A compositional theory of digital circuits

George Kaye
University of Birmingham
07 February 2023 – LFCS seminar, Edinburgh

How we got here

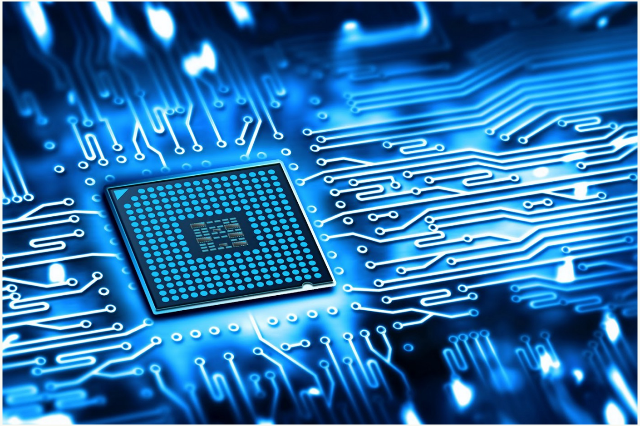
‘Hi Ohad, want to do a seminar at Birmingham?’

Six months later...

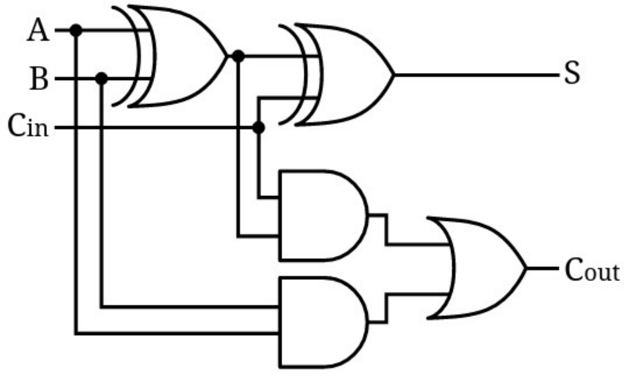
‘How about you do a seminar at Edinburgh first?’

What are we going to be talking about?

Digital circuits!

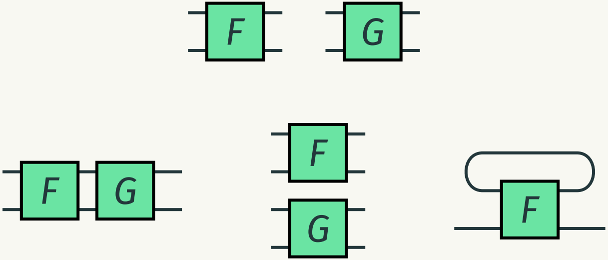


Digital circuits!



What are we going to be talking about?

We want a compositional theory of digital circuits.



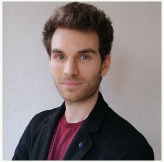
These operations may look familiar to you!

Previous attempts foiled by non-delay-guarded feedback.

What came before

Lafont (2003) *'Towards an algebraic theory of Boolean circuits'*

Ghica, Jung, Lopez (2017) *'Diagrammatic semantics for digital circuits'*

Enter stage left

David Sprunger
Indiana State University

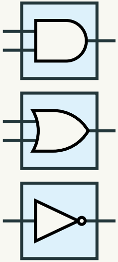
Dan Ghica
University of Birmingham

(and me) 

Syntax

Combinational circuit components

gates

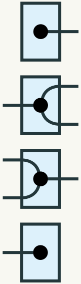


AND gate

OR gate

NOT gate

(co)monoid structure



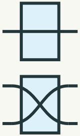
disconnected

fork

join

stub

categorical structure



identity

symmetry

Light circuits

only contain gates and structure.

Sequential circuit components

We need state.

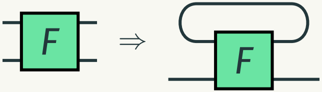
Values

f	false
t	true
b	short circuit

Along with •, models *Belnap's four value logic*

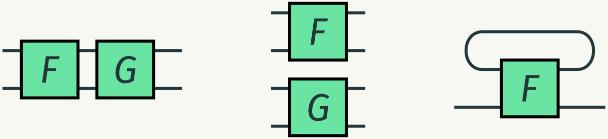
Delay

Feedback



Dark circuits F may contain delay or feedback.

