Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Embedding Boolean Circuits into Fungal Sandpile Automata with Arbitrary Update Sequences

Eric Goles
Adolfo Ibáñez University
Augusto Modanese
Helmholtz Center for Information Security
Martín Ríos
Adolfo Ibáñez University
Domingo Ruiz
druiz@dim.uchile.cl
University of Chile
Thomas Worsch
Karlsruhe Institute of Technology

Research Article

Keywords:

Posted Date: November 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11589827/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Embedding Boolean Circuits into Fungal Sandpile Automata with Arbitrary Update Sequences

Eric Goles[1], Augusto Modanese[2], Martín Ríos-Wilson[1], Domingo Ruiz-Tala[3*], Thomas Worsch[4]

[1]Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Chile.
[2]CISPA Helmholtz Center for Information Security, Germany.
[3]Departamento de Ingeniería Matemática, Facultad de Ingeniería y Ciencias, Universidad de Chile, Chile.
[4]Retired; formerly Karlsruhe Institute of Technology (KIT), Germany.

*Corresponding author(s). E-mail(s): druiz@dim.uchile.cl;
Contributing authors: eric.chacc@uai.cl; augusto.modanese@cispa.de;
martin.rios@uai.cl; thomas.worsch@posteo.de;

Abstract

The sandpile automata of Bak, Tang, and Wiesenfeld [Phys. Rev. Lett., 1987] are a simple model for particle diffusion in space. A fundamental problem related to the complexity of the model is predicting its evolution in the parallel setting. Despite decades of effort, a classification of this problem for two-dimensional sandpile automata remains outstanding. Fungal sandpile automata were recently proposed by Goles et al. [Phys. Lett. A, 2020] as a spin-off of the model in which diffusion occurs either in horizontal ($\boldsymbol{H}$) or vertical ($\boldsymbol{V}$) direction according to a so-called update scheme. Goles et al. proved that the prediction problem for this model with the update scheme $\boldsymbol{H^4V^4}$ is **P**-complete. This result was subsequently improved by Modanese and Worsch [Algorithmica, 2024], who showed the problem is **P**-complete also for the simpler update scheme $\boldsymbol{HV}$. In this work, we complete the characterization and prove the prediction problem is **P**-complete for any update scheme containing both $\boldsymbol{H}$ and $\boldsymbol{V}$ at least once.

Keywords: cellular automata, circuit embedding, parallel and space-bounded complexity, sandpile automata

---

Some of the results in this article were presented at MCU 2026.

1 Introduction

The *sandpile model* of Bak, Tang, and Wiesenfeld [1] is a classical discrete model for particle diffusion. Concretely, one might picture *grains of sand* that are distributed across a lattice of *cells* that partitions Euclidean space. Each cell contains a certain number of grains of sand and, when this number reaches a fixed threshold value, the cell *fires*, equally distributing grains to its neighbors.

The sandpile model has been extensively studied in the context of cellular automata theory, where it is presented as a class of *sandpile automata*. In the conventional model of two dimensions, the neighbors of a cell are fixed to be the ones immediately located along its four cardinal directions. The states are simply numbers from 0 to 7 indicating the number of grains in a cell, and the threshold for firing is 4.

A related formalism, distinct from the fungal sandpile model considered here, is that of sand automata; its connection with cellular automata is studied in [2].

The complexity of predicting the behavior of sandpile automata in two dimensions is a long-standing open problem in the field [3].

Recently, Goles et al. [4] proposed a variant of sandpile automata that was in turn inspired by another class of automata called *fungal automata* [5] (see also [6]). These are a nature-inspired model of computing that mimics the behavior of septa in fungi. Septa are cross-walls between compartments inside the cytoplasm of fungi organisms. The compartments communicate flow through structures called Woronin bodies, which change their behavior depending on external conditions to be closed (preventing the flow from one compartment to another) or open.

Concretely, the *fungal sandpile automata* defined by Goles et al. [4] are (two-dimensional) sandpile automata where the neighborhood of a cell is not fixed but dynamically changes over time according to a so-called *update scheme*. This is a word $Z \in \{H, V\}^+$ that defines the neighborhood of cells at every time step following a cyclic rotation. The H and V symbols in Z indicate whether at the corresponding time point the cells distribute grains to their horizontal (i.e., east and west) or vertical (i.e., north and south) neighbors.

The fungal sandpile prediction problem studied here is formally defined in Section 2.1.

Characterizing the complexity of predicting cellular automata and related models such as automata networks is a recurring object of study [7–16], and the corresponding prediction problem for sandpile automata has been the subject of intensive efforts as well [3]. Letting d denote the number of dimensions of the sandpile automaton, for $d = 1$ it is known that the prediction problem can be solved in a parallel setting as it is contained in a subclass of NC [17]. Conversely, for $d \geq 3$, the problem has been shown to be P-complete [18]. As already mentioned, the complexity of the problem for $d = 2$ has been open for very long. One of the main motivations for studying fungal sandpile automata (or other variants of sandpile automata) is to explore and develop techniques that could shed more light into the case of ordinary two-dimensional sandpiles.

In their original paper, Goles et al. [4] showed that the fungal sandpile prediction problem associated with $Z = H^4V^4$ is P-complete. This result was subsequently improved by Modanese and Worsch [19], who introduced several novel techniques to prove that P-completeness is still there with the simplest non-trivial update scheme

possible, namely $Z = HV$ (see also Section 1.2 further below). The underlying approach in both papers is a reduction from the circuit value problem (CVP), which is well-known to be P-complete, but relying on different strategies to embed Boolean circuits in fungal sandpiles.

1.1 Our Contribution

In this paper, we settle an open question of Modanese and Worsch [19] and show that the fungal sandpile prediction problem is P-complete *whenever the update scheme* $Z \in \{H, V\}^+$ *is non-trivial*, that is, whenever it contains H and V at least once.

Theorem 1 *If $Z \in \{H, V\}^+$ is any word that contains at least one H and one V, then the fungal sandpile prediction problem associated with Z is* P*-complete.*

To show this result, we adapt the modular approach of Modanese and Worsch [19]. The main challenge is that describing and proving the low-level gadgets to be correct must be done without explicit reference to any given Z and, in particular, we no longer can refer to gadgets in a static manner. We overcome this by introducing a *new intermediate layer* that allows the descriptions to remain static despite Z being variable. See Section 1.2 for further discussion.

A preliminary version of this work is available in [20].

1.2 Technical Discussion

In the rest of the paper we describe how to embed any Boolean circuit with description B and an assignment of values x to the inputs into a configuration F of a fungal sandpile automaton in such a way that the following holds:

- A designated cell, initially in state 0, reaches state 1 at some time $t \leq T$, measured in individual update steps, if and only if the output of the circuit is 1. We use the cell adjacent to the positive output pin chosen in Section 7, which receives its first grain from that output.
- Given B and x, the configuration F can be produced in logarithmic space. Thus we have a log-space reduction from the CVP to the fungal sandpile automaton prediction problem.

As already mentioned above, the construction uses a reduction from the circuit value problem (CVP), which we restate here for the sake of concreteness.

> *Input:* The description of a single-output Boolean circuit B with bounded fan-in as well as an input $x \in \{0, 1\}^+$ to the circuit.
> *Question:* Is the final output of B on x equal to 1 (True)?

In order to obtain a sensible problem, some care must be taken regarding the format of the description in which B is given. The problem is well-known and these details are not relevant in the present paper, and so we omit a more detailed treatment.

It suffices to say that the description must be structured in such a way that it can be processed gate-by-gate by a logspace machine. It is a well-known result that the CVP is P-complete, which means the problem is in P and also that any problem in P can be reduced to it by a logspace reduction.

Inspired by [19], the construction is structured into multiple *layers*. We keep the same layer numbering as in [19] and add an intermediate layer between Layers 1 and 2. In addition, there are several modifications throughout that are required for the construction to generalize to an arbitrary update word.

Next we give an overview of these layers and discuss the adaptations needed.

Layer 1: block partitioning.

The first layer consists of partitioning the grid into blocks. This allows better organization of *signals*, which are cells with a state 4 or 5. They are so called because they induce a self-propagating behavior when placed on a path of cells with state 3 (see Fig. 1). In our generic construction, the block sizes are determined by the composition of the parameter word Z. More specifically, letting h and v be the counts of the symbols H and V in Z, respectively, our blocks are of size $h \times v$. To be precise, we use 2×2 blocks for update schemes such as $Z = HHVV$, whereas for $Z = HV$ we would actually have a degenerate partition into 1×1 blocks. To avoid an overly technical discussion, we skip the cases $h = 1$ and $v = 1$ in this high-level overview.

Our construction adapts the block organization of [19], using a more fine-grained division than in the original. The general approach is to have every block simulate a single cell in the original HV construction. This allows us to apply the higher layers of abstraction without having to alter them too much.

That being said, it is not possible to obtain such a simple simulation directly; in fact, it is necessary for a block to use also the cells that are adjacent to its boundaries. Hence it is not trivial how to "glue" blocks together and also not at all clear how to handle corners, that is, cells where four distinct blocks intersect. This is the first technical difficulty to be overcome and requires extra care throughout the construction.

See Section 3.

Layer 1.5: bridges.

This new layer introduces the key notion that we employ to generalize the result in [19] to an arbitrary word Z; these are the so-called *bridges*. Their core functionality is to convey signals from one block to the other. In the construction of Modanese and Worsch, signals are *moved* through wires, which consist of 2×2 blocks of cells with value 3 connected horizontally or vertically. This mechanism arises naturally from the subdivision of the grid into 2×2 blocks used in their construction.

As previously mentioned, signal propagation in our setting requires a new subdivision of the grid into blocks whose size depends on Z. Signals are then transmitted through these larger blocks using structures we call *bridges*. Each Z has a fitting set of bridges and there is a simple schematic for performing this translation in logarithmic space.

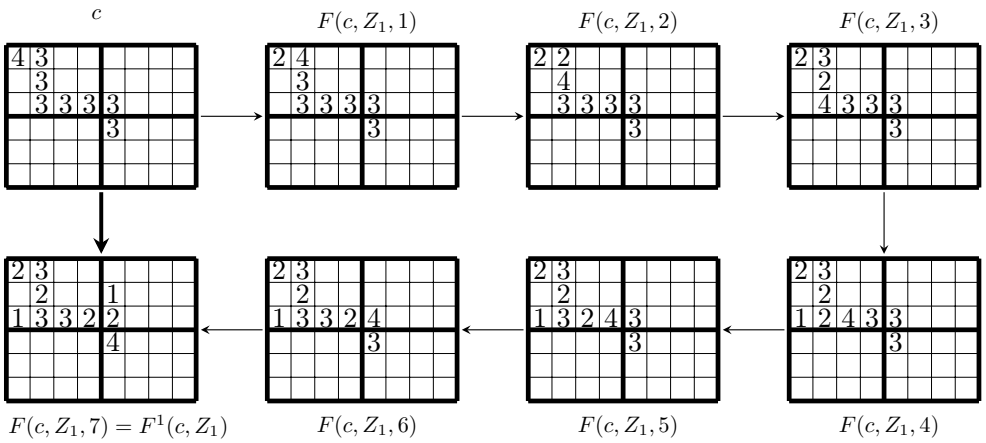


Figure 1: Updates according to the word $Z_1 = HVVHHHV$. The initial configuration c is shown in the top-left. The arrows, starting to the right, indicate the next configuration after each corresponding update. The configuration immediately below the initial one corresponds to the state after completing the entire cycle of the word.

An example can be seen in Fig. 1. Observe that a bridge consists of a sequence of cells with value 3 connecting the same relative positions of two blocks placed diagonally with respect to each other.

See Section 4.

Layer 2: polarized circuits.

