Integrated Information Theory of Consciousness in Conventional Computing:

How a program impacts the causal structure of a computer

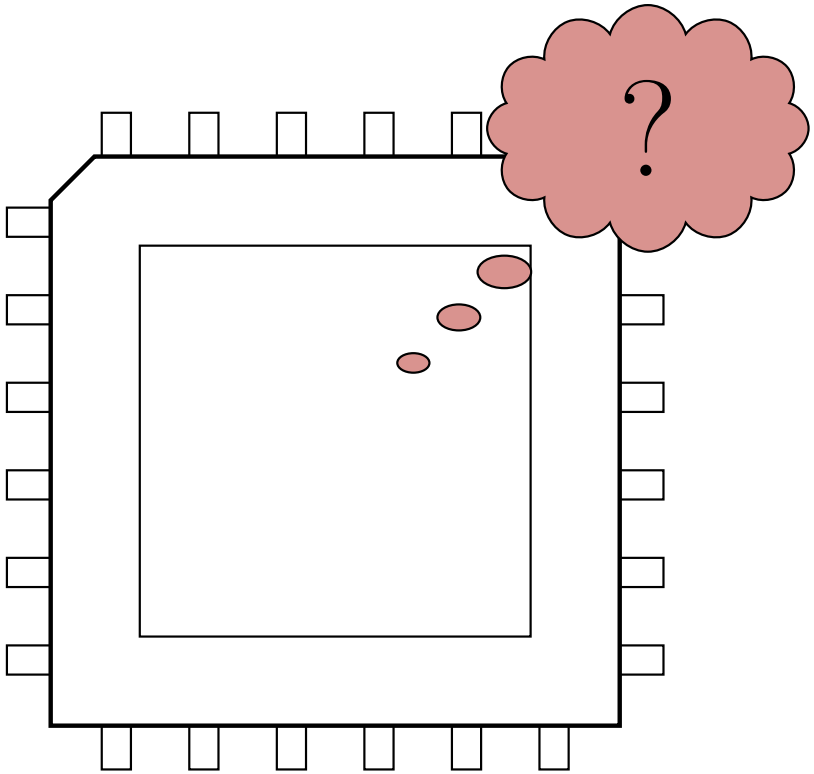
Justin T. Sampson

San Francisco State University
& VMware, Inc.

