (\Delta_1 x = [I,L]) \land (\Gamma_1 x \neq [I,L]))$

In case $\beta$, if the node is a gate node, $\neg cgdtd : \Psi$ holds. Since (use assumption) $CFL : \Psi \land cgdtd : \Psi$ , case $\alpha$ does not occur and case $\beta$ does not occur at gate nodes.

Consequently: $destore : \Gamma_1 = destore : \Delta_1$ \quad $(A s : SW \cdot s : (\Gamma_1 (g,s) = \Delta_1 (g,s)) \land (q_1 s = Q_1 s))$ .

This implies: $(\Psi \in (cgdl : \Psi \land FCA1 \land CFL)) \Rightarrow (\Phi \in (cgdl : \Psi \land FCA1 \land CFL))$

For the other correctness criterion (cstil) the argument is similar.

We can now conclude $WF3_{A_{II}}:C:1\gamma(Tl,r_p,pc)$ .

$\square$ (end part d)
REFERENCES


REFERENCES


REFERENCES


NOTATION INDEX

The page numbers refer to the definitions of the symbols, sets, and functions. Local notations from appendices B and C are not referred to. Restrictions Ri,j are listed at the end of this index.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page</th>
<th>Description</th>
<th>Page, Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>→</td>
<td>14</td>
<td>bcp, cp</td>
<td>26</td>
</tr>
<tr>
<td>·</td>
<td>14</td>
<td>consistent, cocol</td>
<td>24, 81</td>
</tr>
<tr>
<td>$\mathcal{B}$</td>
<td>14</td>
<td>$R \in \text{NST} \rightarrow \text{SST} \rightarrow \text{NST}$</td>
<td>27, 77</td>
</tr>
<tr>
<td>$\mathcal{R}(X)$</td>
<td>14</td>
<td>stable</td>
<td>28</td>
</tr>
<tr>
<td>$\mathcal{L}(A)$</td>
<td>15</td>
<td>next0</td>
<td>29</td>
</tr>
<tr>
<td>$\mathcal{L}^+(A)$</td>
<td>15</td>
<td>(next0)$^+$, feasible0</td>
<td>30</td>
</tr>
<tr>
<td>#-$\mathcal{L}$</td>
<td>15</td>
<td>WM0</td>
<td>31</td>
</tr>
<tr>
<td>$L_i$</td>
<td>15</td>
<td>SF</td>
<td>33</td>
</tr>
<tr>
<td>cat-$\mathcal{L}0$-$\mathcal{L}1$</td>
<td>15</td>
<td>G0 $\in$ SF</td>
<td>33</td>
</tr>
<tr>