This layer demands great technical effort. The original construction uses a set of gadgets to build a limited class of circuits that operates on *polarized signals*, that is, signals that may be *positive* or *negative*, which is determined by their relative placement inside a block.

In our construction, bridges connect B_1^+ to B_2^+ or B_1^- to B_2^-, so polarity is preserved along semiwires built from bridges of the same polarity.

We provide a brief overview of the set of gadgets required:

- Basic wire operations to merge and duplicate signals of the same polarity, implemented by *semi-merges* and *semi-duplicators*.
- *Semi-crossings*, which are a rudimentary form of crossing *between signals of opposite polarity* that is destroyed upon use.
- Operators called *switches* that allow a certain logical operation similar to that of a transistor between signals of opposite polarities.
- *Diodes* to ensure signals traverse a wire in a single direction.
- *Retarders* to allow delaying a signal by a variable amount of time.

The most distinctive components here are the semi-crossings and switches, which fundamentally rely on the distinct behavior of positive and negative signals discussed in Section 9.

We inherit the blueprints from [19] and reimplement all components using the previously defined bridges.

This allows the construction to be compatible with subsequent layers while supporting any underlying update sequence Z. Here the bridges serve as an abstraction layer, allowing us to redefine all components without any explicit dependencies on Z.

See Section 5.

Layer 3: Boolean circuits with delay.

Polarized circuitry is used to implement *Boolean circuits with delay*: "bits" are processed by "gates" connected by "cables", which are pairs of positive and negative wires. (We use the term "wire" strictly for the constructs in Layer 2.) Since cables have a length (which is a concept not available in standard Boolean circuits), they incur delays and hence it is necessary to take into account the arrival times for individual signals.

See Section 6.

Layer 4: logspace constructor.

The final layer is the *logspace constructor* that is in charge of taking a CVP instance and actually producing the description of the embedding associated with it. Since this layer sits on top of the previous one, not much adaptation is needed compared to the original construction. Nevertheless, one needs to take care of a few details; for example, the constructor now depends on Z and there are also additional factors in the calculations incurred by the use of bridges.

See Section 7.

1.3 Structure of the Article

The rest of the article is structured as follows. Section 2 presents basic definitions, notation, and the prediction problem. Sections 3 to 5 present Layer 1, Layer 1.5, and Layer 2, respectively. Section 6 recalls the abstraction of Layer 3 from [19, Section 5], and Section 7 addresses the adaptations to the Layer 4 constructor. Section 8 is devoted to the proof of the main theorem. Finally, Section 9 discusses implications and future work.

2 Preliminaries

The set of integers is denoted by $\mathbb{Z}$, that of positive integers by $\mathbb{N}_+$, and that of non-negative integers by $\mathbb{N}_0$. For a map $f\colon A \to B$ and a subset $A' \subseteq A$, we write $f|_{A'}$ for the restriction of f to A'. We write $[P]$ for the characteristic function of a predicate P; that is, $[P] = 1$ if P is true, or 0 otherwise. For a finite, non-empty set Σ and $k \in \mathbb{N}_0$, we write $w \in \Sigma^k$ for a word $w = w(0)\cdots w(k-1)$ over Σ of length k.

A two-dimensional *cellular automaton* is a tuple (Q, N, f) where Q is a finite set of states, the *neighborhood* N is a finite subset of $\mathbb{Z}^2$, and $f\colon Q^N \to Q$ is the *local rule*. A *(global) configuration* is a map $c\colon \mathbb{Z}^2 \to Q$. It defines a *local configuration* $\lambda_x\colon N \to Q$ at every $x \in \mathbb{Z}^2$ by associating the elements of N with the states of the cells relative to x (i.e., if $y - x \in N$ for some $y \in \mathbb{Z}^2$, then $\lambda_x(y - x) = c(y)$). When applied

everywhere simultaneously, a local rule f induces a global function $F\colon Q^{\mathbb{Z}^2} \to Q^{\mathbb{Z}^2}$ given by $F(c)(x) = f(\lambda_x)$.

A *fungal sandpile automaton* is a two-dimensional cellular automaton equipped with the *von Neumann neighborhood*, i.e., $N = \{(a, b) \in \mathbb{Z}^2 \mid |a| + |b| \leq 1\}$ and set of states $Q = \{q \in \mathbb{N}_0 \mid q \leq 5\}$. Based on N, given a cell $x \in \mathbb{Z}^2$ we let $N_h(x) = \{x+(1,0), x-(1,0)\}$ be the *horizontal neighborhood* and $N_v(x) = \{x+(0,1), x-(0,1)\}$ the *vertical neighborhood* of x. Cells are updated according to a two entry update function $F : Q^{\mathbb{Z}^2} \times \{H, V\} \to Q^{\mathbb{Z}^2}$, where for $x \in \mathbb{Z}^2$:

$$F(c, H)(x) = c(x) - 2[c(x) \geq 4] + \sum_{y \in N_h(x)} [c(y) \geq 4],$$
$$F(c, V)(x) = c(x) - 2[c(x) \geq 4] + \sum_{y \in N_v(x)} [c(y) \geq 4].$$

Note that we are using the threshold 4 here despite each cell firing only two sand grains at a time. This is as in the original definition of Goles et al. [4], which changes only the *receiving end of grain firing*, not the threshold 4 of the original 2D sandpile model. It is plausible that our construction would also work with a threshold of 2 (with certain modifications), but we remain faithful to the original definition as well as to [19] and do not pursue that here.

In Fig. 1, we show an example of the dynamics of the fungal sandpile automaton for the word $Z = HVVHHHV$. All the empty cells (cells with zero grains) are represented by blank spaces. In addition, all the cells that are not part of the figure have zero grains.

Let $Z \in \{H, V\}^k$. We abuse notation and define the global transition function induced by Z as the function $F(c, Z, s + 1) = F(F(c, Z, s), Z(s))$ where $F(c, Z, 0) = c$. The sequence of configurations $F(c, Z, 0), \ldots, F(c, Z, k)$ is a *cycle* of Z. In addition, we denote $F(c, Z, k)$ by simply $F(c, Z)$ and $F(c, Z, ik)$ by $F^i(c, Z)$. Figure 1 shows an example of a complete cycle.

We write $Z \in \{H, V\}^{h,v,k}$ whenever Z is such that the symbol H appears h times and V appears v times in Z. In the rest of the paper we sometimes use the identification $H = (1, 0)$ and $V = (0, 1)$ whenever the context is clear and write $|w|_a := |\{i : w(i) = a\}|$ for any $w \in \{H, V\}^+, a \in \{H, V\}$.

For any configuration $c \in Q^{\mathbb{Z}^2}$, a cell $x \in \mathbb{Z}^2$ with $c(x) \geq 4$ is called a *signal*.

2.1 The Problem

Given a complex discrete dynamical system, a fundamental question is the complexity of *predicting* its behavior. That is, given an initial configuration, an objective entity in the system (e.g., the state of a cell in a cellular automaton), and a bound on the observation time, will the entity assume some state of interest or not? In the case of sandpiles, we are particularly interested in determining if a cell will ever contain a grain of sand. For the specific variant of fungal sandpiles, our *fungal sandpile prediction* problem is formally defined as follows.

Parameters: A word $Z \in \{H, V\}^+$.
Input: The initial configuration c for a finite grid R of cells, the index of a cell x (encoded in binary), and a positive time bound T (encoded in unary).
Question: Suppose the initial configuration of the fungal sandpile automaton is such that we place c in R surrounded by cells all in the state 0. Will cell x assume a non-zero state at some time point $t \leq T$?

Assumptions on Z.

For the construction, we assume without loss of generality that Z starts with H and ends with V, contains each of H and V at least twice (that is, $h, v \geq 2$), and is not a proper power: there are no $r \geq 2$ and $W \in \{H, V\}^+$ such that $Z = W^r$. The conditions $h, v \geq 2$ avoid degeneracy in the block construction. For instance, when $v = 1$, the positive and negative source cells of a block coincide.

If Z starts with V and ends with H, we use the same construction rotated by 90 degrees. If Z contains both symbols and starts and ends with the same symbol, we move its initial run to the end of the word. For instance, let $Z = V^a HWV^b$, where $a, b \in \mathbb{N}_+$ and $W \in \{H, V\}^*$ may be empty. After the first a updates, the dynamics follow the word $Z' = HWV^{a+b}$. We add a waiting steps at the beginning of the construction by extending each source bridge with a wire of a cells carrying the initial signal in the correct direction, as illustrated in Fig. 2. The case in which Z starts and ends with H is handled analogously by rotation.

If $Z = W^r$ for $r \geq 2$ and $W \in \{H, V\}^+$, we perform the reduction for W, since repeating Z and repeating W give the same periodic sequence of updates.

3 Layer 1: Block Partitioning

We start by describing the main structure on which our construction is based. This structure is called a block. For $Z \in \{H, V\}^{h,v,k}$, a Z-block B is a grid of size $h \times v$. We drop the mention of Z when it is clear from the context.

Given a Z-block B, B^+ and B^- are the cells in the upper- and lower-left corners of B, respectively. More precisely, if $x = (a, b)$ denotes the cell in the upper-left corner of B, then $B^+ = x$ and $B^- = x + (0, (v - 1))$. We call B^+ (resp., B^-) the *positive* (resp., *negative*) *source cell* of B. These cells are special since they allow us to define the polarity of a signal in Layer 2 (see Section 1.2) relative to its placement inside the block.

A Z-block subdivision of the grid is a subdivision of the grid in Z-blocks as in Fig. 3 for the word $Z = HVVHHHV \in \{H, V\}^{4,3,7}$. Also, observe that in Fig. 1 there are 4 blocks forming a 2×2 shape for the word $Z_1 = HVVHHHV$.

In a Z-block subdivision, we say two blocks B_1, B_2 are *vertically adjacent* if $B_1^+ = B_2^+ \pm vV$; similarly, we say they are *horizontally adjacent* if $B_1^+ = B_2^+ \pm hH$. In either case, either orientation counts. Two blocks B_1, B_2 are *diagonally connected* if there exists a block B horizontally adjacent to B_1 and vertically adjacent to B_2. In Fig. 3, green blocks are diagonally connected.

Let $Z \in \{H, V\}^{h,v,k}$ and B be a Z-block. The *closure* of B is the set of cells $\overline{B}$ containing B and all its vertical, horizontal and diagonal neighbors, formally

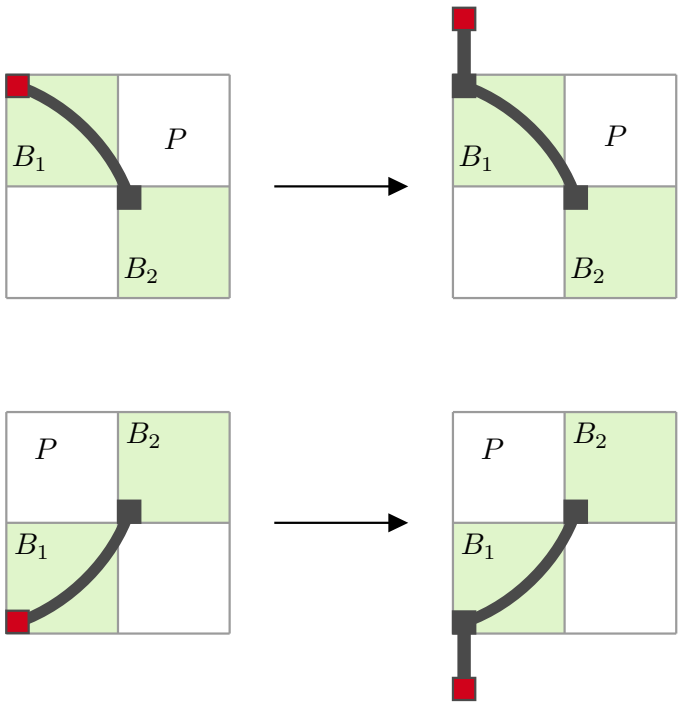


Figure 2: Source bridges modification. At each source cell, we concatenate cells with value 3 either above or below it. This ensures that a signal reaches the source cell of the block, allowing it to start functioning as intended for the new, *non-shifted* word.