Thesis Co-Advisors
Larissa Albantakis
Pooyan Fazli

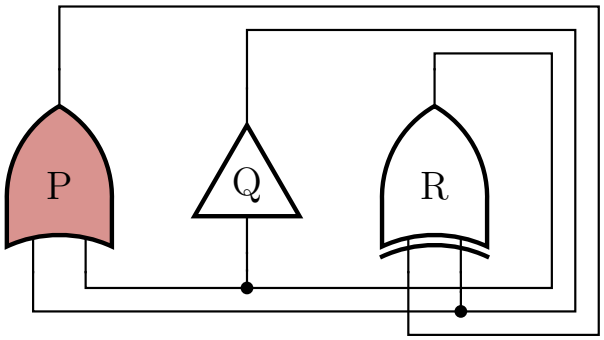
Models of Consciousness 2022

Can a Computer Ever Be Conscious?*

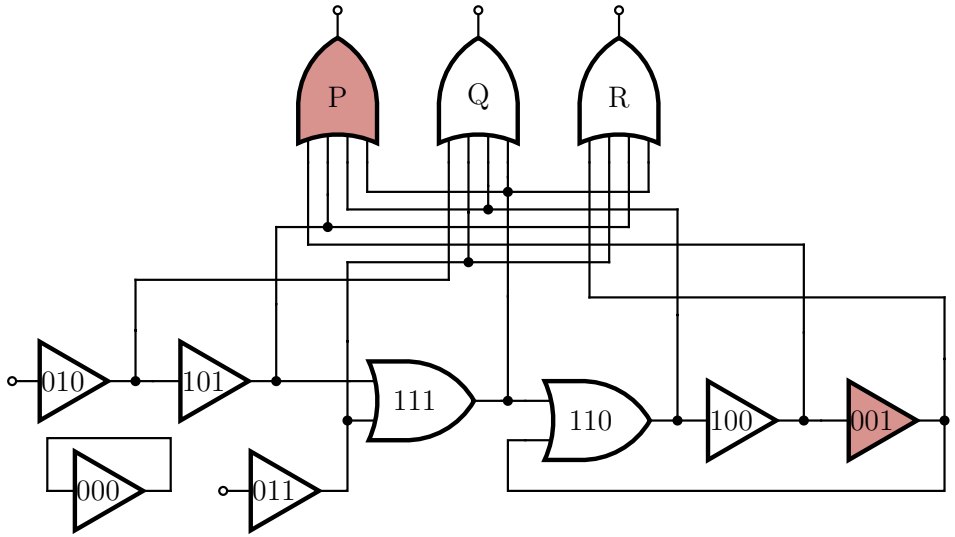


*responding to Findlay, Marshall, et al. (2019), a summary of which is available at http://ceur-ws.org/Vol-2287/short7.pdf

Causal* Structure vs. Functional Equivalence



$\Phi = 2.31$

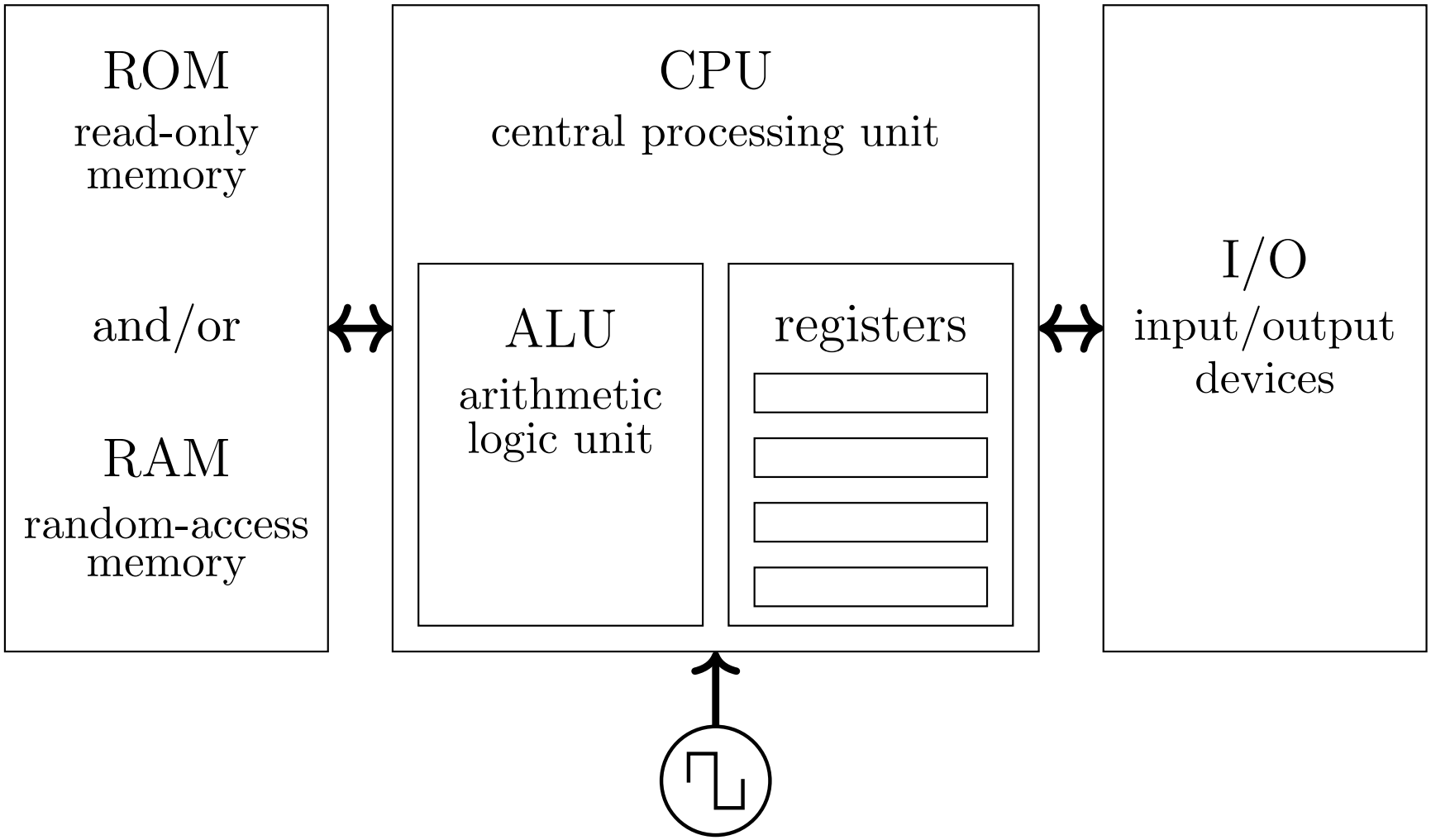


$\Phi = 0$ (entire circuit)

