The Synchrony Hypothesis, Constructive Circuits and Timed Ternary Simulation

Michael Mendler

Faculty of Information Systems and Applied Computer Sciences
University of Bamberg, Germany
Overview

1. Introduction
2. When is a Circuit Combinational?
3. When is a Logic Constructive?
4. An Intuitionistic Modal Logic for Muller Automata (= inertial-delay circuit networks)
5. Constructive Muller Theories & Timed Ternary Simulation
6. Conclusion
1 INTRODUCTION
“A reactive system is faster than its environment, hence reactions can be considered atomic”

High-level Logical View
Reactions are
- discrete, atomic
- deterministic
- functional
- compositional

Low-level System Reality
Reactions may be
- continuous, non-atomic
- non-deterministic
- asynchronous
- causally entangled
The Grand Question

“A reactive system is faster than its environment, hence reactions can be considered atomic”

How to design & implement abstract system reactions so that

• they appear to operate in a functionally atomic way

• robustly and predictable,

• despite unavoidable low-level asynchrony with resource conflicts and scheduling uncertainties
This Class

“A reactive system is faster than its environment, hence reactions can be considered atomic”

We study some lessons learnt from the

- semantics of synchronous programming (e.g. Esterel)
- theory of asynchronous circuits

→ Object of Interest: Constructive Circuits [Gérard Berry 1999]
Background Literature

Asynchronous Hardware


Synchronous Programming

- G. Berry: Esterel de A à Z, https://www.college-de-france.fr/fr/agenda/cours/esterel-de-z
2

WHEN IS A CIRCUIT COMBINATIONAL?
A general (possibly cyclic) circuit is **combinational** if it realises a **functional relationship** stimulus \( \rightarrow \) response.

**Synthesising combinational circuits is non-trivial...**
Example: Romeo & Guilietta

Synchronous Programming Model

Hierarchical, communicating state machines, e.g.:

- Statecharts [D. Harel 1987]
- Esterel [G. Berry 1983, 2000]
- SyncCharts [Ch. André 2003]
- Quartz [K. Schneider, 2009]
- SCCharts [R. von Hanxleden et al. 2014]
- Céu [F. Sant’Anna et al, 2017]
- Blech [F. Gretz & F.-J. Grosch, 2018]
Example: Romeo & Guilietta

Boolean Declarative Semantics

\[
x = 1 \quad \text{state } x \text{ active}
\]

signal x present/emitted

transition x fired

\[
x = 0 \quad \text{state inactive}
\]

signal x absent/not emitted

transition x blocked

Logical Specification

\[
t2 = \neg \text{money}
\]

\[
t3 = \neg \text{money} \land \text{roses}
\]

\[
t4 = \text{kiss} \land \neg t5
\]

\[
t5 = \text{roses} \land \neg t4
\]

roses = t2

kiss = t3

money = t5
Example: Romeo & Guilietta

Generated Boolean Circuit

Logical Specification

\[
t_2 = \neg \text{money} \\
\neg \text{money} \land \text{roses} \\
\neg \text{money} \land \neg t_5 \\
\text{roses} \land \neg t_4
\]

 roses = t_2

kiss = t_3

money = t_5

M. Mendler, Bamberg University
Example: Romeo & Guilietta

Equivalent Circuit

Generated Boolean Circuit

But are these circuits really equivalent?

Unique Boolean Solution

Logical Specification

roses = 1
kiss = 1
money = 0

Boolean Simplification

\[ t2 = \neg \text{money} \]
\[ t3 = \neg \text{money} \land \text{roses} \]
\[ t4 = \text{kiss} \land \neg t5 \]
\[ t5 = \text{roses} \land \neg t4 \]

\[ \text{roses} = t2 \]
\[ \text{kiss} = t3 \]
\[ \text{money} = t5 \]
Combinational Networks

A network is DEL-combinational (in fundamental mode) if for all constant input signals every network node stabilizes in bounded time to a unique response value under DEL-execution semantics.