$\overline{B} = B \cup \{x + aH + bV \in \mathbb{Z}^2 \mid x \in B \text{ and } a, b \in \{-1, 0, 1\}\}$. In Fig. 3, the closure of the center block consists of the cells marked in blue.

Definition 1 Given $Z \in \{H, V\}^{h,v,k}$ and a cell $P(0) \in \mathbb{Z}^2$, a *Z-path* is a sequence $P = (P(0), P(1), \ldots, P(l))$ where $l \leq k$, $P(i) \in \mathbb{Z}^2$ for all i, and there exist $\alpha, \beta \in \{-1, 1\}$ such that, for every $s \in \{0, \ldots, l-1\}$,

$$P(s+1) = \alpha Z(s)[Z(s) = H] + \beta Z(s)[Z(s) = V] + P(s).$$

Observe that a Z-path is characterized only by its starting cell $P(0)$, the signs α and β, and its length l. We say that a Z-path P connects two cells $x, y \in \mathbb{Z}^2$ if $P(0) = x$ and $P(l) = y$.

Lemma 1 *Let $Z \in \{H, V\}^{h,v,k}$ and consider diagonally connected Z-blocks B_1, B_2. For each pair $(x, y) \in \{(B_1^+, B_2^+), (B_1^-, B_2^-)\}$, there exists a Z-path $P = (P(0), \ldots, P(k))$ connecting x and y.*

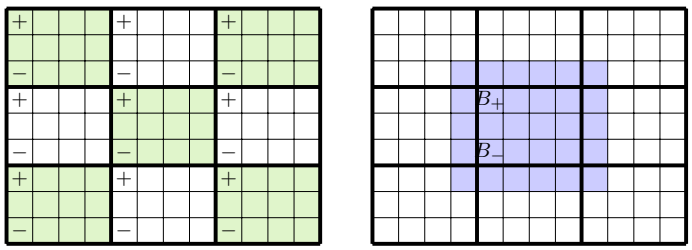


Figure 3: On the left, blocks for $Z = HVVHHHV \in \{H,V\}^{4,3,7}$ divided with thick black lines. Positive and negative sources are marked with $+, -$. The central block B and the blocks diagonally connected to it are colored green. On the right, $\overline{B}$ is shown in blue.

Proof Let x, y be given. We need to show that there exists a Z-path $P = (P(0), \dots, P(k))$ starting from $P(0) = x$ such that $P(k) = y$. In other words, since Z-paths can be defined by $P(0)$ and some α, β we need to show that it is possible to choose $\alpha, \beta \in \{-1, 1\}$ such that $P(k) = y$.

First observe that, for every Z-path $P = (P(0), \dots, P(k))$, we have

$$P(k) = P(0) + \sum_{i=0}^{k-1} Z(i)(\alpha[Z(i) = H] + \beta[Z(i) = V]).$$

On the other hand, we have that cells x and y satisfy $y = x + \alpha' hH + \beta' vV$ for some $\alpha', \beta' \in \{-1, 1\}$. Choosing $(\alpha, \beta) = (\alpha', \beta')$, it follows that

$$\begin{aligned} P(k) &= P(0) + \sum_{i=0}^{k-1} Z(i)(\alpha[Z(i) = H] + \beta[Z(i) = V]) \\ &= x + \alpha hH + \beta vV = x + \alpha' hH + \beta' vV \\ &= y. \end{aligned}$$

Thus, P connects x and y. The lemma holds. □

4 Layer 1.5: Bridges

Now we introduce the main element of our construction: *bridges*. From now on, for every word Z we work with a Z-subdivision of the grid.

For a fixed word Z, a bridge is a structure defined by a particular Z-path and a configuration c. Intuitively, the path marks a trail of cells with 3 grains. These paths connect positive (or negative) source cells of diagonal Z-blocks.

Formally, given 2 blocks B_1, B_2 diagonally connected (notation example in Fig. 4) we define a *Z-bridge* for a word $Z \in \{H,V\}^{h,v,k}$ as the tuple $T = (P, c)$ where $P = (P(0), \dots, P(k))$ is a Z-path connecting B_1^+, B_2^+ or B_1^-, B_2^- and c is a configuration such that $c(x) = 3$ if $x \in P$, or $c(x) = 0$ otherwise. If we set $c(P(0)) = 4$ instead, then T is a *source Z-bridge*. Furthermore, if we have $c(P(l)) = 2$ for some $0 \leq l \leq k$ and $c(P(j)) = 0$ for every $j > l$, then T is a *bridge with a sink at l*. This is well-defined since, by Lemma 1, a path can always be defined that connects the diagonally connected blocks B_1 and B_2.

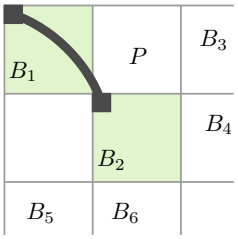


Figure 4: Bridge notation for a positive bridge $T = (P, c)$ connecting blocks B_1 and B_2, which are highlighted in green. Light gray lines indicate the division between Z-blocks while the bridge is marked in dark gray. The squares refer to the cells $P(0)$ and $P(k)$. The concavity of the bridge is intentional, highlighting that T starts with a horizontal step and ends with a vertical one.

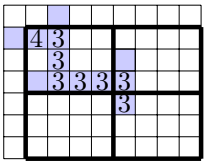


Figure 5: Affected neighbors of a bridge T, whose configuration is shown, for the word $Z_1 = HVVHHHV$. In blue, $N(T)$, the cells that receive at least one grain during the propagation of the signal (together with the source cell), as illustrated in Fig. 1.

A bridge T connecting positive cells, i.e. when $P(0)$ and $P(k)$ are both positive sources, is referred to as a *positive* bridge. In the same way, if P connects negative source cells, i.e. $P(0)$ and $P(k)$ are negative sources, then it is a *negative* bridge.

Remark 1 A positive bridge connects B_1^+ to B_2^+, and a negative bridge connects B_1^- to B_2^-. A positive (respectively, negative) semiwire, defined in Section 5.1, is formed by concatenating positive (respectively, negative) bridges and has positive (respectively, negative) input and output source cells. In this sense, polarity is preserved by construction at the designated source cells.

Let $Z \in \{H, V\}^{h,v,k}$ be a word and $T = (P, c)$ a Z-bridge.

We define $N(T)$, called the *affected neighbors* of T, as the set of cells which receive at least one grain during the propagation of the signal (together with the source cell) (see Fig. 5). Formally, letting $I = \{i \mid Z(i) = H\}$ and $J = \{j \mid Z(j) = V\}$, we define

$$N(T) = P \cup \left[\bigcup_{i \in I} N_h(P(i))\right] \cup \left[\bigcup_{j \in J} N_v(P(j))\right].$$

Now, we show that bridges can transmit signals. This is a key element of our construction. The idea is that a signal can go through the bridge following the structure of the Z-path contained in T.

Lemma 2 *Let $Z \in \{H,V\}^{h,v,k}$ and let us consider a source Z-bridge $T = (P, c)$ connecting B_1^+, B_2^+ from two diagonally connected blocks B_1, B_2, with $P = (P(0), \dots, P(k))$. Let $s\colon \mathbb{Z}^2 \to \{0,1,2,3\}$ satisfy $s(x) \le 2$ for $x \in N(T) \setminus P$, and define the global configuration*

$$\tilde{c}(x) = \begin{cases} c(x), & \text{if } x \in P, \\ s(x), & \text{if } x \notin P. \end{cases}$$

Thus $P(0)$ is the only initial signal, with value 4, and $\tilde{c}(P(j)) = 3$ for $1 \le j \le k$. Then $F(\tilde{c}, Z)(P(k)) = 4$.

Proof Let α, β be the constants defining the Z-path P as in Definition 1. By induction on i, we show that for all $0 \le i \le k$, $F(\tilde{c}, Z, i)(P(i)) = 4$ and $F(\tilde{c}, Z, i)(P(j)) = 3$ for each $j > i$. For $i = 0$ the result holds directly by definition.

Write $h_t = |Z(0) \cdots Z(t-1)|_H$ and $v_t = |Z(0) \cdots Z(t-1)|_V$, with $h_0 = v_0 = 0$, so that $P(t) = P(0) + \alpha h_t H + \beta v_t V$. Lowering the only initial signal $P(0)$ to 3 gives a stable configuration $\hat{c}$. If $P(0) = (p_x, p_y)$, set

$$\mathcal{K}_t = \{(x,y) \in \mathbb{Z}^2 : |x - p_x| \le h_t,\ |y - p_y| \le v_t\}.$$

The configuration agrees with $\hat{c}$ outside $\mathcal{K}_t$: this holds initially, and each H update expands the rectangle by one column on each side, each V update by one row on each side. A cell outside the expanded rectangle and both its neighbors in the update direction were outside the preceding rectangle, so its update agrees with the stable background. This bound and the transport induction use only the initial background bound 3 outside P.

Let $i \ge 1$ and let us assume $F(\tilde{c}, Z, i-1)(P(i-1)) = 4$ and $F(\tilde{c}, Z, i-1)(P(j)) = 3$ for each $j > i - 1$. From the bridge definition we have

$$P(i) = \alpha Z(i-1)[Z(i-1) = H] + \beta Z(i-1)[Z(i-1) = V] + P(i-1).$$

Suppose $Z(i-1) = H$; the case $Z(i-1) = V$ is similar. Then $P(i) = P(i-1) + \alpha H$. The other horizontal neighbor $P(i) + \alpha H$ lies outside $\mathcal{K}_{i-1}$, at horizontal distance $h_i + 1 > h_{i-1}$ from $P(0)$, and has value at most 3. Since we have:

- $F(\tilde{c}, Z, i-1)(P(i-1)) = 4$;
- $F(\tilde{c}, Z, i-1)(P(i)) = 3$;
- $P(i) \in N_h(P(i-1))$;
- $Z(i-1) = H$;

we conclude by definition that $F(\tilde{c}, Z, i)(P(i)) = 3 + 1 = 4$.

Observe that $F(\tilde{c}, Z, i)(P(j)) = 3$ for all $j > i$ follows from the same influence bound: $h_j \ge h_i$, $v_j \ge v_i$, and $h_j + v_j = j > i = h_i + v_i$, so at least one inequality is strict. Since every $P(j)$ with $j > i$ lies outside $\mathcal{K}_i$, its value remains unchanged. Taking $i = k$ proves the result. □

Corollary 1 *Let $Z \in \{H,V\}^{h,v,k}$ and consider two Z-bridges $T_i = (P_i, c_i)$, $i \in \{1,2\}$, connecting blocks B_1, B_2 and B_2, B_3, respectively, as in Fig. 7. Assume that $P_1(k) = P_2(0)$*

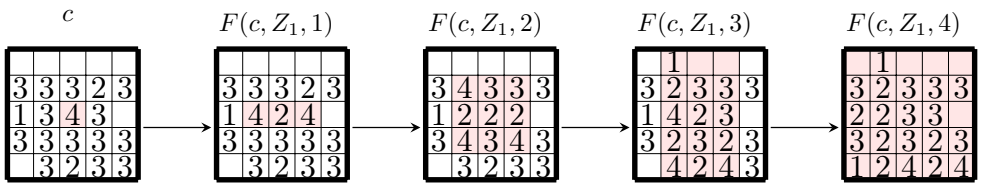


Figure 6: Sequence of updates for a configuration c according to the word $Z_1 = HVVHHHV$. The cells outside the light-red rectangle in the configuration shown remain unchanged with respect to the initial configuration. This observation is relevant to the proof of Lemma 2.