$\Phi = 1$ (three-gate ring)

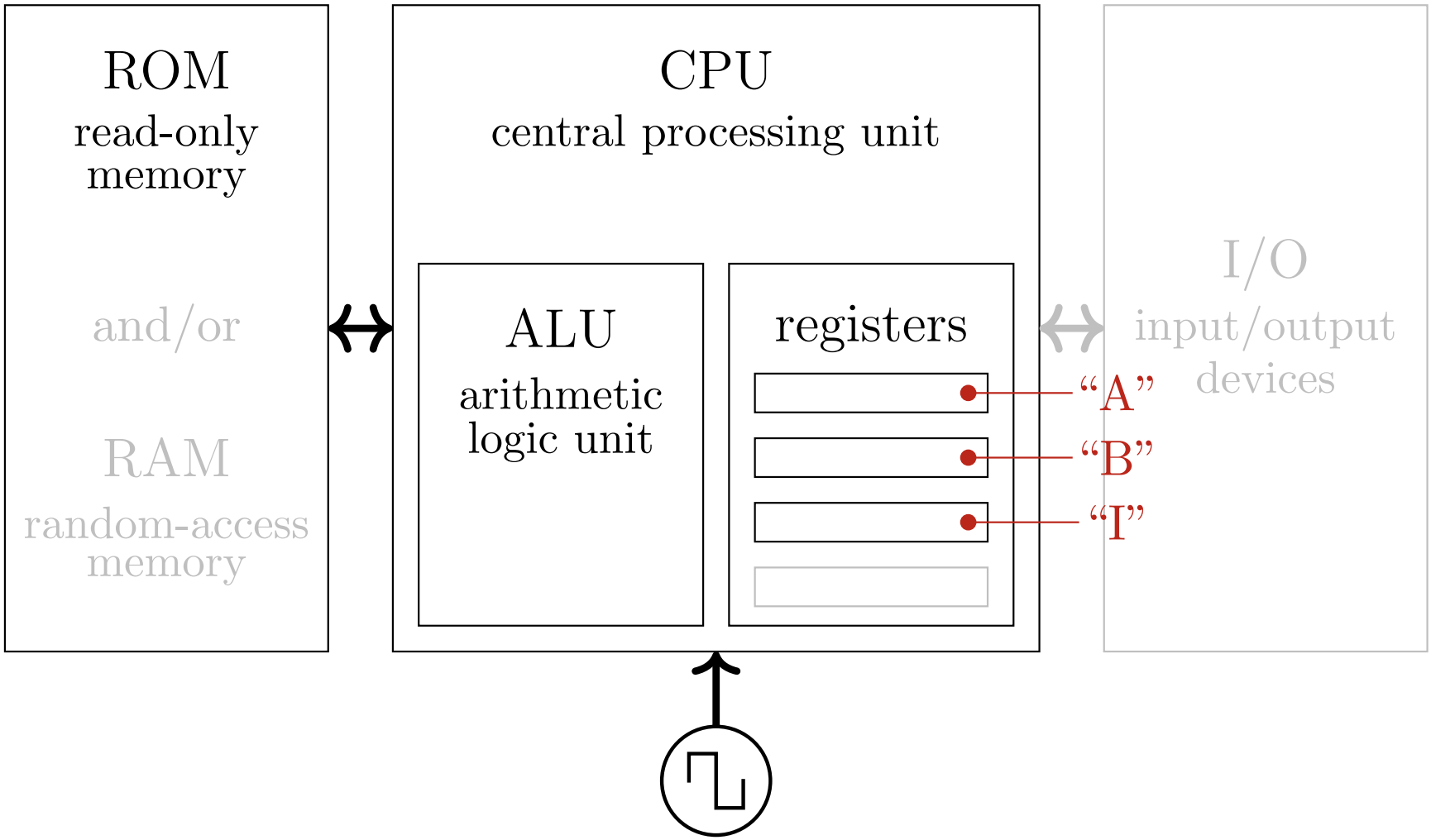
*and therefore *phenomenal* according to IIT