Circuits are morphisms in a freely generated symmetric traced monoidal category (STMC).

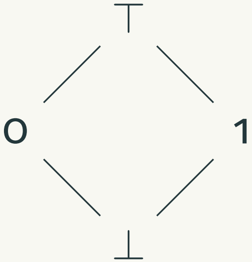


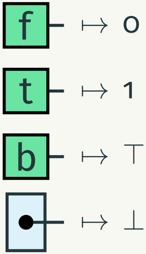
SCirc

Semantics

We need some meaning

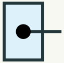
Values are interpreted in a lattice:



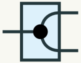


Let's make everything a function

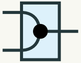
monotone functions $\overline{g} : \mathbf{V}^m \to \mathbf{V}$



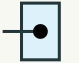
initialise $() \mapsto (\bot)$



copy $x \mapsto (x, x)$



join in the lattice $(x, y) \mapsto x \sqcup y$



discard $x \mapsto \bullet$

Feedback is interpreted as the least fixed point.

Functions are not enough

How do we model delay?

Streams!

Streams

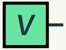
A stream $\mathbf{V}^\omega$ is an infinite sequence of values.

$$v_0 :: v_1 :: v_2 :: v_3 :: v_4 :: v_5 :: v_6 :: v_7 :: \cdots$$

A stream function $\mathbf{V}^\omega \to \mathbf{V}^\omega$ consumes and produces streams.

$$f(v_0 :: v_1 :: v_2 :: v_3 :: v_4 :: \cdots) = w_0 :: w_1 :: w_2 :: w_3 :: w_4 :: \cdots$$

Interpreting the sequential components



$$() := v :: \bot :: \bot :: \bot :: \cdots$$

$$(v_0 :: v_1 :: v_2 :: \cdots) := \bot :: v_0 :: v_1 :: v_2 :: \cdots$$

Maybe there are too many streams

Does every circuit correspond to a stream function

$$(\mathbf{V}^m)^\omega \to (\mathbf{V}^n)^\omega?$$

No.

(but this is to be expected!)

Circuits are causal and monotone

Circuits are causal.

They can only depend what they've seen so far.

Circuits are monotone.

They are constructed from monotone functions.

Is that all? Not quite... (but we'll get there)

Some operations on stream function

Given a causal stream function $f\colon (\mathbf{V}^m)^\omega \to (\mathbf{V}^n)^\omega$ and an element $a \in \mathbf{V}^m$...

initial output $\quad f[a] \in \mathbf{V}^n$

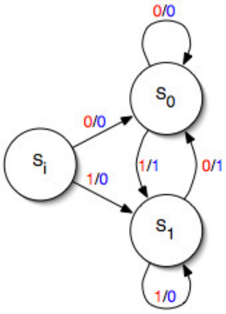
'the first thing f produces given a'

stream derivative $\quad f_a \in (\mathbf{V}^m)^\omega \to (\mathbf{V}^n)^\omega$

'how f behaves after seeing a first'

Hold on, these look familiar...

An old friend



Mealy machine moment!

Stream functions are the *states* in a Mealy machine.

Circuits have finitely many behaviours

Circuits have a finite number of components.

So there are finite number of states in the Mealy machine.

So the outputs of streams given some input must be periodic.

(There are finitely many stream derivatives).

These are the streams we're looking for

Theorem

A stream function is the interpretation of a sequential circuit if and only if it is ***causal, monotone*** *and has* ***finitely many stream derivatives****.*

Equational reasoning

Now what?

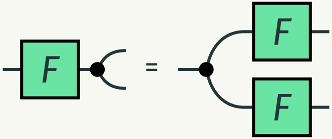
We have a sound and complete semantics for circuits as stream functions.

But reasoning with streams can be a pain...

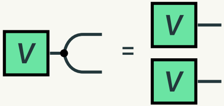
Why not reason equationally?