<td>*[L]</td>
<td>15</td>
<td>$\gamma^<em>, \gamma^</em>$</td>
<td>34</td>
</tr>
<tr>
<td>(A l : R : E)</td>
<td>15</td>
<td>next0G0, feasibleG0</td>
<td>35</td>
</tr>
<tr>
<td>(B l : R : E)</td>
<td>15</td>
<td>G1 $\in$ SF</td>
<td>38</td>
</tr>
<tr>
<td>($\omega l : R : E$)</td>
<td>15</td>
<td>$\gamma^1$, $\gamma^2$</td>
<td>39</td>
</tr>
<tr>
<td>(n l : R : E)</td>
<td>15</td>
<td>next0G1, feasibleG1</td>
<td>39</td>
</tr>
<tr>
<td>$C = (N, SW, t, g, pn), \text{CIR}$</td>
<td>18</td>
<td>CF</td>
<td>41, 91</td>
</tr>
<tr>
<td>${L, H}$</td>
<td>18</td>
<td>cst0</td>
<td>42, 91</td>
</tr>
<tr>
<td>t $\in$ SW - ${L, H}$</td>
<td>18</td>
<td>bni0, nio, si0</td>
<td>46</td>
</tr>
<tr>
<td>g $\in$ SW + N</td>
<td>18</td>
<td>A0 $\in$ CIR - $\mathcal{B}$</td>
<td>46</td>
</tr>
<tr>
<td>pn $\in$ SW + $\mathcal{B}2N$</td>
<td>18</td>
<td>PC, bni1, nii, si1</td>
<td>47</td>
</tr>
<tr>
<td>$B2N$, $[x, y]$</td>
<td>18</td>
<td>AI $\in$ CIR - $\mathcal{B}$</td>
<td>47</td>
</tr>
<tr>
<td>NST = N - $\mathcal{R}(L, H)$</td>
<td>20</td>
<td>$\xi$</td>
<td></td>
</tr>
<tr>
<td>(NST, $\xi$)</td>
<td>20</td>
<td>DC, $\emptyset$</td>
<td>59</td>
</tr>
<tr>
<td>$T_i$</td>
<td>20, 173</td>
<td>nexti</td>
<td>59</td>
</tr>
<tr>
<td>SST = SW - $\mathcal{B}$</td>
<td>21</td>
<td>stable1, feasible1</td>
<td>60</td>
</tr>
<tr>
<td>(SST, $\xi$)</td>
<td>21</td>
<td>WM1</td>
<td>60</td>
</tr>
<tr>
<td>$T_i, F$</td>
<td>21</td>
<td>nexti$_{nd}$</td>
<td>66</td>
</tr>
<tr>
<td>ST0 = SSTxNST</td>
<td>22</td>
<td>RD, stable$^{nd}$</td>
<td>67</td>
</tr>
<tr>
<td>(ST0, $\xi$)</td>
<td>22</td>
<td>feasible$^{nd}$, WM1$^{nd}$</td>
<td>67</td>
</tr>
<tr>
<td>gcd0, cgdc0</td>
<td>24, 79</td>
<td>rdH, rdL</td>
<td>67</td>
</tr>
<tr>
<td>consistent0, coc0</td>
<td>24, 77</td>
<td>CA</td>
<td>75, 76</td>
</tr>
</tbody>
</table>
NOTATION INDEX