Conventional* Computer Architecture



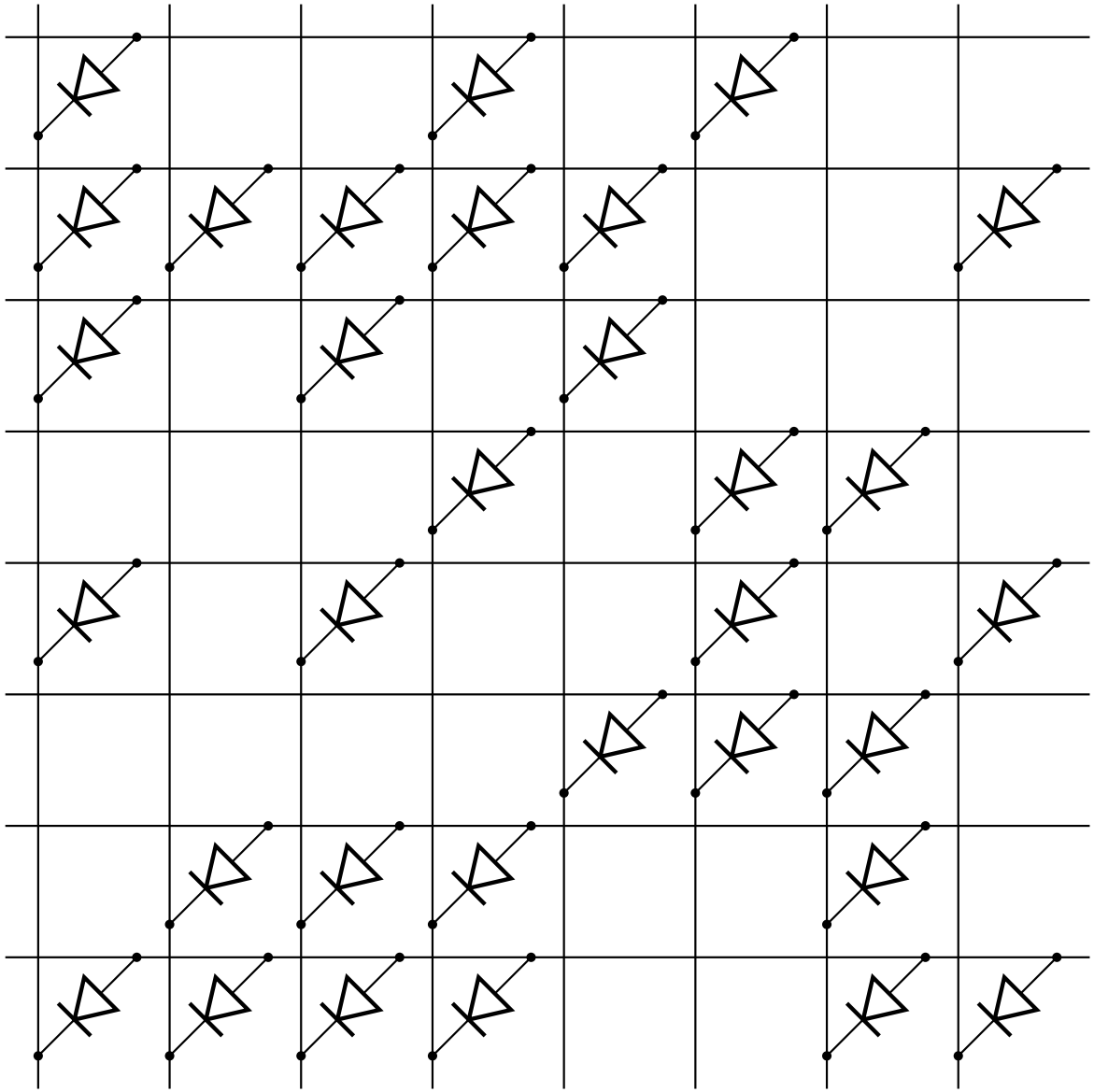
*a.k.a. *von Neumann* as opposed to *neuromorphic*