Refined Definition (operational)

Let DEL be a network delay/scheduling model. A network is DEL-combinational (in fundamental mode) if for all constant input signals every network node stabilizes in bounded time to a unique response value under DEL-execution semantics.
Some Delay Models

- **Fixed/up-bounded/bi-bounded Ideal Delay**  
  [e.g., Lam/Brayton’94]

- **UIN**: Up-bounded Inertial Delay  
  [Huffman’54, Miller’65, Brzozowski/Seger’89]

- **UNI**: Up-bounded Noninertial Delay  
  [“XBD0“, McGeer’92], [“binary chaos“, Burch’92]  
  [“delay-modality“, Mendler/Fairtlough’96]

- **Bi-bounded Inertial Delay**  
  [Brzozowski/Seger‘95]

Our focus
Up-bounded Inertial Delay (UIN)

Example
D=2

(1) **Up-bounded Propagation**: The delay cannot remain unstable for longer than $D$ time without changing output

(2) **Inertiality**: The output only changes if delay is unstable
**Example: Romeo & Guilietta**

*Muller Diagram* [Muller’56]

UIN system trajectories

---

**Up-bounded Inertial Delays (UIN)**

\[ \text{roses} := D \neg \text{money} \]
\[ \text{kiss} := D \neg \text{money} \land \text{roses} \]
\[ \text{money} := D \text{roses} \land \neg (\text{kiss} \land \neg \text{money}) \]

**General Multiple Winner Model (GMW)** [Huffman’54, Brzozowski/Yoeli 79]
Example: Romeo & Guilietta

\[ \text{Muller Diagram [Muller’56]} \]

UIN system trajectories

\[ \text{functional hazard: the network is not UIN-combinational!} \]

\[ \text{total state} \]

\[ x \, y \, z = \text{roses kiss money} \]

Oscillation!
Example Adjusted: No Delay in Kissing

Eliminate variable "kiss"

- Only two equations
- \( 1 \circ 2 \circ 3 \) evaluated atomically

\[
\begin{align*}
1^*1 & \quad \downarrow \\
01^* & \quad \downarrow \\
0^*0 & \quad \downarrow \\
10 & \\
\end{align*}
\]

total state \( x, z = \text{roses money} \)

\[
\begin{align*}
\text{roses} & := D \neg \text{money} \\
\text{money} & := D \text{roses} \land \neg ((\neg \text{money} \land \text{roses}) \land \neg \text{money})
\end{align*}
\]

The network is UIN-combinational!

- All UIN-trajectories converge

stable state!
Definition (informal)

A Boolean circuit is **DEL-insensitive**, if its behaviour is **invariant** under **arbitrary introduction** of DEL-delays in the gates‘ input and output wires.

**Question**

What would be an abstract specification language that

- extends **Boolean algebra** (in modest way)
- can **identify** **UIN-combinational/delay-insensitive circuits**
- is **expressive enough** to capture the effect of scheduling delays under causality and sharing („function hazards“)?
Interludio Logico

3

WHEN IS A LOGIC CONSTRUCTIVE?
1. **Aristotelian Truth:** Every sentence is either **true** or **false**
2. **Truth Functionality:** Truth of a composite sentence is a function of the truth value of its constituents

### Truth Table examples

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>( A \supset B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \neg B )</th>
<th>( A \land \neg B )</th>
<th>( (A \land \neg B) \supset F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

**Sherlock Holmes Principle:**

"If all contradictory scenarios have been excluded, what remains must be the truth"
Classical Logic is Reactively Inadequate

$$\text{love} \supset \text{roses}$$

constructively distinct

$$(\text{love} \land \neg \text{roses}) \supset \text{false}$$

classically equivalent
Omniscience of Classical Logic

From the classical “Principle of Omniscience“ the following is provable ...

\[ \vdash_{cl} \forall \text{Marriage} \in \text{Universe}. \]

\[ \exists \text{magic\_day} \in \text{Marriage}. \]

\[ (\text{love(} \text{magic\_day} \text{)} \supset \]

\[ \forall \text{day} \in \text{Marriage. love(day)}) \]

... yet, by all we know, constructively, this is nonsense!