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page(s)</th>
<th>Description</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I,A)</td>
<td>75, 76</td>
<td>(MD, ≠), (MUT, ≤)</td>
<td>143</td>
</tr>
<tr>
<td>ast . NSTC, $\vdash$, $\subseteq$, $\subseteq$</td>
<td>76</td>
<td>(1@), mp, bpq, pq</td>
<td>144</td>
</tr>
<tr>
<td>ST1 = SST x NSTC</td>
<td>76</td>
<td>$\kappa \in$ MUT = MUT</td>
<td>145</td>
</tr>
<tr>
<td>(ST1, ⊆)</td>
<td>76</td>
<td>mp1</td>
<td>150</td>
</tr>
<tr>
<td>store, destore</td>
<td>77</td>
<td>ML(u)</td>
<td>151</td>
</tr>
<tr>
<td>next2</td>
<td>78</td>
<td>RSL4 . WF4 . MLL(u)</td>
<td>152</td>
</tr>
<tr>
<td>stable2, feasible2</td>
<td>79</td>
<td>(γ, imd), md'</td>
<td>155</td>
</tr>
<tr>
<td>gdl, cgdl</td>
<td>79</td>
<td>in, out, plx, phs, local</td>
<td>159</td>
</tr>
<tr>
<td>FSL2</td>
<td>79</td>
<td>GLOBAL, σ, γ(α)</td>
<td>160</td>
</tr>
<tr>
<td>FCA0, cgdlL</td>
<td>80</td>
<td>sof(G,α), sof(α)</td>
<td>160</td>
</tr>
<tr>
<td>WM2</td>
<td>81</td>
<td>OAS</td>
<td>161, 163</td>
</tr>
<tr>
<td>next2, Td</td>
<td>92</td>
<td>PO, POL</td>
<td>161, 163</td>
</tr>
<tr>
<td>RSL2, Td</td>
<td>93</td>
<td>RS_D</td>
<td>162</td>
</tr>
<tr>
<td>CFL, FCA1, cs0L</td>
<td>93</td>
<td>SOB</td>
<td>163</td>
</tr>
<tr>
<td>WF2, Td</td>
<td>93</td>
<td>join, meet</td>
<td>173</td>
</tr>
<tr>
<td>FCL</td>
<td>94</td>
<td>lub, sup, glb, inf</td>
<td>173</td>
</tr>
<tr>
<td>pc</td>
<td>107</td>
<td>fix</td>
<td>173</td>
</tr>
<tr>
<td>TV</td>
<td>111, 112</td>
<td>lfp, gfp</td>
<td>174</td>
</tr>
<tr>
<td>next2, Z</td>
<td>112</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stable3, feasible3</td>
<td>112</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FSL3, WM3</td>
<td>113</td>
<td></td>
<td></td>
</tr>
<tr>
<td>csst, csst</td>
<td>117</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathcal{N}(\mathcal{P}_d)$</td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>next3, Td</td>
<td>123</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D, stable3_D, feasible3_D</td>
<td>125</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FSL3_D, WM3_D</td>
<td>125</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSL3, Td</td>
<td>125</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WF3, D</td>
<td>126</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wd</td>
<td>131</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(N,SW,W,t,g,pn)</td>
<td>132</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XCIR, XSSST, ST2</td>
<td>132</td>
<td></td>
<td></td>
</tr>
<tr>
<td>consistent2, cs2</td>
<td>133</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_L$, $V_R$, $V$</td>
<td>138</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MD, m, α</td>
<td>140, 143</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mdH</td>
<td>141, 147, 150</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The restrictions on state transitions:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R0.0, R0.1, R1.0</td>
<td>58</td>
</tr>
<tr>
<td>R2.0, R2.1, R2.2</td>
<td>58</td>
</tr>
<tr>
<td>R2.2</td>
<td>65</td>
</tr>
<tr>
<td>R2.2'</td>
<td>66</td>
</tr>
<tr>
<td>R0.2, R0.3, R1.1</td>
<td>78</td>
</tr>
<tr>
<td>R1.2</td>
<td>108</td>
</tr>
<tr>
<td>R3.0, R3.1, R3.2</td>
<td>110</td>
</tr>
<tr>
<td>r3.0, r3.1, r3.2</td>
<td>111</td>
</tr>
<tr>
<td>R3.2'</td>
<td>118</td>
</tr>
</tbody>
</table>
SUBJECT INDEX