The two types of equation

Global



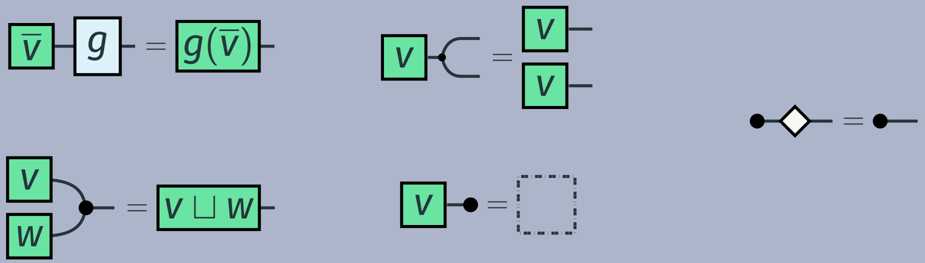
Local



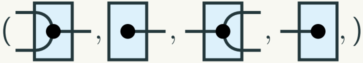
We want to stick to local equations as much as possible.

What do the values do?

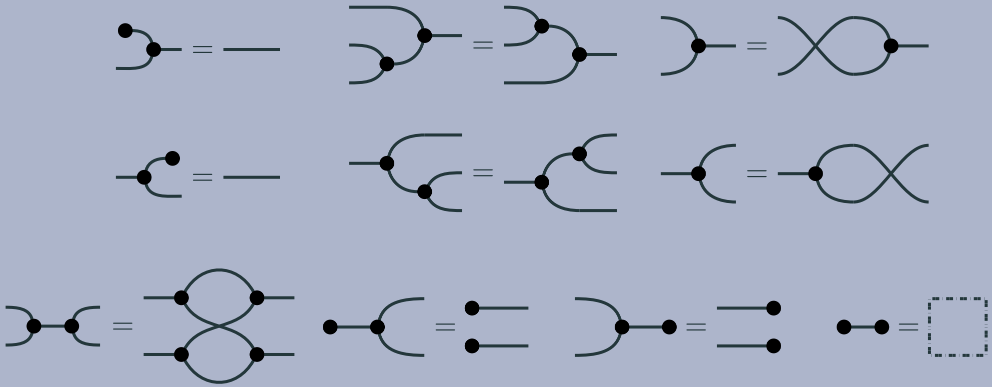
Axiom(s)



Let's get structural

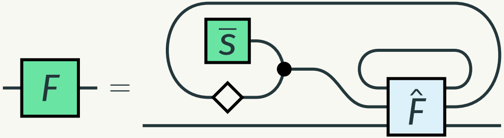
( ,) is a bialgebra.

Axiom(s)



Splitting things up

Isolate the sequential and combinational components...



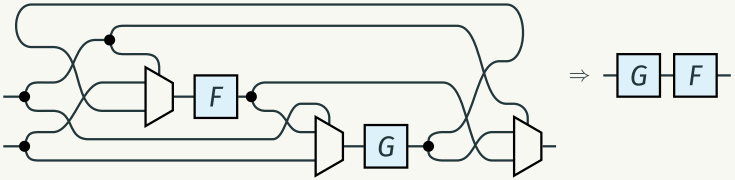
This is *almost* another Mealy machine moment...

...but there is the non-delay-guarded trace!

Do we even need it?

In industry, normally circuits must be delay-guarded.

But this rules out some clever circuits!



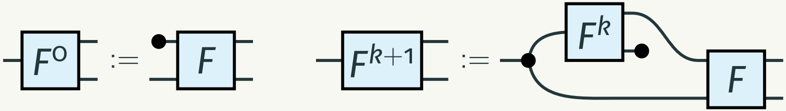
(And also it would be cheating)

Getting rid of non-delay-guarded feedback

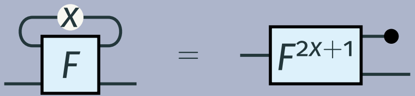
V is a finite lattice...

The functions are monotone...

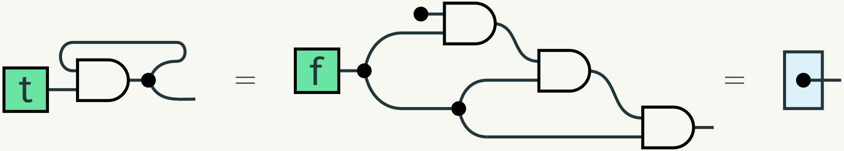
We can compute the least fixed point in finite iterations!