Conventional* Computer Architecture

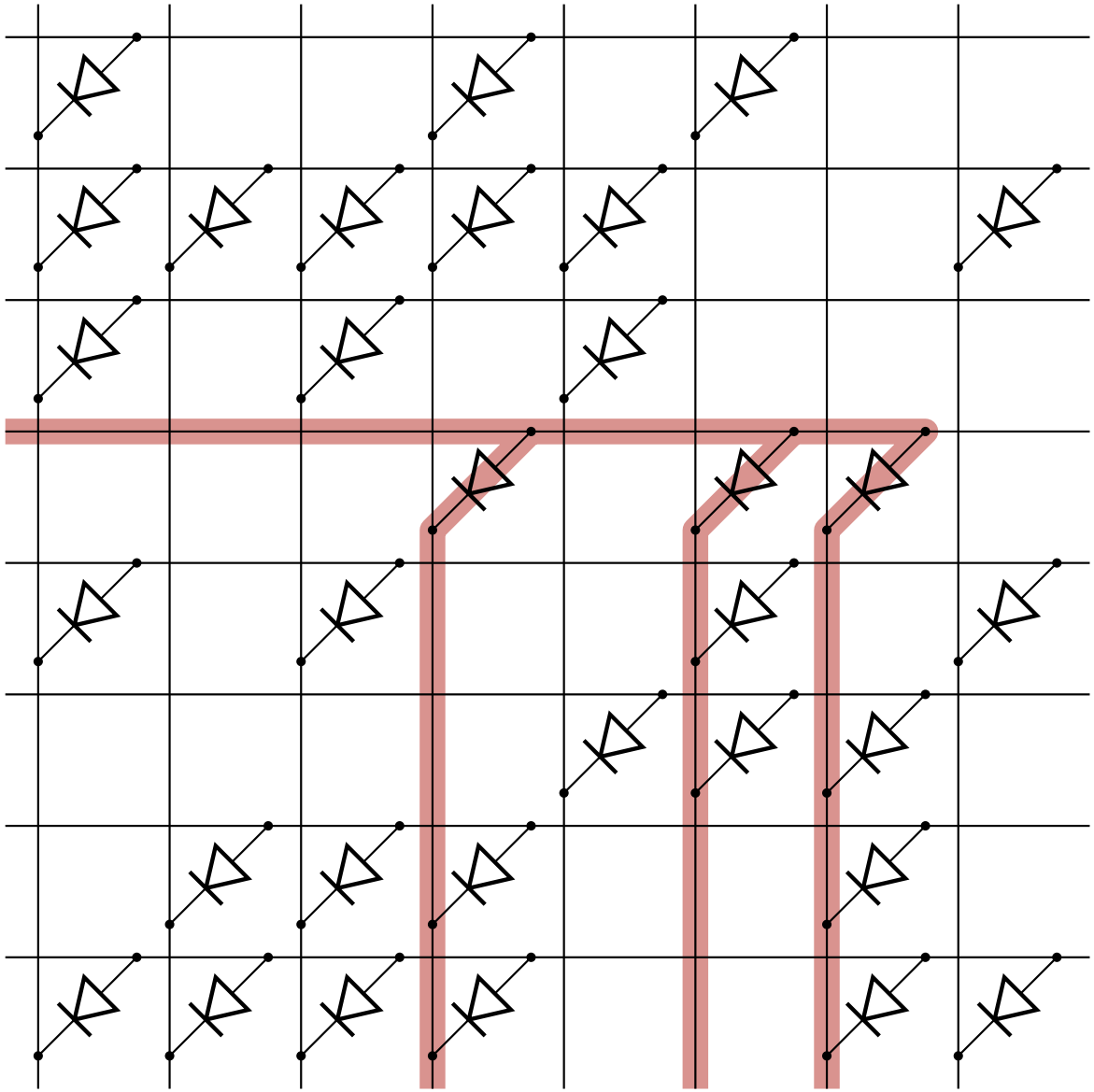


*a.k.a. *von Neumann* as opposed to *neuromorphic*

Close-Up on Read-Only Memory (ROM)



Close-Up on Read-Only Memory (ROM)