Figure 7: On the left, the structure for Corollary 1 is shown: Z-blocks B_1, B_2, and B_3 are diagonally connected, with two bridges T_1 and T_2 connecting B_1 to B_2, and B_2 to B_3, respectively. On the right, an example is given for the word $Z = HVVHHHV$. We consider two bridges T_i for $i = 1, 2$, where T_1 connects the top-left Z-block (B_1) with the central Z-block (B_2), and T_2 connects the central Z-block with the bottom-right block (B_3, not fully shown in the figure). The cells that belong to either bridge are highlighted in blue. All cells outside have values strictly less than 4.

and both paths have the same orientation, so that their concatenation is a ZZ-path. Let $s\colon \mathbb{Z}^2 \to \{0, 1, 2, 3\}$ and define the global configuration

$$\tilde{c}(x) = \begin{cases} 4, & \text{if } x = P_1(0), \\ 3, & \text{if } x \in (P_1 \cup P_2) \setminus \{P_1(0)\}, \\ s(x), & \text{if } x \notin P_1 \cup P_2. \end{cases}$$

Then $F^2(\tilde{c}, Z)(P_2(k)) = 4$.

Proof The concatenated path $Q = (P_1(0), \ldots, P_1(k), P_2(1), \ldots, P_2(k))$ is a path for the word ZZ (that is, Z concatenated with itself). Its initial values are 4 at $Q(0)$ and 3 at every other path cell, with all exterior values at most 3. The cone argument and transport induction in the proof of Lemma 2 use only this exterior bound and apply identically to Q for $2k$ updates. Therefore $F(\tilde{c}, ZZ)(Q(2k)) = F^2(\tilde{c}, Z)(P_2(k)) = 4$. □

Lemma 3 *Let $Z \in \{H, V\}^{h,v,k}$ and consider a source Z-bridge $T = (P, c)$ with a sink at $l \in \{1, \ldots, k\}$, connecting B_1^+, B_2^+ from two diagonally connected blocks B_1, B_2, with*

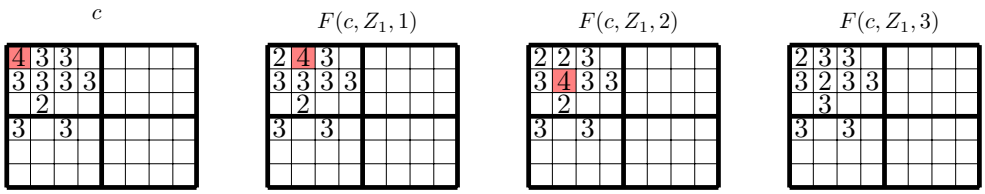


Figure 8: Structure and functionality for Lemma 3, corresponding to the word $Z_1 = HVVHHHV$, and a bridge $T = (P, c)$ with a sink at 3. Thick black lines indicate the division between blocks. Cells outside the bridge have values less than 4, and after 3 updates, $P(3)$ reaches value 3, having started at 2.

$P = (P(0), \ldots, P(k))$. *Let* $s\colon \mathbb{Z}^2 \to \{0, 1, 2, 3\}$ *satisfy* $s(x) \leq 2$ *for* $x \in N(T) \setminus P$, *and define the global configuration*

$$\tilde{c}(x) = \begin{cases} 4, & \text{if } x = P(0), \\ 3, & \text{if } x = P(j),\ 1 \leq j < l, \\ 2, & \text{if } x = P(l), \\ 0, & \text{if } x = P(j),\ l < j \leq k, \\ s(x), & \text{if } x \notin P. \end{cases}$$

Then $F(\tilde{c}, Z)(P(l)) = 3$.

Proof The proof is analogous to the previous lemma, except that now we have

$$F(\tilde{c}, Z, l)(P(l)) = F(\tilde{c}, Z, l-1)(P(l)) + 1 = 2 + 1 = 3.$$

We justify that, with our assumption on $N(T) \setminus P$, no cell has value 4 or greater at step l. The path is monotone, with Manhattan distance $|a - b|$ between $P(a)$ and $P(b)$. An exterior cell has at most two path neighbors, since opposite neighbors would force the path through that cell. If there are two, they are $P(a), P(a+2)$ at opposite corners of a unit square, with $P(a+1)$ at the other corner. The update $Z(a)$ is perpendicular to the direction toward the exterior cell. Thus prescribed, single firings of the path cells send at most one grain to each exterior cell, necessarily in $N(T) \setminus P$.

As in the previous induction, at time $0 \leq t < l$ the only signal is $P(t)$ with value 4, earlier cells have fired once, and later cells retain their initial values. Initially this holds by definition. In update $Z(t)$, $P(t)$ falls to 2 and sends a grain to $P(t+1)$. Its predecessor is the only earlier path cell that can receive the backward grain; this is its only receipt after falling to 2, so it stays at most 3 and cannot fire again. By the geometry above, each exterior recipient receives at most one grain and starts at most 2, hence stays at most 3. Cells outside $N(T)$ retain their initial values at most 3. There are therefore no exterior firings. The next cell has received no earlier grain and rises to 4 if $t + 1 < l$, or to 3 if $t + 1 = l$, proving the displayed equality and closing the induction. All earlier path cells and exterior cells are at most 3, and the tail remains 0. Thus no signal remains at step l; the configuration is stable through time k, and $F(\tilde{c}, Z)(P(l)) = 3$. □

That is, bridges with a sink interrupt the propagation of signals through the bridge, which is useful for constructing Layer 2 gadgets.

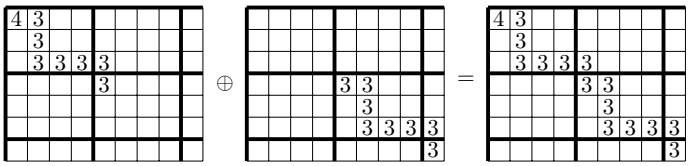


Figure 9: Operation $\oplus$ to concatenate two bridges.

Combining bridges.

Our circuit embedding relies on bridges as an intermediate modular structure. Since bridges depend on the word Z (which is arbitrary), we need to define a formal way to combine them.

As bridges are defined as a tuple consisting of a set and a configuration, we define an operation between configurations to be used in the union of bridges. Given the configurations c_1, c_2 we define $c_1 \oplus c_2$ as the configuration c such that:

$$c(x) = \begin{cases} 2, & \text{if } c_1(x) = 2 \vee c_2(x) = 2 \\ \max\{c_1, c_2\}(x), & \text{otherwise} \end{cases}$$

Given two bridges $T_1 = (P_1, c_1)$ and $T_2 = (P_2, c_2)$, we define $T_1 + T_2 := (P_1 \cup P_2, c_1 \oplus c_2)$, which will be called a bridge gadget, or simply a gadget when the context is clear. We extend the definition to gadgets obtained from other gadgets: formally, given two gadgets $G_1 = (Q_1, c_1), G_2 = (Q_2, c_2)$, we define the gadget $G = G_1 + G_2 := (Q_1 \cup Q_2, c_1 \oplus c_2)$.

5 Layer 2: Polarized Circuits

We inherit the blueprints from [19] but reimplement all components using the previously defined bridges.

As explained in Section 4, the equal-polarity endpoints of bridges and semiwires preserve polarity by construction.

The intended interface, under the hypotheses below, is:

- Semiwire: transport through successive blocks (Section 5.1; proof under review).
- Semi-duplicator: one input, two outputs (Corollary 2).
- Semi-merge: two non-simultaneous inputs, one usable output (Lemma 10; interface assumed).
- Semi-crossing: opposite polarities cross (Fig. 14; general functionality assumed).
- Switch: a source passes after an earlier gate (Lemmas 7 and 8).
- Diode: coordinated forward transmission (Fig. 21; by reference).
- Retarder: adjustable delay as in the original construction (Fig. 16).

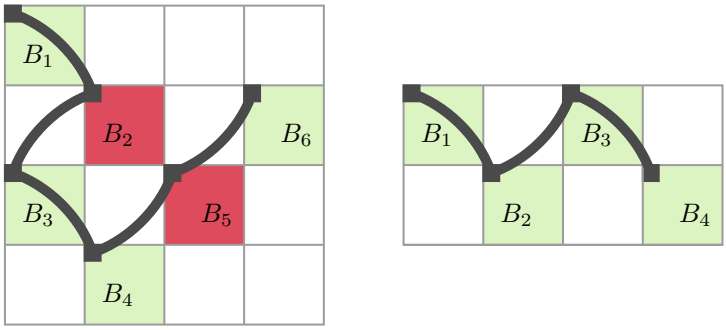


Figure 10: Invalid sequence for a semiwire (left), as blocks B_2 and B_5 (shown in red) are diagonally connected but are not consecutive in the enumeration defining the semiwire. Valid sequence for a semiwire (right)

5.1 Semiwires

Formally, let a sequence of distinct Z-blocks $B_0, \ldots, B_m$ be given such that, for all $i \in \{0, ..., m-1\}$, B_i is diagonally connected to B_{i+1} and is not diagonally connected to any block B_j with $j \neq i+1, i-1$. In addition, let the positive (resp., negative) bridges $T_i = (P_i, c_i)$ be given, where P_i connects B_i and B_{i+1}.

A *positive* (resp., *negative*) *semiwire* is defined as the bridge gadget $(S, c) = \sum_{i=0}^{m-1} T_i$ and it has B_0^+ (resp., B_0^-) as its *input* and B_m^+ (resp., B_m^-) as its *output*.

The next lemma states how signals travel through semiwires. Roughly speaking, condition 1 assures that after every cycle, a signal arrives at the next Z-block and condition 2 verifies that the remaining part of the semiwire remains unchanged so it can transmit the information.

Lemma 4 *Let (S, c) be a positive semiwire with T_0 a source bridge and all cells outside S initially zero. For every $i \in \{0, \ldots, m\}$:*

1. $F^i(c, Z)(B_i^+) = 4$.
2. $F^i(c, Z)(P_j(l)) = 3$ *for* $i \leq j \leq m-1$ *and* $1 \leq l \leq k$.

The second condition is empty when $i = m$.

Proof Proceed by induction on i. The base case $i = 0$ follows from the initial configuration. Suppose $0 \leq i < m$ and both conditions hold for $c^i := F^i(c, Z)$. The targets are $F(c^i, Z)(B_{i+1}^+) = 4$ and unchanged values 3 on $P_j(l)$ for $i+1 \leq j \leq m-1$, $1 \leq l \leq k$. For the first target, the intended application is the transport argument in the proof of Lemma 2, rather than its stronger affected-neighbor hypothesis. Its stable exterior bound after the preceding cycles still requires justification.

For the second target, consider the single-front passage along P_i in Fig. 11. At time l, $0 \leq l < k$, the cell $P_i(l)$ fires under update $Z(l)$. Its affected neighbors are in the purple blocks. The exclusion of nonconsecutive diagonally connected blocks keeps P_j, $j > i+1$, out of those blocks. During this passage, no additional exterior signals are assumed.

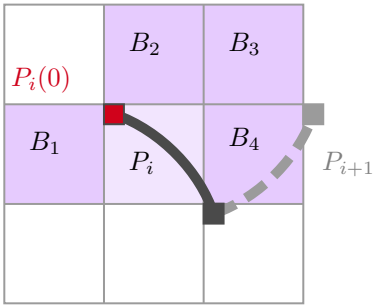


Figure 11: Affected blocks for a bridge P_i over a cycle. In purple, the blocks adjacent to P_i that can gain grains during the cycle starting with a signal at $P_i(0)$ are shown. In gray, the orientation for the path P_{i+1} that could cause it to gain grains during the cycle.

