Analyzing Isochronic Forks with Potential Causality

Rajit Manohar[1] **Yoram Moses**[2]

[1]Cornell Tech, New York, NY 10011, USA

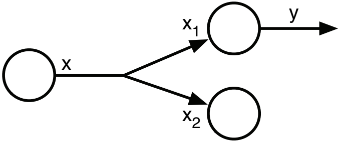
[2]Technion-Israel Institute of Technology, Haifa 32000, Israel

May 6, 2015

CORNELL TECH
HOME OF THE JACOBS INSTITUTE

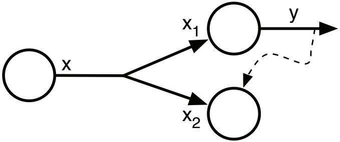
Isochronic forks

- Difference between purely delay-insensitive circuits and *quasi* delay-insensitive (QDI) circuits
- Some informal descriptions:
  - "we have to assume that the difference beween the delays in the branches of the fork is negligible compared to the delays in the gates."
  - "we assume that, when transition $x_1\uparrow$ has been acknowledged by transition $y\uparrow$, transition $x_2\uparrow$ is also completed."



Isochronic forks

Most recent approach notes the impact of an *adversarial path*



Intuition:

- If x to x_2 is an isochronic branch, then an error due to a slow transition on x_2 must manifest itself because some other path from x eventually causes a mis-firing of the gate that has x_2 as input.

A complex proof sketch in Keller et al. (ASYNC 2009).

Distributed systems

- "Asynchronous" processes
- Message-passing for communication
- Many classic results

Connecting this theory to circuits:

- Processes $\mapsto$ gates
- Messages $\mapsto$ signals

Foundational techniques:

- "Happened-before" causality relation (Lamport 1978)

Contributions

- Connecting asynchronous design with the distributed systems literature
  - Formalization of asynchronous computations
  - The notion of **potential causality** adapted
  - The **past** theorem
- Using this formalism, rigorous proofs of:
  - **Firing loop** theorem
  - **Aversarial firing chain** theorem
    A rigorous proof of the nature of the isochronic fork timing assumption

CORNELL TECH
HOME OF THE JACOBS INSTITUTE

AVLSI

The model

V: a set of *variables*

Production rules:

- $B \mapsto z\uparrow$ or $B \mapsto z\downarrow$
  where $z \in V$, B is a formula over the variables in V
- A gate is a pair $B_u \mapsto z\uparrow$, $B_d \mapsto z\downarrow$

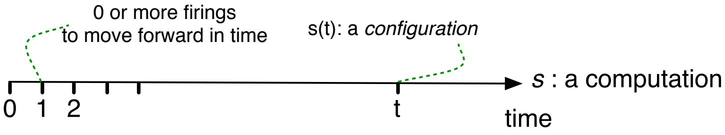
Circuit: a collection of $|V|$ gates, one per $z \in V$

A *configuration* of a circuit is an assignment $c \colon V \to \{0, 1\}$

A PR is *enabled* in a configuration if its guard is true.

Computations

A *computation* is an infinite sequence $s\colon \mathbb{N} \to \mathcal{C}$



- $s(t+1)$ is obtained from $s(t)$ by firing zero or more PRs enabled at $s(t)$

$s_x(t)$: the value of variable x at time t

- "x changes at time t in s" $\stackrel{\text{def}}{=} s_x(t+1) \neq s_x(t)$

CORNELL TECH
HOME OF THE JACOBS INSTITUTE

AVLSI

Nodes

We use the notation

$$\langle x, m \rangle$$

to represent a **node**.

- Node: represents the state of a variable at different times (Note that the value of x at time m in a computation s is $s_x(m)$)
- We will reason about the relation between nodes in a computation

CORNELL TECH
HOME OF THE JACOBS INSTITUTE

AVLSI

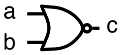
Potential causality

We write:

$$\langle y, m \rangle \hookrightarrow_s \langle z, m+1 \rangle$$

iff

- a PR with output z performs an effective firing at $s(m)$;
- y is in the support of the guard of the PR that fired



If in a computation s, $c\downarrow$ fires at time 100, then

$$\langle a, 100 \rangle \hookrightarrow_s \langle c, 101 \rangle \quad \wedge \quad \langle b, 100 \rangle \hookrightarrow_s \langle c, 101 \rangle$$

Not the same as true causality

Potential causality

For a computation s, we define $\preceq_s$ as the unique minimal relation that satisfies:

Locality:	$\langle y, t\rangle \preceq_s \langle y, t'\rangle$	if $t \leq t'$;
Successor:	$\langle y, t\rangle \preceq_s \langle z, t+1\rangle$	if $\langle y, t\rangle \hookrightarrow_s \langle z, t+1\rangle$;
Transitivity:	$\langle y, t\rangle \preceq_s \langle z, t'\rangle$	if, for some $\langle x, m\rangle$, both $\langle y, t\rangle \preceq_s \langle x, m\rangle$ and $\langle x, m\rangle \preceq_s \langle z, t'\rangle$.