[Bishop, Bridges: Constructive Analysis, Springer 1985]
Constructivity & Reactivity in System Design ...

\[ \psi_{AI, \vec{a}} = \text{set of all reactions of system AI under environment } \vec{a} \]

\[ s = \text{boolean output signal of AI system} \]

Assume \( \psi_{AI, \vec{a}} \) is constructive. Then ...

**Thesis**

Constructive Reactions (in constructive logic) are combinational & delay-insensitive!

- functionally determinate
- time-bounded
- stable, convergent, predictable, ...

\( s = 0 \) or \( s = 1 \)
4

MULLER LOGIC:
AN (INTUITIONISTIC) MODAL LOGIC
FOR MULLER AUTOMATA
The formulas $\Phi$ of Muller Logic are given by

\[
\phi \ ::= \ e \quad \text{boolean expression over wire variables } \mathcal{Z}.
\]

\[
\begin{align*}
| \quad & \phi \land \phi \quad \text{conjunction} \\
| \quad & \phi \lor \phi \quad \text{disjunction} \\
| \quad & \phi \Rightarrow \phi \quad \text{(intuitionistic) implication} \\
| \quad & \Diamond_D \phi \quad \text{bounded delay with } D \in \mathbb{R} \\
| \quad & \Box \phi \quad \text{inertiaillity}
\end{align*}
\]

The semantics is intuitionistic on time intervals ...
Muller Logic in a Nutshell

Definition

• A Muller theory $\Phi$ is a conjunction (or set) of formulas

$$\phi \iff e \mid e \supset \Diamond_D e \mid e \supset \Box e$$

where $e$ is a boolean expression over wires $\mathcal{Z}$ and $D \in \mathbb{R}$.

Semantics

$h \in \mathbb{R}^+ \rightarrow \mathbb{B}^{\mathcal{Z}}$ non-zeno, right-continuous waveform

$I = [s, t)$ time interval

$h; I \models e$ “$h$ ; $I$ remains in region $e$“

$h; I \models e_1 \supset \Diamond_D e_2$ “$h$ ; $I$ must enter $e_2$ within $D$ time, inside $e_1$“

$h; I \models e_1 \supset \Box e_2$ “$h$ ; $I$ cannot ever enter $\overline{e_2}$ from inside $e_1$“
Boolean expressions hold pointwise throughout the interval

$h; [5, 8) \models \text{roses} \cdot (\text{love} + \text{kiss})$

“roses and always love or kiss“

$h; [0, \infty) \models \overline{\text{roses}} \supset (\overline{\text{love}} \lor \overline{\text{love}})$

“while without roses, no change in love“

Example
◊ Modality for Propagation Delay („Set-up“)

\[ h; [s, t] \models \Diamond_D \phi \text{ iff } s + D < t \Rightarrow h; [s + D, t] \models \phi. \]

\[ h; [0, \infty) \models (\text{roses} \supset \Diamond_4 \text{love}) \land (\text{love} \supset \Diamond_2 \neg \text{roses}) \]

\[ h; [0, \infty) \models \text{kiss} \supset \Diamond_3 \text{false} \]
Modality for Inertiality ("Hold")

\[ h; [s, t) \models \Box \phi \iff s < t \Rightarrow \exists \delta > 0. h; [s, t + \delta) \models \phi \]

\[ h; [0, \infty) \models (\text{love} \supset \Box \text{love}) \land (\text{roses} \supset \Box \text{roses}) \]

\[ h; [0, \infty) \models (\overline{\text{love} + \text{kiss}}) \supset \Box \text{roses} \]
(Up-bounded) Muller Theories

Definition

• A Muller theory $\Phi$ is a conjunction of formulas

$$\phi ::= e \mid e \supset \lozenge_D e \mid e \supset \square e$$

where $e$ is a boolean expression over wires $\mathcal{Z}$ and $D \in \mathbb{R}$.