Page numbers given in boldface refer to definitions. For most of the generally used notions, only the definitions are referred to.

abstractions of reality; see restrictions (a)-(e) 1, 5, 169
active capacitance 75, 76
active delays 55, 106
cylic circuits 11, 45-55
cylicity (general) 46, 52, 54, 55
cylicity (w.r.t. source-connection) 47, 48, 52-53
anti-monotonic 172
anti-symmetric 171
assumptions on charge storage 74, 102
assumptions on pass-delays 108, 109, 134
assumptions on reality; see restrictions (a)-(e) 134
arbitrary pass-delays 106-117
arbitrary reaction-delays 57, 64
asynchronous network 2, 4
basic model 11, 17-44
Boolean values 14
bottom element 173

capacitances 9, 72-103
capacitance-strength 75, 98, 99
circuit 18
circuit states (STD) 22 ; (STI) 76
chain 172
circuit sharing 1, 74
circuit storage 9-11, 17, 71, 73-103
closures of relations 171
componentwise order 172
conducting path 25, 26
conflict/conflicting 5, 6, 20
conflict-freeness 41, 64, 91, 93
conservative model; see pessimistic model 108
consistent (0) 7, 23-24, 77 ; (1) 7, 23-24, 26, 81
continuous function 173
continuous transistor model 3-5, 169
correctness criteria 6, 11, 17, 43, 44, 70, 116-117, 151, 161, 165, 167-168
correctness state transition (0) 41, 42, 64, 91, 93, 117 ; (1) 116, 117
countdown function 58, 59, 71, 107, 169
delay counter, set of 59
delays 8 ; see reaction-delays, pass-delays, and wire-delays
design method 1, 5, 169
discrete switch-level model 3, 12-14, 167 169
dual order 172
dynamic behaviour 5, 70, 163, 135
feasible (0) 30, (35) ; (1) 60, 67 ; (2) 79 ; (3) 112, 125
feasible state list (2) 79 ; (3) 113, 125
fight; see conflict
finite line capacitance control 78, 80, 93
finite line delays; see countdown function
fixpoint 173
fixpoint theorems 174
flip-flop; see latch
floating 6, 20
function application (notation) 14
function sets (notation) 14
fundamental mode 2, 4, 12-14, 164-166
gate defined (0) 5, 24, 79 ; (1) 79, 80
gate node 18
gate-level model 2-4, 169
global state, set of 160
global state transition diagram 160
greatest element 172
greatest lower bound 173
Hasse diagram 172
hazards 4, 5, 163

imperfectness of switches 1, 5, 11, 137-157
inconsistency of switches; see consistent
informant 173
inital behaviour 5, 43, 71, 102-103, 135, 168
input function 160
input-output mode 2, 4, 164, 165
Knaster-Tarski, fixpoint theorem of 174
latch 55, 93, 90, 153
lattice 173
Lattice Theory 171-174
layout-level 1, 5, 169
least element 172
least upper bound 173
linear order 172
lists 15
maximal (maximum) element 172
minimal (minimum) element 172
model-reality relation 0, 3, 4, 8, 11, 167-169
monosonic 172
multiple capacitance-strengths 98, 99
multiple threshold voltages 156
mitigating input signals 155
mitigating signals 139, 155
mitigating power 137-157, 144, 150
mitigation degree 11, 137-157, 147, 150, 160
mitigation degree, domain of 143
mitigation functions, set of 143
mitigation limit 151, 157

natural numbers 14
network of components 1, 2
network of transistors 3
next-state function (0) 28-29, 35; (1) 57-59, 66;
(2) 77, 78, 92; (3) 107-112, 123
nodes, set of 18
node-state, set of 19, 20, 76
notational conventions 14-16

optimistic model 3
order 171
oscillating 5, 19; see feasible
output as specified 164-165

partially ordered set 171
partial order 171
partition of the node set 159
pass-delay 5, 103-135
pass node 18
path quality 142-144
perfect outputs 161-163
perfect switch 10, 17, 137, 156
pessimistic model 3, 168
pointwise order 172
poset 171
power set 14
priority of operators 15
proof denotation 15, 16
pull-down resistance 9, 100
pull-up resistance 9, 100, 101
quantification (notation) 15
reaction delay 57-71
reflexive and transitive closure 171
reflexivity 171
relative timing/delays 1, 8; see delays
resistances 9-10, 100-101
response function 21, 77
restrictions (a) - (6) 10, 11, 17, 167
restrictions on pass-delays 117-131
restrictions on reaction-delays 64-71
restrictions on state transitions;
see notation index
resulting state 5, 20, 28-31; see feasible
resulting state list (2) 93; (2) 126; (4) 152
run pulse 4, 41-42, 163

short circuit; see conflict
smooth output behaviour 163
source-connections 7, 18, 19
sources 6, 19
specification 1, 2, 4, 11, 159-166, 160
specified output function 160
spike; see run pulse
stable 5; (0) 19, 28 (35); (1) 60, 67;
(2) 79; (3) 112, 125
starting states 65, 70, 164
states 7, 8, 22
state transition; see next-state function
stored charges 76
successive approximation 174
supremum 173
switch 6, 7, 10, 18
switches, set of 18
switch function 33-40
switch gate 18
switch-level model 3, 167-169
switch pass node pair 18
switch-state 21
switch type 6, 18
synchronous network 2, 4

threshold voltages 137-138, 157
top element 173
transistor model; see switch-level model
transistor network 1
transitive closure 171
transitivity 171
transporting 106
trickle inverter 101

undefined gates 1, 7, 21; see gate defined
unknown state 12-14, 21

well-functioning 6, 64, 70-71, 92-97;
(2) 93; (3) 127; (4) 152
well-matchedness 6, 116; (0) 22-23, 30-31;
(1) 60, 67; (2) 81; (3) 113, 125
wire delay 8, 131-135
SAMENVATTING