If $i + 1 < m$, the possible contact with P_{i+1} is the gray orientation: its cells in B_4 avoid that block's first column. Thus $N(T_i) \cap P_{i+1} \subseteq \{P_{i+1}(0)\}$. The shared endpoint may receive a grain, but every $P_j(l)$ with $i + 1 \leq j \leq m - 1$, $1 \leq l \leq k$, is excluded from that contact. Horizontal and vertical reflections give the other orientations and the negative case. These checks give the second target for the single-front passage; the preceding exterior-bound reservation prevents concluding the full induction here.

□

5.2 Semi-duplicators

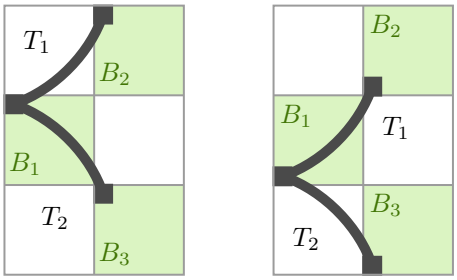
For the implementations below, let $Z_2 = HHHVV$.

Formally, given 3 blocks (B_1, B_2, B_3) included in a 3×2 block shape as in Fig. 12 and the positive (resp. neg.) bridges T_1, T_2 connecting B_1, B_2 and B_1, B_3, we define the positive (resp. neg.) semi-duplicator as the gadget $(D, c) = T_1 + T_2$. We call B_1^+ (resp. B_1^-) the input and B_2^+, B_3^+ (resp. B_2^-, B_3^-) the outputs of the semi-duplicator.

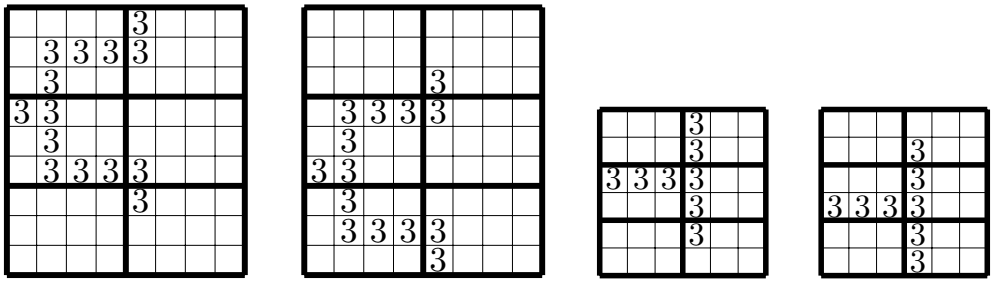
The functionality of the structure follows from the transport argument in the proof of Lemma 2: a bridge gadget formed by two source bridges with their source at the same cell produces both bridges to carry a signal, essentially duplicating it.

Corollary 2 *Let $D = T_1 + T_2$ be a positive semi-duplicator gadget with blocks (B_1, B_2, B_3). If T_1 or T_2 are source bridges, then $F(c, Z)(B_2^+) = F(c, Z)(B_3^+) = 4$.*

Proof The two bridges have the same source cell B_1^+, which is the only cell initially in state 4. For either path, all later cells have value 3 and the other bridge contributes only values at most 3 outside it. Apply the general cone induction from the proof of Lemma 2 to each path: lowering the common source gives a stable background; at each step the next path cell and its other neighbor in the update direction are beyond the preceding cone, so exactly one grain reaches the next cell. This argument uses only the background bound 3, without requiring the stronger affected-neighbor hypothesis of the lemma. It gives $F(c, Z)(B_2^+) = F(c, Z)(B_3^+) = 4$. □



(a) Blueprint for the positive and negative semi-duplicator. The bridge T_1 connects B_1 and B_2 and T_2 connects B_1 and B_3.



(b) Implementation of the positive and negative semi-duplicator in Z_1

(c) Implementation of the positive and negative semi-duplicator in Z_2

Figure 12: Blueprint and low-level implementation of the semi-duplicator gadget. At the top, the gadget is shown using the standard notation for bridges and the Z-blocks involved in the structure. Below, particular cases of the gadget are illustrated. The positive version of the gadget is always shown on the left of each figure.

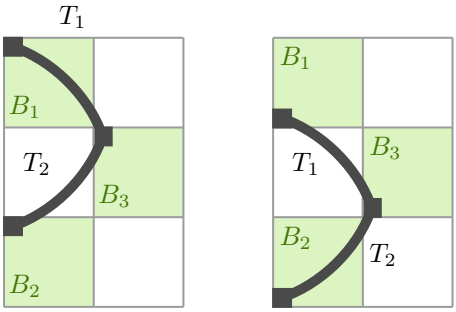
5.3 Semi-merges

Formally, given 3 blocks (B_1, B_2, B_3) included in a 3×2 block shape as in Fig. 13, and the positive bridges (resp., negative) T_1, T_2 connecting B_1, B_3 and B_2, B_3, we define the positive (resp., negative) semi-merge as the gadget $(M, c) = T_1 + T_2$. We call B_1^+, B_2^+ (resp. B_1^-, B_2^-) the *inputs* and B_3^+ (resp. B_3^-) the *output* of the semi-merge.

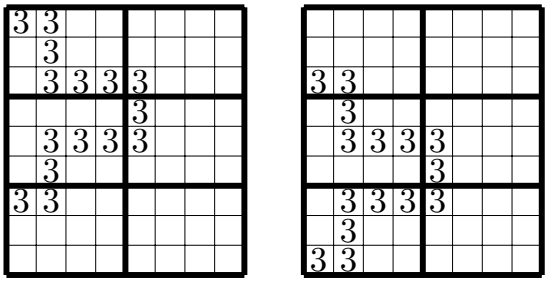
As with the semi-duplicator, the functionality of the structure follows from the transport argument in the proof of Lemma 2: this gadget, formed by two bridges with the same end cell, can carry a signal to that cell independently.

Corollary 3 *Let $(M, c) = T_1 + T_2$ be a positive semi-merge in (B_1, B_2, B_3). If exactly one of T_1 and T_2 is a source bridge, then $F(c, Z)(B_3^+) = 4$.*

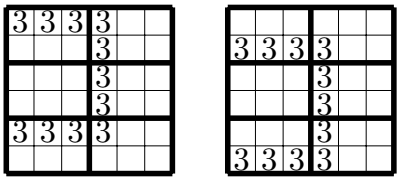
For use with two inputs, we assume their signals do not arrive simultaneously. If the other signal arrives later, a coordinator following the merge filters that second arrival under the two-pulse protocol of Lemma 10.



(a) Blueprint for the positive and negative semi-merge



(b) Implementation of the positive and negative semi-merge in Z_1



(c) Implementation of the positive and negative semi-merge in Z_2

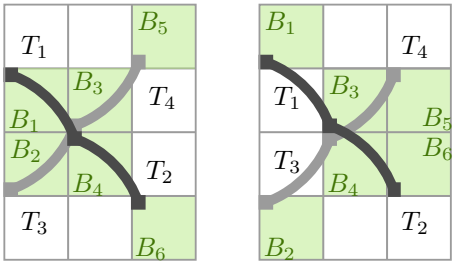
Figure 13: Blueprint and implementation of the semi-merge gadget. We use the same conventions as in Fig. 12.

Proof Apply the general cone induction in the proof of Lemma 2 to the source bridge among T_1, T_2: its initial cell is the only signal, its later cells have value 3, and all cells outside its path have value at most 3. The same transport argument therefore gives $F(c, Z)(B_3^+) = 4$. If both inputs receive a signal, the following coordinator semiwire covers the two-input case when the first arrival is coordinated and the second adds one grain at any later step, with no further contributions: Lemma 10 shows that only one usable signal is transmitted in the stated pure-path configuration. We use the interface as in the original construction. □

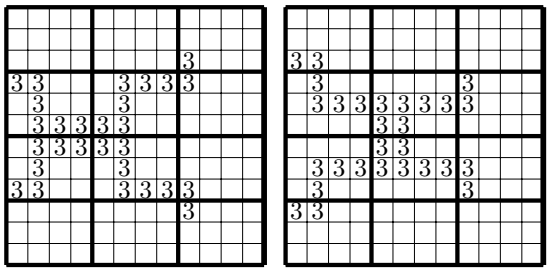
5.4 Semi-crossings

Semi-crossings provide a rudimentary crossing between signals of opposite polarity. We follow the blueprints and the construction of [19], reimplementing them with bridges as in Fig. 14. The local result below concerns the end of the second cycle.

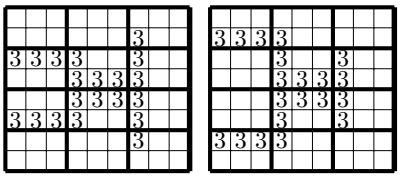
Consider 6 blocks $(B_1, ..., B_6)$ included in a 4×3 block shape as in Fig. 14, positive bridges T_1, T_2 connecting B_1, B_4 and B_4, B_6 and negative bridges T_3, T_4 connecting B_2, B_3 and B_3, B_5. We define a positive semi-crossing as the bridge gadget $(Q, c) = T_1 + T_2 + T_3 + T_4$. We call B_1^+, B_2^- the inputs of the semi-crossing, and B_5^-, B_6^+ its outputs.



(a) Blueprint for the positive and negative semi-crossing



(b) Implementation of the positive and negative semi-crossing in Z_1



(c) Implementation of the positive and negative semi-crossing in Z_2

Figure 14: Blueprint and implementation of the semi-crossing gadget. We follow the same conventions as in Fig. 12.

Lemma 5 *Let $Z \in \{H, V\}^{h,v,k}$, with $h, v \geq 1$, and consider the positive semi-crossing of Fig. 14. Write $T_r = (P_r, c_r)$ for $r \in \{1, 2, 3, 4\}$ and $Q = \bigcup_{r=1}^{4} P_r$. Initially, give every cell of Q value 3 and every cell outside Q value 0, except for exactly one of the two inputs, whose value is 4. The following conditions hold:*

- *If the unique source is $P_1(0) = B_1^+$, then $F^2(c, Z)(B_6^+) = 4$ and $F^2(c, Z)(B_5^-) < 4$.*
- *If the unique source is $P_3(0) = B_2^-$, then $F^2(c, Z)(B_5^-) = 4$ and $F^2(c, Z)(B_6^+) < 4$.*

For a configuration c, we say that c has the *vertical parity property* (vpp) if, for every pair of signals s_1, s_2 in c, we have that $s_1 - s_2 = (a, 2b)$ for some $a, b \in \mathbb{Z}$. Observe that this property is preserved by the dynamics. In fact we have the following lemma:

Lemma 6 *Let c be a configuration with the vpp. Then, $F(c, H), F(c, V)$ have the vpp.*

Proof The case of H transitions is simple: Since $F(\cdot, H)$ only moves signals along the same row, the configuration $F(c, H)$ necessarily has the vpp.