Axiom(s)

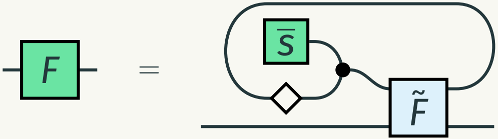


Getting rid of non-delay-guarded feedback



Here's Mealy

For any circuit

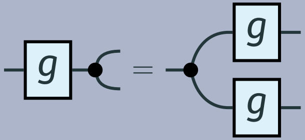


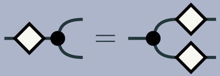
Let's get (even more) structural

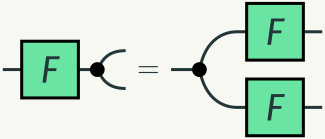
What more structure can we add?

Forking is natural...

Axiom(s)

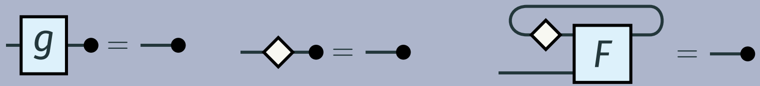




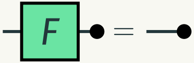
For any circuit... 

Let’s do the same for stubbing...

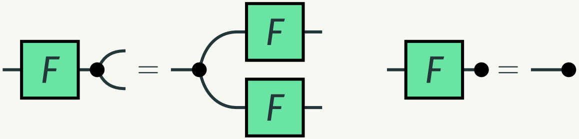
Axiom(s)



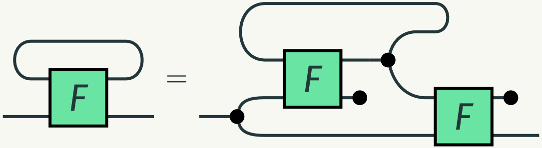
For any circuit...



Let's get structural

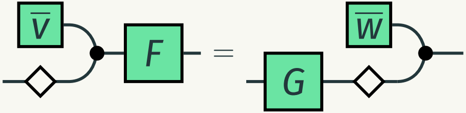


Cartesian!



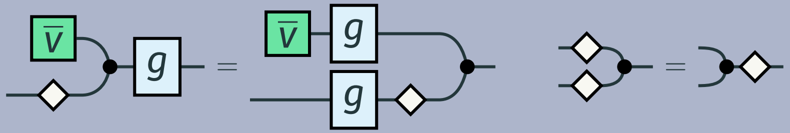
Running the simulation

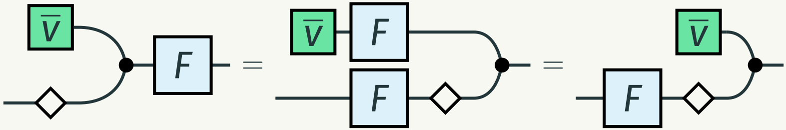
Goal:



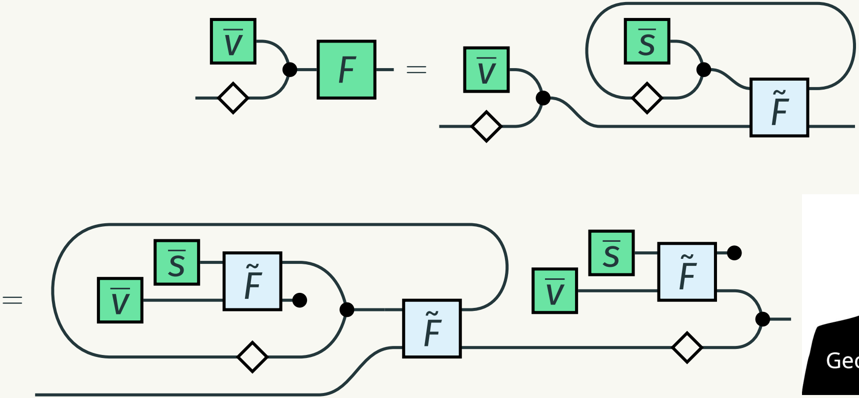
Running the simulation

Axiom(s)





Running the simulation



We made it!

We have defined a sound and complete compositional theory of digital circuits.

Also defined an equational theory for sequential circuits!

Ghica, Dan R., K., and David Sprunger (2022). *A compositional theory of digital circuits*. URL: https://arxiv.org/abs/2201.10456.

(currently not the most up to date version...)