In dit proefschrift is een discreet switch-level model voor digitale CMOS circuits ontwikkeld. Het model beschrijft het logische gedrag zowel statisch als dynamisch - van transistor netwerken en van logische fouten ten gevolge van conflicts, imperfectie van switches, hazards, charge-sharing en relatieve vertragingen. Een discreet switch-level model is bedoeld als een tussenliggend model — op transistor niveau— tussen (discrete) hoger-niveau modellen en (continue) lager-niveau modellen. Het kan fungeren als semantisch domein voor hoger niveau modellen en calculi, als basis voor een switch-level simulator of een switch-level ontwerpmethode, of als hulpmiddel bij de vertaalslag van een switch-level ontwerp naar layout level.

Enerzijds worden de fysische aspecten van CMOS circuits bestudeerd en worden de begrippen geïntroduceerd die nodig zijn om (1) de fysische (in)correctheid van circuitgedrag te beschrijven, of (2) de relatie tussen gecodeerde gedrag en gespecificeerd gedrag te formaliseren. Anderezijds, nadat deze begrippen zijn ingevoerd, is het model puur wiskundig, en worden de eigenschappen van de begrippen en de relaties ertussen bestudeerd. We concentreren ons met name op de wiskundige nauwkeurigheid van het model, op de elegantie van de formele presentatie, en op de analyse van het model, en niet zoveel op directe "praktische" toepassingen zoals het ontwerp van een circuitimulator gebaseerd op het model.

Met betrekking tot de relatie tussen het model en de werkelijkheid is gestreefd naar een conservatief ( pessimistisch) model. Dat wil zeggen dat correctheid van gecodeerde gedrag ook correctheid van fysisch gedrag impliceert (maar niet noodzakelijk andersom). Uiteraard is gepoogd het pessimisme van het model zo gering mogelijk te laten zijn. De reden voor de keuze van een conservatieve aanpak is dat we in eerste instantie correctheid en niet incorrectheid van circuits willen kunnen concluderen op basis van het model.

In verband met de gewenste formalisering van de relatie met specificaties worden twee soorten gedrag bestudeerd. Dit zijn: (A) initieel gedrag, dat wil zeggen het gedrag van een circuit met stabiele invoerwaarden waarbij de vorige toestand van het circuit onbekend is, en (B) dynamisch gedrag, dat wil zeggen het gedrag van een circuit dat in een (bekende) toestand start wanneer de invoerwaarden veranderen. De twee centrale begrippen in het model met betrekking tot de correctheid van circuitgedrag zijn 'well-matchedness' en 'well-functioning'; de eerste is gerelateerd aan (A) en de tweede aan (B).
Vanwege de complexiteit van het fysisch gedrag van switch-level circuits gebruiken we in eerste instantie een aantal simplificerende annamen over en abstracties van dit gedrag. Nadat de op deze wijze "gesimplificeerde werkelijkheid" gemodelleerd is in een basismodel, worden stapsgewijs de annamen verzwakt en het model verfijnd. Deze aanpak stelt ons in staat om de verschillende aspecten van switch-level circuits in isolement te bestuderen en om grip te houden op de modellering. Bovendien is hierdoor een bestudering van de gevolgen van iedere verfijning mogelijk. Tengevolge van deze aanpak bevat dit proefschrift een hierarchisch geordende collectie van modellen.