For the case of V transitions, note that, since the vpp holds on c, only the rows of a given parity contain signals. Without restriction, assume these are rows with an even coordinate. Through a transition $F(\cdot, V)$, all signals topple simultaneously, thus distributing grains to

rows of odd parity. Because the rows of odd parity initially contained no signals, all rows with even parity on $F(c, V)$ are now signal-free, and therefore the vpp holds. □

Proof of Lemma 5 Translate the grid so that $B_1^+ = (0, 0)$, and let $h_t = |Z(0) \cdots Z(t-1)|_H$ and $v_t = |Z(0) \cdots Z(t-1)|_V$ for $0 \le t \le k$. The blueprint and the definition of a Z-path give

$$P_1(t) = (h_t, v_t), \quad P_2(t) = (h{+}h_t, v{+}v_t), \quad P_3(t) = (h_t, 2v{-}1{-}v_t), \quad P_4(t) = (h{+}h_t, v{-}1{-}v_t).$$

In particular, $P_1(k) = P_2(0)$ and $P_3(k) = P_4(0)$. For source $P_1(0)$, the paths P_1, P_2 have the same orientation, meet at $P_1(k) = P_2(0)$, and have value 3 except for that source, whose value is 4. All exterior values are at most 3. Thus Corollary 1 gives $F^2(c, Z)(B_6^+) = 4$. For source $P_3(0)$, the same corollary applies to P_3, P_4, which share the opposite vertical orientation and meet at $P_3(k) = P_4(0)$, giving $F^2(c, Z)(B_5^-) = 4$.

It remains to exclude the other output at the end of the second cycle. The row coordinates of the two outputs satisfy

$$B_6^+ = (2h, 2v), \qquad B_5^- = (2h, -1), \qquad B_5^- = B_6^+ - (2v+1)V.$$

With the first input selected, its initial row is even and the other output has odd row; with the second input selected, its initial row $2v-1$ is odd and the other output has even row. By the parity tracking in the proof of Lemma 6, each vertical update switches the row parity of the signals. After two cycles there have been $2v$ such updates, an even number, so every signal has its initial row parity. The other output has the opposite parity and therefore has value less than 4 in $F^2(c, Z)$. □

We define the negative semi-crossing gadget as shown in Fig. 14. The difference is in the polarities of the upper input and the bottom one: The positive semi-crossing has the positive signal input on top, while the negative semi-crossing has it below.

For the negative blueprint the same conclusions at the end of two cycles hold with the upper input B_1^- selecting output B_6^-, and the lower input B_2^+ selecting output B_5^+, under the same initial assignment of 3 on the paths, 0 outside them, and 4 at exactly one input. The absence of signals at the other output follows from Lemma 6 after exactly two cycles.

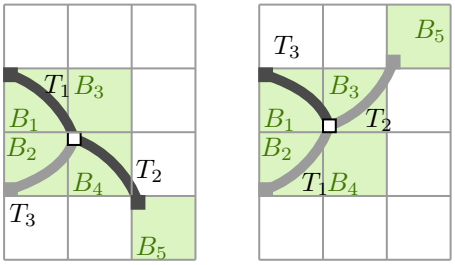
5.5 Switches

The constructions of the positive and negative switches differ slightly but follow the same principle. The polarity of the switch is determined by the polarities of the upper bridges. Like the semi-crossing, its inputs/outputs have different polarities.

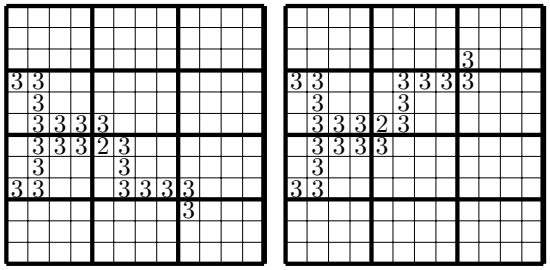
5.5.1 Positive switch.

For the positive switch, we arrange 5 blocks $(B_1, ..., B_5)$ included in a 4×3 block shape as in Fig. 15, positive bridges T_1, T_2 connecting B_1, B_4 and B_4, B_5 and a negative bridge $T_3 = (P_3, c_3)$ with a sink at $k-1$ connecting B_2, B_3. We define a positive off switch as the bridge gadget $(S, c_{\text{off}}) = T_1 + T_2 + T_3$. We call B_1^+, B_2^- the inputs of the switch, and B_5^+ its output.

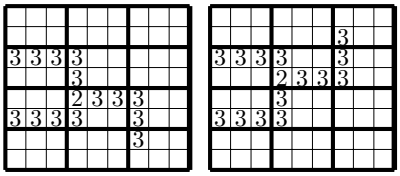
Lemma 7 (Local positive switch) *Suppose $h, v \ge 2$, $Z(0) = H$, and $Z(k-1) = V$. Consider the positive switch in Fig. 15 in its pristine configuration c_{off}: the shared sink $R = P_1(k) =$*



(a) Blueprint for positive and negative switches. The bridges T_1, T_2 connect B_1 to B_4 and B_4 to B_5 in the positive switch, while bridges T_1, T_2 connect B_2 to B_3 and B_3 to B_5 in the negative one. The bridge with a sink (denoted as a white dot) T_3 connects B_2 to B_3 in the positive switch, whereas the bridge with a sink (denoted as a white dot) T_3 connects B_1 to B_4 in the negative one.



(b) Implementation of positive and negative switches in Z_1



(c) Implementation of positive and negative switches in Z_2

Figure 15: Blueprint and implementation of the switch gadget. Note that, in this case, bridge T_3 does not end at the block it would normally reach if it were not a bridge with a sink.

$P_3(k-1)$ has state 2, all other cells of $P_1 \cup P_2 \cup P_3$ have state 3, and all cells outside that union have state 0. A pulse means the addition of one grain at its input pin, at the start of a cycle. Make no external additions other than the specified pulses. Then the following hold.

- *If there is only a source pulse at B_1^+ at time $t_s = q_s k$, the configuration has no signals after one cycle, at $t_s + k$, and remains stable. In particular it has no signals at $t_s + 2k$.*
- *A gate pulse alone at B_2^- at time $t_g = q_g k$ prepares, by $t_g + k$, a stable configuration with state 3 throughout $P_1 \cup P_2$ and state at most 3 elsewhere.*
- *After that gate pulse, if a source pulse is added at $t_s = q_s k$ with $q_s > q_g$, the output B_5^+ has state 4 at $t_s + 2k$. The prepared configuration in the preceding item is understood before the source addition when $t_s = t_g + k$.*

Here q_s, q_g are nonnegative integers; the first item has no gate pulse, and the last item has exactly one gate pulse followed by exactly one source pulse.

Proof The argument follows the functionality of bridges with a sink. For a source pulse alone, the geometry of the blueprint and the first and last letters of Z ensure that P_1 sends no grain to $P_3 \setminus P_1$ or $P_2 \setminus P_1$ before reaching R. Thus its affected exterior is initially zero, and Lemma 3 applies to the effective bridge along P_1, with sink $R = P_1(k)$. Its proof gives no signals after one cycle and stability thereafter, proving the first item.

For a gate pulse alone, the only main-path cell reached by the firing prefix $P_3(0), \ldots, P_3(k-2)$ is $R = P_3(k-1) = P_1(k)$. The superposed tail $P_3(k) = P_1(k-1)$ has state 3, but is at distance at least two from every cell of that prefix, so it receives no grain. The same sink argument therefore applies up to R: the signal raises its state from 2 to 3 at step $k-1$ and disappears. As in the proof of Lemma 3, each preceding gate cell receives at most one return grain and each exterior cell at most one grain. Hence there are no second firings or exterior relays. All other cells of $P_1 \cup P_2$ remain at 3, proving the second item and stability of the prepared configuration.

For the third item, $q_s > q_g$ ensures that this configuration persists until the source pulse. After that addition, P_1, P_2 have their common orientation, state 3 except for the source of value 4, and exterior states at most 3. Applying Corollary 1 gives state 4 at B_5^+ at time $t_s + 2k$. □

5.5.2 Negative switch.

For the negative switch, given 5 blocks $(B_1, ..., B_5)$ included in a 4×3 block shape as in Fig. 15, negative bridges T_1, T_2 connecting B_2, B_3 and B_3, B_5 and a positive bridge $T_3 = (P_3, c_3)$ with a sink at $k-1$ connecting B_1, B_4. We define a negative off switch as the bridge gadget $(S, c_{\text{off}}) = T_1 + T_2 + T_3$. We call B_1^+, B_2^- the inputs of the switch, and B_5^- its output.

Lemma 8 (Local negative switch) *Under the word, isolation, and pulse hypotheses of Lemma 7, the negative switch has the same conclusions, with source input B_2^-, gate input B_1^+, shared sink B_3^-, and output B_5^-. In particular, a source pulse without a gate pulse leaves no signals after one cycle, a gate pulse alone prepares the two main paths in one cycle, and a source pulse in a strictly later cycle reaches the output with state 4 after two further cycles.*

Proof The proof is analogous to the positive case (Lemma 7). □

Observe that the case in which two signals arrive at the same time is not interesting to analyze since we can control in our construction that this particular case never happens.

5.6 Retarders

The retarder in Fig. 16 delays a signal as it travels from left to right, with the same functionality as in [19].

5.7 Connecting gadgets

Recall that Z is primitive: there are no $a > 1$ and nonempty word U such that $Z = U^a$. This assumption ensures that the structure eliminating uncoordinated signals functions

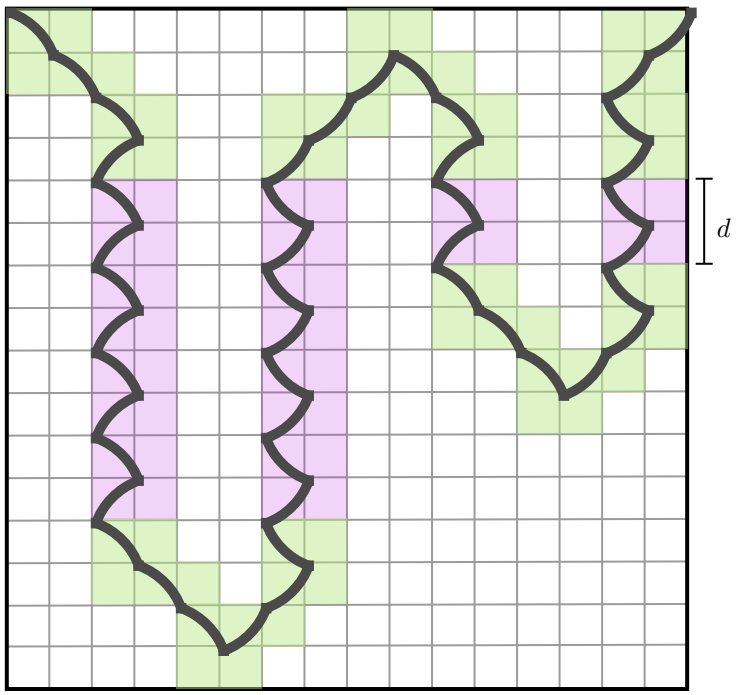


Figure 16: Positive retarder formed by bridges connecting diagonally adjacent blocks. The marked Z-blocks indicate those that may contain grains in this gadget. The purple blocks represent sections of the retarder that could vary.

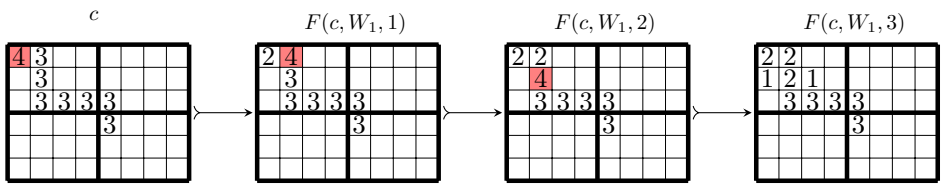


Figure 17: Bridge failure under updates following a shifted cycle. Signals are highlighted in red. Compare Fig. 1 for $Z_1 = HVVHHHV$: for $W_1 = HV|HVVHH$, the configuration $F(c, W_1, 3)$ has no signals. The first index where Z_1 and W_1 differ is 2, so the signal disappears at the third update.

correctly. An *uncoordinated signal* is a signal appearing at a gadget's input at a time other than the start of a cycle.