Theorem

• Muller theories $\Phi$ specify the (timed) General Multiple Winner behaviour $\text{GMW}(\Phi)$ upbounded inertial delay (UIN) Boolean networks.
\[ \Phi = \{ x \} \]
\[ \Phi = \{ x, \overline{y} \supset \square \overline{y}, \overline{y} \cdot z \supset \square z \} \]
Specifying GMW Automata in Muller Logic

\[ \Phi = \{ x, \overline{y} \supset \square \overline{y}, \overline{y} \cdot z \supset \square \overline{z}, y \cdot z \supset \Diamond_D \text{false}, \overline{z} \supset \Diamond_E \text{false} \} \]

inertiality

contraction

bounded regions
Specifying GMW Automata in Muller Logic

\[ \Phi = \{ x, \overline{y} \supset \square \overline{y}, \overline{y} \cdot z \supset \square z, y \cdot z \supset \Diamond_D \text{false}, \overline{z} \supset \Diamond_E \text{false} \} \]

**Fairness:** „The system trajectory cannot infinitely remain inside a transient region“
Specifying GMW Automata in Muller Logic

\[ \Phi \models z \supset \Diamond_D \neg y \quad \text{forced by (contraction and) inertiality} \]

\[ \Phi \models \neg y \supset \Diamond_E z \quad \text{forced by contraction alone} \]

\[ \Phi = \{ x, \]

\[ \neg y \supset \square \neg y \]

\[ \neg y \cdot z \supset \square \neg z \]

\[ y \cdot z \supset \Diamond_D \text{false} \]

\[ \neg z \supset \Diamond_E \text{false} \} \]

**Fairness:** „The system trajectory cannot infinitely remain inside a transient region“

M. Mendler, Bamberg University

FDL’23 PhD School, Torino, Sep 13, 2023
Specifying GMW Automata in Muller Logic

\[ \Phi \not\models z \supset \Diamond_D \overline{y} \quad \text{forced by inertiality} \]
\[ \Phi \models \overline{y} \supset \Diamond_E \overline{z} \quad \text{forced by contraction} \]

\[ \Phi = \{ x, \]
\[ y \cdot z \supset \Diamond_D \text{false} \]
\[ \overline{z} \supset \Diamond_E \text{false} \} \]

Purely non-inertial theory may lose stabilisation!
**Upbounded Inertial Delay (Romeo & Guilietta)**

\[ e_1 := D e_2 \]

stands for

\[ (\overline{e_2} \supset \Diamond_D \overline{e_1}) \land (e_2 \supset \Diamond_D e_1) \]
\[ (e_1 \cdot e_2 \supset \Box e_1) \land (\overline{e_1} \cdot \overline{e_2} \supset \Box \overline{e_1}) \]

\[ \Phi_{UIN} \models \Diamond (\text{roses} \cdot \text{kiss} \cdot \overline{\text{money}}) \]

\[ t_2 := D \overline{\text{money}} \]
\[ t_3 := D \overline{\text{money}} \cdot \text{roses} \]
\[ t_4 := D \text{kiss} \cdot t_5 \]
\[ t_5 := D \text{roses} \cdot \overline{t_4} \]
\[ \text{money} := D t_5 \]
\[ \text{roses} := D t_2 \]
\[ \Phi_{UIN} \]
Upbounded Non-Inertial Delay (Romeo & Gulietta)

$t2 : \approx_D \overline{\text{money}}$
$t3 : \approx_D \overline{\text{money}} \cdot \text{roses}$
$t4 : \approx_D \text{kiss} \cdot t5$
$t5 : \approx_D \text{roses} \cdot \overline{t4}$

money : \approx_D t5
roses : \approx_D t2
kiss : \approx_D t3

$\Phi_{\text{UNI}}$

$e_1 \vdash \approx_D e_2$

stands for

$(\overline{e_2} \supset \Diamond_D \overline{e_1}) \land (e_2 \supset \Diamond_D e_1)$

$\Phi_{\text{UNI}} \not\models \Diamond(\text{roses} \cdot \text{kiss} \cdot \overline{\text{money}})$
5

CONSTRUCTIVE MULLER THEORIES & TIMED TERNARY SIMULATION
Definition

• A Muller theory $\Phi$ is constructive if
  \[ \Phi \models \lozenge_D (e_1 \lor e_2) \text{ implies } \Phi \models \lozenge_D e_1 \text{ or } \Phi \models \lozenge_D e_2 . \]

• $\Phi$ is stabilising if $\forall z \in Z$ there is $D$ with $\Phi \models \lozenge_D (z \lor \neg z)$.