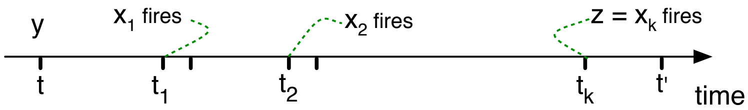
CORNELL TECH
HOME OF THE JACOBS INSTITUTE

AVLSI

Firing chains

Definition

There is a **chain of firings from $\langle y, t\rangle$ to $\langle z, t'\rangle$ in the computation** s if there is a sequence of variables $x_1, \ldots, x_k = z$ and a sequence of monotonically increasing times $t_1, \ldots, t_k$ with $t \le t_1$ and $t_k < t'$, such that $\langle y, t_1\rangle \hookrightarrow_s \langle x_1, t_1 + 1\rangle$ and such that $\langle x_{i-1}, t_i\rangle \hookrightarrow_s \langle x_i, t_i + 1\rangle$ holds for all $2 \le i \le k$.



Firing chains

Lemma

Let $y \neq z$. Then $\langle y, t \rangle \preceq_s \langle z, t' \rangle$ **iff** *both $t < t'$ and there is a chain of firings from $\langle y, t \rangle$ to $\langle z, t' \rangle$ in s.*

Why?

- The only way to move to a different variable in $\preceq_s$ is through the successor clause, i.e., a firing
- Related nodes are ordered in time

CORNELL TECH
HOME OF THE JACOBS INSTITUTE

AVLSI

The past

Definition

Given a computation s and a set T of variable-time nodes, we define:

$$\mathtt{past}_s(T) = \bigcup_{\langle y', m' \rangle \in T} \{\langle x, m \rangle : \langle x, m \rangle \preceq_s \langle y', m' \rangle\} .$$

Intuition:

- Given a node, its state can only be impacted by its past

The past theorem

- Given a circuit A, a computation s, times $m < m'$
- Given T, a set of nodes $\langle y, m' \rangle$ at time m'

Theorem (past theorem)

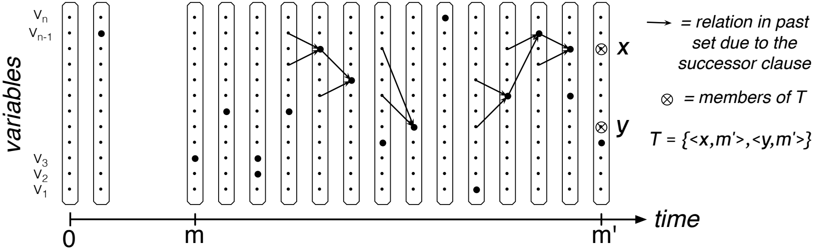
There is a computation s' of A such that:

- $s'(t) = s(t)$ *for all times* $t \leq m$;
- *For all variables* x *and times* t *in the range* $m < t \leq m'$:
  - (a) $s'_x(t) = s_x(t)$ *if* $\langle x, t \rangle \in \mathtt{past}_s(T)$, *and*
  - (b) $s'_x(t) = s_x(m)$ *if* $\langle x, m+1 \rangle \notin \mathtt{past}_s(T)$.

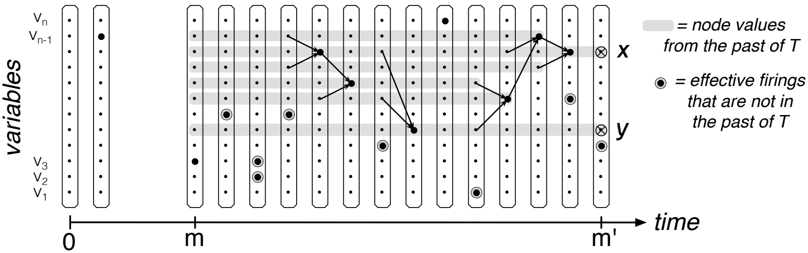
CORNELL TECH
HOME OF THE JACOBS INSTITUTE

AVLSI

Original computation:



Nodes in the past of T:



CORNELL TECH
HOME OF THE JACOBS INSTITUTE
AVLSI

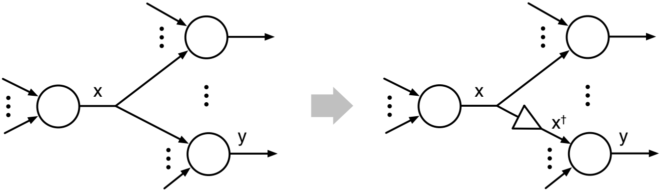
The past theorem

The construction of s':

- Upto time m, replicate firings from s;
- Beyond time m, only replicate firings when the appropriate node is in $\mathtt{past}_s(T)$

Main proof obligation: enabled firings in s are enabled in s' (see paper)

Non-isochronic branches



Original circuit: A (left)

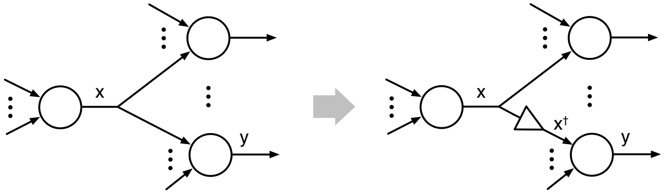
Modified circuit: $A^{\dagger}$ (right)

What can we say about $A^{\dagger}$?

If the introduced buffer is on a non-isochronic branch, then:

Theorem (firing loop)

For every computation $w^\dagger$ *of* $A^\dagger$ *where* x *changes at times* t *and* $t' > t$, *there is a chain of firings from* $\langle x, t+1 \rangle$ *to* $\langle x, t'+1 \rangle$ *in* $w^\dagger$ *that includes a change in* $x^\dagger$.



CORNELL TECH
HOME OF THE JACOBS INSTITUTE
AVLSI

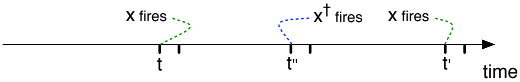
The firing loop theorem

Proof:

- Use the past theorem with

$$T = \{\langle x, t' + 1\rangle\}; \quad m := t; \quad m' := t' + 1$$

- The past theorem gives us $u^\dagger$: a new computation in $A^\dagger$ with only the firings from the past of T
- We know that $x^\dagger$ is stable, and x changed twice.
- $\Rightarrow x^\dagger$ must have fired in $u^\dagger$ at some time t'', $t < t'' < t'$.



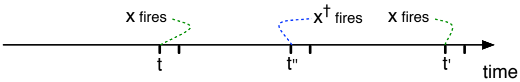
1. $\langle x^\dagger, t'' + 1\rangle \preceq_{u^\dagger} \langle x, t' + 1\rangle$
2. $\langle x, t''\rangle \hookrightarrow_{u^\dagger} \langle x^\dagger, t'' + 1\rangle$

Proof:

- Use the past theorem with

$$T = \{\langle x, t' + 1\rangle\}; \quad m := t; \quad m' := t' + 1$$

- The past theorem gives us $u^\dagger$: a new computation in $A^\dagger$ with only the firings from the past of T
- We know that $x^\dagger$ is stable, and x changed twice.
- $\Rightarrow x^\dagger$ must have fired in $u^\dagger$ at some time t'', $t < t'' < t'$.



1. $\langle x^\dagger, t'' + 1\rangle \preceq_{u^\dagger} \langle x, t' + 1\rangle$
2. $\langle x, t''\rangle \hookrightarrow_{u^\dagger} \langle x^\dagger, t'' + 1\rangle$

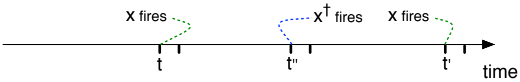
The firing loop theorem

Proof:

- Use the past theorem with

$$T = \{\langle x, t' + 1\rangle\}; \quad m := t; \quad m' := t' + 1$$

- The past theorem gives us $u^\dagger$: a new computation in $A^\dagger$ with only the firings from the past of T
- We know that $x^\dagger$ is stable, and x changed twice.
- $\Rightarrow x^\dagger$ must have fired in $u^\dagger$ at some time t'', $t < t'' < t'$.