Consider a subdivision $Z = XY$, with X, Y nonempty, and the *shifted Z-cycle* $W = YX$. The dynamics of W are the same as those of Z, except that they start "out of sync". Since Z is primitive, every such nontrivial rotation differs from Z. An example appears in Fig. 17, where the bridge ceases to function properly. We show below that a signal cannot traverse a pure coordinator under the updates of W.

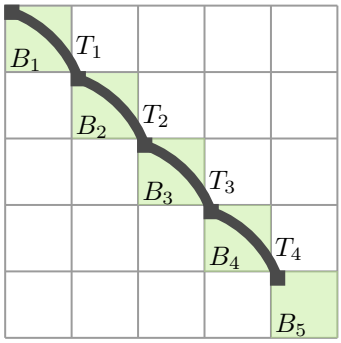


Figure 18: Structure of a coordinator semiwire consisting of four bridges $T_1, \ldots, T_4$.

A positive coordinator semiwire consists of a constant number of bridges in the same direction, as shown in Fig. 18. The following lemma concerns a pure coordinator with one initial signal and no other inputs.

Lemma 9 (Shifted cycles on a coordinator) *Let $(S, c) = \sum_{j=1}^{m} T_j$, where $T_j = (P_j, c_j)$ and $m \geq 1$, be a pure coordinator semiwire for a primitive word Z of length k, with T_1 its only source bridge: its initial cell has value 4, every other path cell has value 3, and all cells outside the path have value 0. For every nontrivial shifted Z-cycle W, there is $t < km$ such that $F(c, W, t)$ has no signals.*

Proof Every path cell with value 3 has two adjacent path cells, except for the endpoints. The initial signal at $P_1(0)$ has only one neighbor with value 3. After any update $Q \in \{H, V\}$, $P_1(0)$ has value 2, and its neighbor can become a signal only if the update follows the path. Thus there is at most one signal, at $P_1(1)$, whose preceding neighbor now has value 2. Continuing this reasoning, until the first disagreement between W and Z, there is exactly one signal at the next path cell; its preceding neighbor has value 2, and its successor has value 3.

Let d be the first index with $W(d) \neq Z(d)$. It exists because Z is primitive. In $F(c, W, d)$ the signal has its successor in direction $Z(d)$, but the next update uses the other direction. It therefore sends grains only to cells of value at most 2, and $F(c, W, d+1)$ has no signals. Since W and Z have the same letter multiplicities, their first disagreement cannot be their last index. Hence $d + 1 < k \leq km$. □

For two arrivals, the coordinator transmits only the first signal under the following isolated-path hypotheses. The second arrival may occur at any later step, without a fixed separation or a cycle-boundary condition.

Lemma 10 (Two pulses on a coordinator semiwire) *Let $Z \in \{H, V\}^k$ contain both letters, and let $(S, c) = T_1 + \cdots + T_m$, where $T_j = (P_j, c_j)$ and $m \geq 1$ is constant, be a positive or negative coordinator semiwire, with all bridges in the same diagonal direction as in Fig. 18. Its input is $P_1(0)$ and its output is $P_m(k)$. Let T_1 be its only source bridge: initially $c(P_1(0)) = 4$, every other cell of S has value 3, and all cells outside S have value 0. Apply the updates of Z periodically, so the first signal is coordinated. At an integer time $\tau \geq 1$, add exactly one grain*

at $P_1(0)$ before the next update; make no other external additions. Then $P_m(k)$ is a signal exactly once, at time mk, when its state is 4. The signal, if any, produced by the second addition is extinguished before the first turn of T_1. Every cell outside S remains in state at most 2.

Proof Write $N = mk$ and enumerate the cells of this coordinator by concatenating its bridge paths: $P((a-1)k+s) = P_a(s)$ for $1 \le a \le m$ and $0 \le s \le k$. Thus $P(0) = P_1(0)$, $P(N) = P_m(k)$, and the cells of P are exactly S. Since all bridges have the same diagonal direction, horizontal and vertical reflections, which commute with the H and V updates, reduce the proof to $P(j+1) = P(j) + Z(j \bmod k)$ for $0 \le j < N$. This includes both polarities. Put $e_j = Z(j \bmod k)$, including $e_N = Z(0)$. Without the second addition, induction gives the unique signal $P(t)$ in state 4 at each $0 \le t \le N$; after its firing, the preceding path cell can receive at most one return grain. As in the path geometry in the proof of Lemma 3, each exterior cell receives at most one grain during this passage: if it has two path neighbors, they are $P(a), P(a+2)$ and the first fires perpendicular to the direction toward that exterior cell. There are therefore no exterior relays or second firings, and $P(N)$ fires at the next update, leaving no signal. For $0 \le j < N$, the settled residue is

$$2 + [e_j = e_{j+1}],$$

since only the successor can return a grain after $P(j)$ fires.

Let r be the length of the initial run of $E = Z(0)$, so $1 \le r < k$. If $r = 1$, $P(0)$ stays at 2 after its first firing and receives no return grain; the addition raises it only to 3. Suppose $r \ge 2$. An addition at $\tau = 1$ raises $P(0)$ from 2 to 3, and the return grain from $P(1)$ raises it to 4 at time 2. Its configuration then agrees with that obtained by adding the grain at $\tau = 2$. Thus the second signal starts at $P(0)$ at time $\lambda = \max\{2, \tau\}$.

As long as the next updates are E, this second signal can move only along the initial straight segment. If the first signal is still present, its path index exceeds the second's by $\lambda \ge 2$. Once the first signal has disappeared, all the residues have settled. For a separation greater than 2, the next cell has its settled residue: 3 before the first turn, and 2 at $P(r-1)$. For separation 2, that cell has just fired and has value 2; an E update gives it one grain from each signal, so it becomes 4. Both signals advance, with the same separation. Cells left behind have value at most 3 after any return grain, so no third signal is created.

If an update different from E occurs while the second signal is present, it fires perpendicular to the initial segment and disappears. Otherwise it reaches the cell preceding $P(r-1)$ and raises the settled value 2 at $P(r-1)$ only to 3. In the separation-2 case the perpendicular update occurs first, when the first signal has reached $P(r)$. Thus the second signal never passes the initial run. Each exterior cell receives at most one extra grain from this straight passage and its possible final perpendicular firing: the straight firing segment cannot send grains to the same exterior cell twice. It hence remains at most 2. There are no exterior relays, the two signals are never adjacent, and these bounds close the induction for their joint evolution. The first passage is unchanged, and $P(N) = P_m(k)$ is a signal exactly once, at time $N = mk$. Since $r < k \le N$, the extinction before the first turn occurs within T_1, including when $m = 1$. □

We refer to the time it takes for a signal to travel from a gadget's input to the corresponding output as its *delay*.

Connections are made as in the examples in Figs. 19 and 20. For gadgets with two inputs, the semiwire of the upper input arrives from above and the semiwire of the lower input arrives from below.

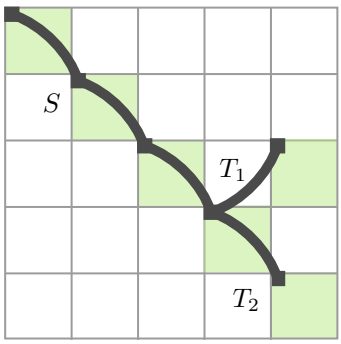


Figure 19: Correct connection of a coordinator semiwire S to a semi-duplicator composed of the bridges T_1, T_2.

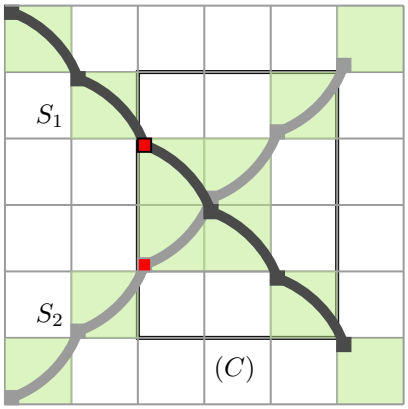


Figure 20: Correct connection of two semiwires S_1 and S_2, each consisting of at least two bridges, to a semi-crossing gadget C, which is contained within the eight Z-blocks marked by the black rectangle. The cells marked in red indicate positions that may contain a signal, depending on whether the corresponding semiwire carries one.

5.8 Diodes

The positive diode in Fig. 21 is formed by positive bridges connecting the blocks $B_0, \ldots, B_{13}$. Its input is $I = B_0^+$ and its output is $O = B_{13}^+$. The cell B_{10}^+ initially has value 2, obtained by making the bridge from B_8 to B_{10} a bridge with a sink at its endpoint.

The diode has the same functionality as in [19].

In the positive diode of Fig. 21, only coordinated signals can travel from the input I to the output O, while a signal starting at O cannot reach I. A coordinated signal arriving at I travels to B_3^+, where it is duplicated. The two signals reach B_{10}^+, creating a signal there despite its initial value 2, and this signal travels to O. In the reverse direction, a signal starting at O that passes through B_{11} is consumed at B_{10}^+. The negative diode is constructed analogously.

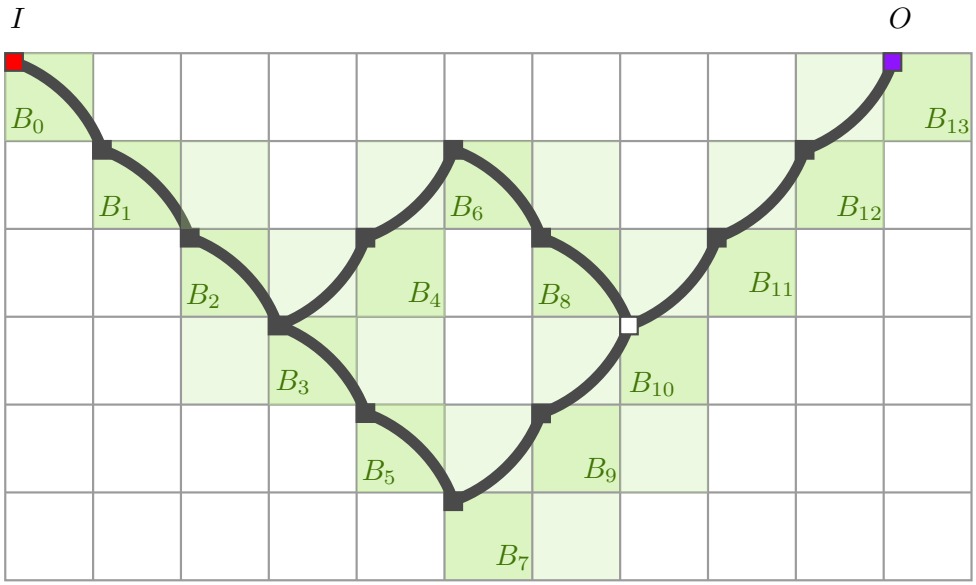


Figure 21: Positive diode with input cell I in red and output cell O in purple, formed by positive bridges connecting blocks $B_0, \ldots, B_{13}$. The white square represents the cell B_{10}^{+} with initial value 2, the sink at the endpoint of the bridge from B_8 to B_{10}.

6 Layer 3: Boolean Circuits with Delay

Layer 3 supplies the circuit abstraction used by the logspace constructor: *Boolean circuits with delay*, whose gates process bits and are connected by cables consisting of a positive and a negative wire. Cable lengths entail propagation delays, so arrival times must be taken into account. For the description of this layer in terms of polarized components, we refer to [19, Section 5].

7 Layer 4: The Logspace Constructor

For the logarithmic-space construction we follow an analysis similar to that of [19].