• A Muller theory $\Phi$ is non-inertial if it does not contain the $\Box$ operator.

Theorem [derived from Mendler, Shiple, Berry 2012]

• Every non-inertial Muller theory is constructive.

• Stabilisation can be decided by timed ternary simulation...
(Timed) Ternary Algebra

Recursion theory [Kleene'52]

Asynchronous Circuits & Fault-modelling (hazards, races, oscillation)
[Yoeli/Rinon'64, Eichelberger'65, Roth'66]
[Bryant'87] CMOS transistor-level simulation

Analysis of Muller Automata
[Yoeli/Brzozowski'77, Brzozowski/Seger'95] A/B-algorithms

Cyclic Combinational Circuits
[Burch/et.al.'93, Malik'93, Shiple'96]
[Huang/Parng/Shyu'91] Timed D-Calculus
[Fairtlough/Mendler’96,Mendler/Shiple/Berry‘2012]
modal logic/real-time interpretation
[Namjoshi/Kurshan‘99, Backes/Fett/Riedel‘2008]
improved (untimed) Algorithm

Synchronous programming
[Berry‘99, Schneider/Brandt/Schüle’2004, ...]
Timed Ternary Algebra

\[
\begin{array}{c|c|c|c}
\text{DEL}(d) & [\alpha, s] & [\alpha, s + d] & \text{NOT} \\
\hline
\downarrow & \downarrow \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
\text{OR} & [0, t] & [1, t] & \downarrow \\
\hline
[0, s] & [0, \max(s, t)] & [1, t] & \downarrow \\
[1, s] & [1, s] & [1, \min(s, t)] & [1, s] \\
\downarrow & \downarrow & [1, t] & \downarrow \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
\text{AND} & [0, t] & [1, t] & \downarrow \\
\hline
[0, s] & [0, \min(s, t)] & [0, s] & [0, s] \\
[1, s] & [0, t] & [1, \max(s, t)] & \downarrow \\
\downarrow & [0, t] & \downarrow & \downarrow \\
\end{array}
\]

\[\bot = [0, \infty) = [1, \infty) = [\bot, t)\]
Example I

Fixed Point
Example I

Fixed Point

\[ [0, 0] \]

\[ x \]

\[ s_1 [\bot, \infty], [\bot, \infty] \]

\[ s_2 [\bot, \infty], [\bot, \infty] \]

\[ s_3 [\bot, \infty], [\bot, \infty] \]
The following statements are equivalent:

- A network \( N \) is semantically stabilising in non-inertial Muller-Logic
- The \textbf{ternary simulation} of \( N \) generates Boolean solutions for the state variables
- \( N \) reaches in bounded time a \textbf{unique steady state} under \textbf{non-inertial delay} assumptions
CONCLUSION
Summary

- **Intuitionistic Muller Logic** (NEW!)
  - expressively adequate specification language for Boolean Networks
  - for inertial and non-inertial delay models
- **Timed Ternary Simulation** as an algorithmic decision procedure for non-inertial delay networks
- Definition of „Constructive Circuits“ (G. Berry) as networks that are stabilising
  - in **constructive Muller Logic** (axiomatic)
  - in **timed ternary simulation** (denotational)
  - under **non-inertial delay scheduling** (operational)
Open Research Problems

- Complete axiomatisation of Muller Logic
- Computational complexity of decision procedures
- Proof that complete (input and output) inertial delay networks are equivalent to non-inertial delay networks and thus constructive
- Exact separation between delay-insensitive and speed-independent networks in Muller Logic.

Nota Bene: These results are relevant (mutatis mutandis) to distributed systems at higher levels of abstraction, too (RTL, distributed shared memory, middleware, ...)

M. Mendler, Bamberg University