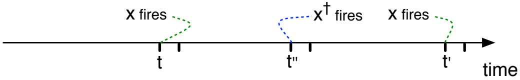
1. $\langle x^\dagger, t'' + 1\rangle \preceq_{u^\dagger} \langle x, t' + 1\rangle$
2. $\langle x, t''\rangle \hookrightarrow_{u^\dagger} \langle x^\dagger, t'' + 1\rangle$

Proof:

- Use the past theorem with

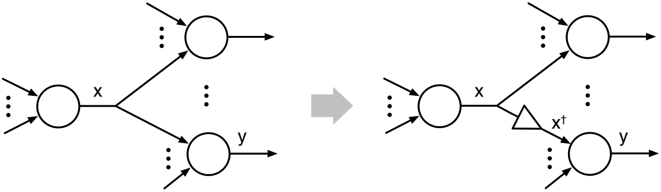
$$T = \{\langle x, t' + 1\rangle\}; \quad m := t; \quad m' := t' + 1$$

- The past theorem gives us $u^\dagger$: a new computation in $A^\dagger$ with only the firings from the past of T
- We know that $x^\dagger$ is stable, and x changed twice.
- $\Rightarrow x^\dagger$ must have fired in $u^\dagger$ at some time t'', $t < t'' < t'$.



1. $\langle x^\dagger, t'' + 1\rangle \preceq_{u^\dagger} \langle x, t' + 1\rangle$
2. $\langle x, t''\rangle \hookrightarrow_{u^\dagger} \langle x^\dagger, t'' + 1\rangle$

Isochronic branches



Original circuit: A (left)

Modified circuit: $A^\dagger$ (right)

When are they "the same"?

Stuttering free computations

Given a computation s, we define $\underline{s}$ as the **stuttering-free** variant of s.

Lemma

If s is a computation of a circuit A, then $\underline{s}$ is also a computation of A.

Consistent computations

Given two circuits:

- A with variables V and a computation s
- A' with variables added to V, modified production rules, and a computation w
- $w|_V$: the **restriction** of w to the variables in V

Definition

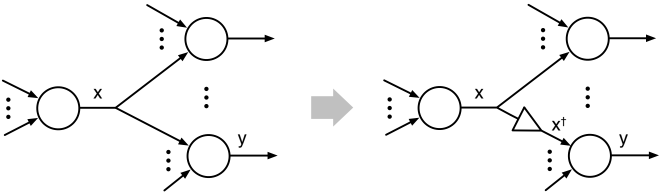
s is **consistent** with w, denoted $s \approx w$, if $\underline{s} = \underline{w|_V}$

Idea: hiding the new variables in w gives back s

CORNELL TECH
HOME OF THE JACOBS INSTITUTE

AVLSI

Isochronic branches



Lemma

For every computation s of A, there exists a computation $w^\dagger$ of $A^\dagger$ where $s \approx w^\dagger$.

Compatible for m rounds

An even stronger requirement:

Definition

Given computation s of A and $w^\dagger$ of $A^\dagger$, we write $s \sim_m w^\dagger$ (s and $w^\dagger$ are **compatible for** m **rounds**) if $s(t) = w^\dagger|_V(t)$ for $t = 0, \ldots, m$.

Lemma

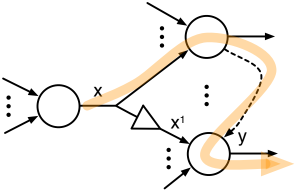
For all $w^\dagger$ of $A^\dagger$, there exists s of A such that $s \approx w^\dagger$ iff there is an s' of A such that $s' \sim_m w^\dagger$ for all m.

- Suppose $w^\dagger$ is a computation of $A^\dagger$, and
- $s \approx w^\dagger$ holds for *no computation* s of A

Then:

Theorem (adversarial firing chain)

There is a firing chain in $w^\dagger$ from $\langle x, t\rangle$ to $\langle y, t'\rangle$ for some times $t < t'$ that does not include a firing of $x^\dagger$; in particular, $x^\dagger$ is unchanged between t and t' in $w^\dagger$.



The adversarial firing chain theorem

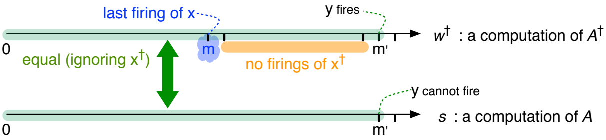
Let $m' > 0$ be the largest time where there is some s of A where

$$s \sim_{m'} w^{\dagger}$$

- $\Rightarrow$ an effective firing at m' in $w^{\dagger}$ that cannot occur in s
- the only choice is y

  $$\Rightarrow s_x(m') \neq w^{\dagger}_{x^{\dagger}}(m'), \text{ hence } w^{\dagger}_x(m') \neq w^{\dagger}_{x^{\dagger}}(m')$$

- $w^{\dagger}_x(0) = w^{\dagger}_{x^{\dagger}}(0)$ implies a firing of x in $w^{\dagger}$ before m'