Een belangrijk resultaat is dat de effecten van vertragingen en charge storage geen invloed hebben op de correctheid van initieel gedrag. Dat betekent dat bij de evaluatie van de correctheidscriteria voor initieel gedrag deze effecten buiten beschouwing kunnen worden gelaten en dat hiervoor dus het basismodel met de in hoofdstuk 6 gegeven uitbreiding kan worden gebruikt. In geval van correctheid van initieel gedrag is er, als we abstraheren van stored values, slechts één resulterende toestand, die dan uiteraard stabiel is. Het totale correctheidscriterium voor initieel gedrag, dat in het algemeen niet efficiënt te berekenen is, is voor acyclic circuits die gedefinieerd zijn in hoofdstuk 2, wel efficiënt te berekenen.

De formalisering van de effecten van imperfectheid van switches zoals gegeven in hoofdstuk 6 is verrassend eenvoudig en kan in de meeste switch-level modellen worden gebruikt, hetgeen dan leidt tot een algemener en nauwkeurigere beschrijving van circuitgedrag.

Zoals op verschillende plaatsen in het proefschrift is aangegeven (zie hoofdstuk 8 voor een overzicht), bezit het gepresenteerde model voldoende uitdrukkingskracht om generalisaties van en variaties op het model en de correctheidscriteria op eenvoudige wijze toe te laten.
CURRICULUM VITAE


Sinds augustus 1992 is hij als docent verbonden aan de Rijksuniversiteit te Leiden bij de vakgroep Informatica van de faculteit der Wiskunde en Natuurwetenschappen.
Statements

that go with the Ph.D. thesis

A Discrete Switch-Level Circuit Model

that uses 4-valued node states

by

Wilbert Körver

Eindhoven,
December 9, 1993
1
The formalization of the effects of imperfection of switches proposed in this thesis, viz. by using the notion of mutation degree, can be used in most existing discrete switch level circuit models, and leads to a more general and more accurate description of switch-level circuit behaviour.

lit:  This thesis, chapter 6.

2
Pass-delays and wire-delays cannot, in general, be modelled as part of reaction-delays.

lit:  This thesis, section 5.1.2.

3
When verifying whether the initial behaviour of a circuit satisfies the correctness criteria defined in this thesis, the effects of delays and charge storage need not be taken into account. The dynamic behaviour of circuits, however, cannot, even for small and simple circuits, be described in a satisfactory way without considering delays and charge storage effects.

lit:  This thesis.

4
In view of correctness considerations, it is a good design philosophy to take care that the designed transistor network is acyclic with respect to all allowed initializing inputs.

lit:  This thesis.

5
Let $R_0$ and $R_1$ be binary relations on set $X$, let $R_0^*$ and $R_1^*$ be the reflexive and transitive closures of $R_0$ and $R_1$ respectively, and let $P$ be a predicate on $X$ such that

$$(\forall x, y : x \in X \land y \in X : P \land R_0 \land R_0 \circ R_1 \land P)$$

Let $z$ be an element of $X$. Then:

$$(\forall y : R_0^* \circ z \circ y : P \circ y) \Rightarrow (\forall y : R_0^* \circ z \circ y : R_1^* \circ z \circ y).$$
At lower abstraction levels of circuits (e.g., switch-level), state-based arguments are much more convenient—in a sense more natural—than the transition-based (algorithmic) arguments used at higher levels. A change of design method, from transition-based to state-based, is, therefore, necessary in a design path of switch-level circuits. As demonstrated in [1], a calculus using production rule sets is a very suitable state-based design method.


7

The number of transistors, although often used as such, is not a good size measure for CMOS circuits.

lit: [1] above, section 1.4 'Costs measures for (C) MOS implementations'.

8

When modelling reality, the choice between pessimism or optimism of the model, which depends on the application in mind, should be made in as early a stage of the development of the model as possible.

9

Statements that go with Ph.D. theses often have the result that people form, completely unjustly, an opinion of a thesis without reading it. Statements should therefore be forbidden instead of required, and, if required, be formulated as cryptically as possible.

10

In contrast to statement 10, statement 9 is not a good example of statement 9 is not a good example of statement 9.