The layout template is that of [19, Section 6]. Let B describe a circuit with n inputs and m gates $G_1, \ldots, G_m$, and let F denote the finite configuration. Input cables are denoted by $x_1, \ldots, x_n$, and gate-output cables by $g_1, \ldots, g_m$. Tile U_i contains gate G_i; its upper part routes cables and its lower part contains the gate implementation. We retain D for the layout's delay parameter.

The complexity of the construction now lies in the fact that the constructor must determine in which diagonally adjacent Z-blocks a bridge must be placed. This introduces an additional layer of complexity where the constructor is also required to explicitly build the bridges between such blocks, compared to the previous construction, where it directly adds cells onto the 2×2 subdivision if necessary.

Consider an instance of the CVP with m gates. The description of F is column by column. At each step, we keep track of the coordinates we are currently at and return

them as needed. The size of the structure is polynomial in m, so its coordinates can be stored using logarithmic space. Before initializing the construction, we compute and store D. Since $D = \Theta(m^4)$, this can be done using logarithmic space by examining the end of the input to obtain m, which is also stored in logarithmic space.

Before describing each tile, we must define the inputs. Knowing D, since each cable maintains a fixed height in each tile that only depends on D, we construct the cables $x_1, \ldots, x_n$ at the corresponding heights, assigning positive or negative signals as appropriate.

Given a circuit encoded node by node, the construction of F proceeds tile by tile. Since all tiles have the same size, we can calculate the relative coordinates of each tile and, from there, assemble the cables and crossings at the corresponding positions.

To construct tile U_i, we need to know:

- the number of cables in the upper part;
- the delays for the crossings;
- the locations of the crossings;
- the corresponding gate G_i.

The number of cables in the upper part comes directly from the tile number. There are $n + i - 1$ incoming cables and $n + i$ outgoing cables. The position and length of the cable that descends to the gate depend only on the number of cables in the tile, which is known.

The delays required for each crossing and NAND gate are computed sequentially from the prescribed schedule: first the entry delay d_{U_i} of the tile, then the entry delay d_{C_j} of each crossing C_j, and then the delay d_{N_r} of each NAND gate N_r of that crossing. The template bounds these quantities by an entry delay of order $O(imD)$, an additional delay of order $O(jD)$ from the tile entry to crossing C_j, and an additional delay of order $O(rD)$ from that crossing's entry to N_r. At each step we store these three exact delays. Since they are polynomially bounded in m in the template, they require only logarithmic space.

The possible crossing positions are fixed in U_i; we only need to determine where crossings actually occur. This is obtained directly from the encoding: each gate has its two inputs y_1, y_2 specified there, and all cables below these two must have crossings.

The type of gate in the tile is determined from the input encoding. The necessary delay for its NAND gates has order $O(iD)$ in the template and is computed from the schedule. Its position in the tile is fixed, so calculating the delays of its NAND gates supplies the remaining data for constructing it.

Let p_{out} be the final positive-wire pin of output cable g_m at tile U_m. For $Z(0) = H$, the target cell is $q = p_{\text{out}} + H$, unused by the initial layout and initially in state 0.

Finally, we describe the coordinates of the target cell q adjacent to the positive output pin of the last gate. For the choice of the time bound T and its proof, we follow the original construction of [19, Section 6.4]: the waiting time is obtained from the exit time of the last tile, including the update toward q. It is expressed in individual updates, with k updates per Z-cycle. The time is stored in binary during the construction and written in unary in the output.

8 Proof of the Main Theorem

To conclude, we put together the complete proof of our result.

Theorem 1 *If $Z \in \{H, V\}^+$ is any word that contains at least one H and one V, then the fungal sandpile prediction problem associated with Z is* P*-complete.*

Proof The proof proceeds via an efficient reduction from the CVP to the fungal sandpile prediction problem associated with Z. Given an instance B of the CVP, we compute the triple (R, x, T).

The region R is a finite grid with configuration c that follows the same tile scheme described in **Layer 4** of [19], with the following modifications:

- The subdivision is performed into blocks whose size depends on Z (instead of a 2×2 subdivision), as described in Section 3.
- All gadgets introduced in [19] are replaced by the generalized gadgets defined in Section 5, and the delay is redefined as in that section.

The parameters x and T are chosen analogously to [19], taking into account the adaptations above. The correctness of the reduction follows directly from the functionality of each gadget stated in Section 5 and the overall correctness stated in [19].

Finally, the reduction is computable in logarithmic space by the arguments given in Section 7.

□

9 Discussion

We now briefly discuss what the result means for the case of ordinary sandpiles. If one could show how to implement fungal sandpiles in ordinary sandpiles, then of course our result would immediately imply hardness in that case as well. Unfortunately, however, no reduction between the two problems is presently known.

That being said, there are still meaningful inferences to be made. We focus here on points that were not already made by Modanese and Worsch in their work [19].

1. (As is the case with the work of Modanese and Worsch) the construction relies heavily on the fact that we have alternations between horizontal and vertical phases, which lets us induce *signals* of different polarities (positive and negative). This distinction is fundamental for the construction to function since the signals behave distinctly. In our construction, polarity is maintained by bridges that connect equal-polarity positions in successive blocks and by semiwires built from these bridges.

It is already known how to embed circuits that are monotone and planar [18], so one could consider similarly restricted classes of circuits.

Finally, an interesting approach that has been explored in recent years is to study the two-dimensional prediction problem through the structural properties of the underlying neighborhoods. In [8], the authors analyze sandpile dynamics under all possible sub-neighborhoods of the Moore neighborhood, identifying a distinction

between neighborhoods that admit P-complete prediction problems and those that fundamentally lack the ability to transmit crossing information.

On the other hand, it has been noted that sandpile models are closely related to majority cellular automata, and despite the latter not being number-conserving, both models exhibit remarkably similar open questions concerning two-dimensional prediction complexity [3]. In light of this connection, it is particularly interesting that [21] studies freezing majority cellular automata on L-shaped neighborhoods, proving that some minimal neighborhoods yield prediction problems in NC, while larger neighborhoods become P-complete. A natural next step would be to study L-shaped neighborhoods in the context of sandpile automata, potentially extending these previous results and providing further insight into the complexity transition in two-dimensional prediction problems.

Acknowledgements. During the preparation of this manuscript, OpenAI's ChatGPT/Codex tools were used to assist with language editing, manuscript organization, LaTeX preparation, exploratory computational checks, and the drafting and review of mathematical arguments. The authors remain responsible for verifying all mathematical statements and for the final content of the manuscript.

Declarations

Competing interests. The authors declare that they have no competing interests relevant to the content of this article.

Data availability. This article reports theoretical results and has no associated research datasets to share.

References

[1] Bak, P., Tang, C., Wiesenfeld, K.: Self-organized criticality: An explanation of the 1/f noise. Physical Review Letters **59**(4), 381–384 (1987) https://doi.org/10.1103/PhysRevLett.59.381

[2] Dennunzio, A., Guillon, P., Masson, B.: Sand automata as cellular automata. Theoretical Computer Science **410**(38–40), 3962–3974 (2009) https://doi.org/10.1016/j.tcs.2009.06.016

[3] Formenti, E., Perrot, K.: How hard is it to predict sandpiles on lattices? a survey. Fundamenta Informaticae **171**(1-4), 189–219 (2020) https://doi.org/10.3233/FI-2020-1879

[4] Goles, E., Tsompanas, M.-A., Adamatzky, A., Tegelaar, M., Wosten, H.A., Martínez, G.J.: Computational universality of fungal sandpile automata. Physics Letters A **384**(22), 126541 (2020) https://doi.org/10.1016/j.physleta.2020.126541

[5] Adamatzky, A., Goles, E., Martínez, G.J., Tsompanas, M.-A., Tegelaar, M., Wosten, H.A.: Fungal automata. In: Adamatzky, A. (ed.) Fungal Machines: Sensing

and Computing with Fungi. Emergence, Complexity and Computation, vol. 47, pp. 323–339. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-38336-6_22

[6] Sepúlveda, C.S., Goles, E., Ríos-Wilson, M., Adamatzky, A.: Exploring dynamics of fungal cellular automata. In: Adamatzky, A. (ed.) Fungal Machines: Sensing and Computing with Fungi. Emergence, Complexity and Computation, vol. 47, pp. 341–370. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-38336-6_23

[7] Goles, E., Montealegre, P., Salo, V., Törmä, I.: PSPACE-completeness of majority automata networks. Theoretical Computer Science **609**, 118–128 (2016) https://doi.org/10.1016/j.tcs.2015.09.014

[8] Concha-Vega, P., Goles, E., Montealegre, P., Perrot, K.: Sandpiles prediction and crossover on $\mathbb{Z}^2$ within Moore neighborhood. Natural Computing **24**(1), 29–66 (2025) https://doi.org/10.1007/s11047-024-10002-9

[9] Ollinger, N., Theyssier, G.: Freezing, bounded-change and convergent cellular automata. Discrete Mathematics & Theoretical Computer Science **24**(1) (2022) https://doi.org/10.46298/dmtcs.5734

[10] Nakar, Y., Ron, D.: Characterizing and Testing Configuration Stability in Two-Dimensional Threshold Cellular Automata. arXiv:2507.14569 [cs.DS] (2025). https://doi.org/10.48550/arXiv.2507.14569

[11] Goles, E., Montealegre, P.: The complexity of the asynchronous prediction of the majority automata. Information and Computation **274**, 104537 (2020) https://doi.org/10.1016/j.ic.2020.104537

[12] Goles, E., Maldonado, D., Montealegre, P., Ríos-Wilson, M.: On the complexity of asynchronous freezing cellular automata. Information and Computation **281**, 104764 (2021) https://doi.org/10.1016/j.ic.2021.104764

[13] Goles, E., Montealegre, P., Perrot, K.: Freezing sandpiles and Boolean threshold networks: Equivalence and complexity. Advances in Applied Mathematics **125**, 102161 (2021) https://doi.org/10.1016/j.aam.2020.102161

[14] Ríos-Wilson, M., Theyssier, G.: Intrinsic universality in automata networks I: Families and simulations. Theoretical Computer Science **997**, 114511 (2024) https://doi.org/10.1016/j.tcs.2024.114511

[15] Ríos-Wilson, M., Theyssier, G.: Intrinsic universality in automata networks II: Glueing and gadgets. Theoretical Computer Science **1016**, 114779 (2024) https://doi.org/10.1016/j.tcs.2024.114779

[16] Ríos-Wilson, M., Theyssier, G.: Intrinsic universality in automata networks III: On symmetry versus asynchrony. Theoretical Computer Science **1022**, 114890 (2024) https://doi.org/10.1016/j.tcs.2024.114890

[17] Miltersen, P.B.: The computational complexity of one-dimensional sandpiles. Theory of Computing Systems **41**(1), 119–125 (2007) https://doi.org/10.1007/s00224-006-1341-8

[18] Moore, C., Nilsson, M.: The computational complexity of sandpiles. Journal of Statistical Physics **96**(1–2), 205–224 (1999) https://doi.org/10.1023/A:1004524500416

[19] Modanese, A., Worsch, T.: Embedding arbitrary Boolean circuits into fungal automata. Algorithmica **86**(7), 2069–2091 (2024) https://doi.org/10.1007/s00453-024-01222-7

[20] Goles, E., Modanese, A., Ríos-Wilson, M., Ruiz-Tala, D., Worsch, T.: Embedding arbitrary Boolean circuits into fungal automata with arbitrary update sequences. CoRR **abs/2602.19477** (2026) https://doi.org/10.48550/arXiv.2602.19477 2602.19477

[21] Concha-Vega, P., Goles, E., Montealegre, P., Perrot, K.: Complexity of the freezing majority rule with L-shaped neighborhoods. Theoretical Computer Science **1075**, 115938 (2026) https://doi.org/10.1016/j.tcs.2026.115938