CORNELL TECH
HOME OF THE JACOBS INSTITUTE

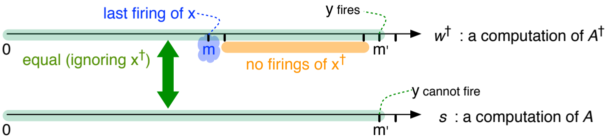
AVLSI

The adversarial firing chain theorem

Let $m' > 0$ be the largest time where there is some s of A where

$$s \sim_{m'} w^{\dagger}$$

- $\Rightarrow$ an effective firing at m' in $w^{\dagger}$ that cannot occur in s
- the only choice is y
  $\Rightarrow s_x(m') \neq w^{\dagger}_{x^{\dagger}}(m')$, hence $w^{\dagger}_x(m') \neq w^{\dagger}_{x^{\dagger}}(m')$
- $w^{\dagger}_x(0) = w^{\dagger}_{x^{\dagger}}(0)$ implies a firing of x in $w^{\dagger}$ before m'



CORNELL TECH
HOME OF THE JACOBS INSTITUTE

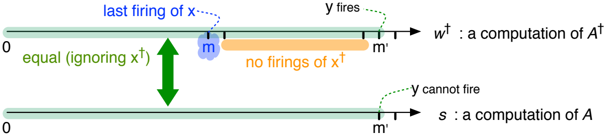
AVLSI

The adversarial firing chain theorem

Let $m' > 0$ be the largest time where there is some s of A where

$$s \sim_{m'} w^{\dagger}$$

- $\Rightarrow$ an effective firing at m' in $w^{\dagger}$ that cannot occur in s
- the only choice is y
  $$\Rightarrow s_x(m') \neq w^{\dagger}_{x^{\dagger}}(m'), \text{ hence } w^{\dagger}_x(m') \neq w^{\dagger}_{x^{\dagger}}(m')$$
- $w^{\dagger}_x(0) = w^{\dagger}_{x^{\dagger}}(0)$ implies a firing of x in $w^{\dagger}$ before m'

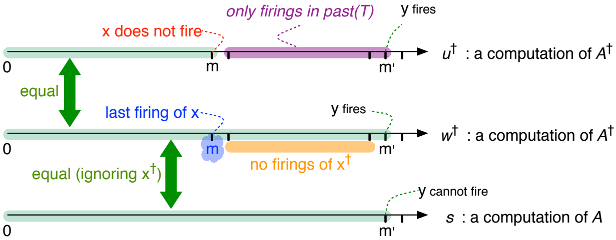


Assume that $\langle x, m+1\rangle \not\preceq_{w^\dagger} \langle y, m'+1\rangle$.

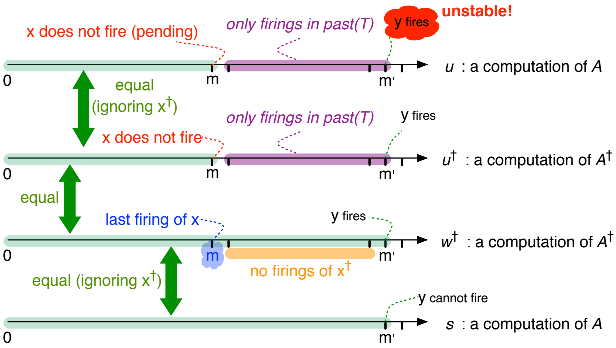
Let $B_y^\dagger$ be the guard of y in question in $A^\dagger$. We apply the **past theorem** with times m and m' to:

$$T = \{\langle h, m'\rangle \colon h \text{ is an input to } B_y^\dagger \text{ in } A^\dagger\}$$

and obtain $u^\dagger$.



The adversarial firing chain theorem



$$\therefore \qquad \langle x, \underbrace{m+1}_{t} \rangle \preceq_{w^\dagger} \langle y, \underbrace{m'+1}_{t'} \rangle$$

The adversarial firing chain theorem

Corollary

Let $x^\dagger$ be an input to y. If the fastest adversarial firing chain from a change in x to y is slower than the delay of the buffer $x^\dagger$, then for every computation $w^\dagger$ of $A^\dagger$, there exists a computation s of A such that $s \approx w^\dagger$.

CORNELL TECH
HOME OF THE JACOBS INSTITUTE

AVLSI

Summary

- The notion of **potential causality** adapted from distributed systems
- The **past** theorem
- Using this formalism, rigorous proofs of:
  - **Firing loop** theorem
  - **Adversarial firing chain** theorem
- If delays on isochronic branches are smaller than their corresponding adversarial firing chains, then the set of possible computations is the same as the set in a zero-delay fork model.

CORNELL TECH
HOME OF THE JACOBS INSTITUTE

Support: ISF grant 1520/11; Ruch grant, Jacobs Institute.

AVLSI