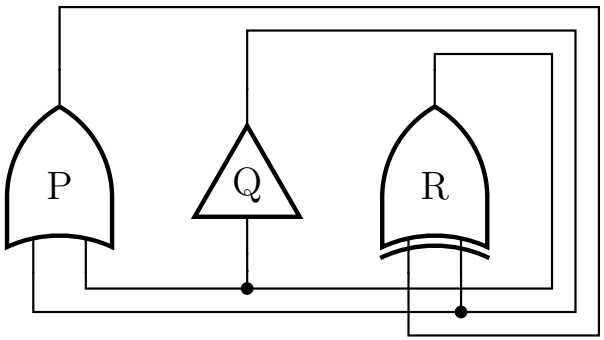


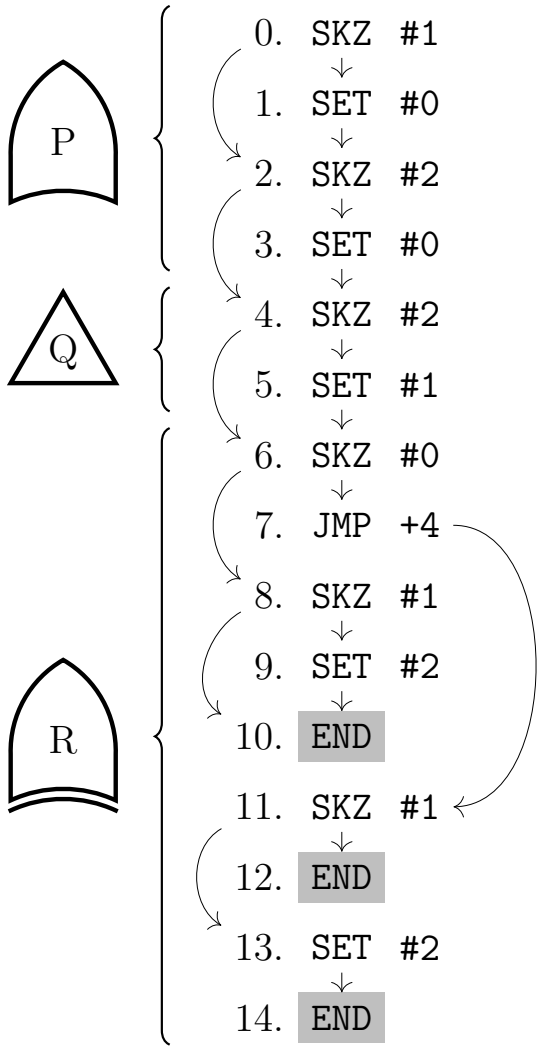
A Program Impacts the Causal Structure!

Premise

- A computer may seem general-purpose
- And a program may seem abstract
- But a program *running* on a computer is a physical arrangement of matter
- $\therefore$ We have to account for the program in a computer's causal structure *somehow*

Multiplexing of Black Boxes Based on Program*





*All code available at https://github.com/jsampson/iit-thesis

Clocks and Flip-Flops

Insight #1

- The clock must be oscillating for a computer to work
- Micro-oscillations are inherent to all physical systems
- $\therefore$ Treat clock *rate* as a background condition

Clocks and Flip-Flops

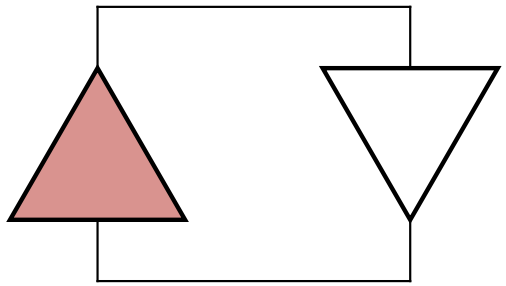
Insight #1

- The clock must be oscillating for a computer to work
- Micro-oscillations are inherent to all physical systems
- $\therefore$ Treat clock *rate* as a background condition

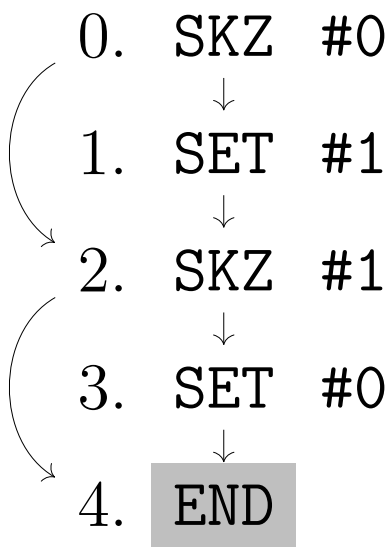
Insight #2

- Clock ticks trigger flip-flop state transitions
- All other logic gates are subordinate to the flip-flops and can be randomized after each clock tick
- $\therefore$ The stateful elements that matter are flip-flops, with the clock period setting the time scale

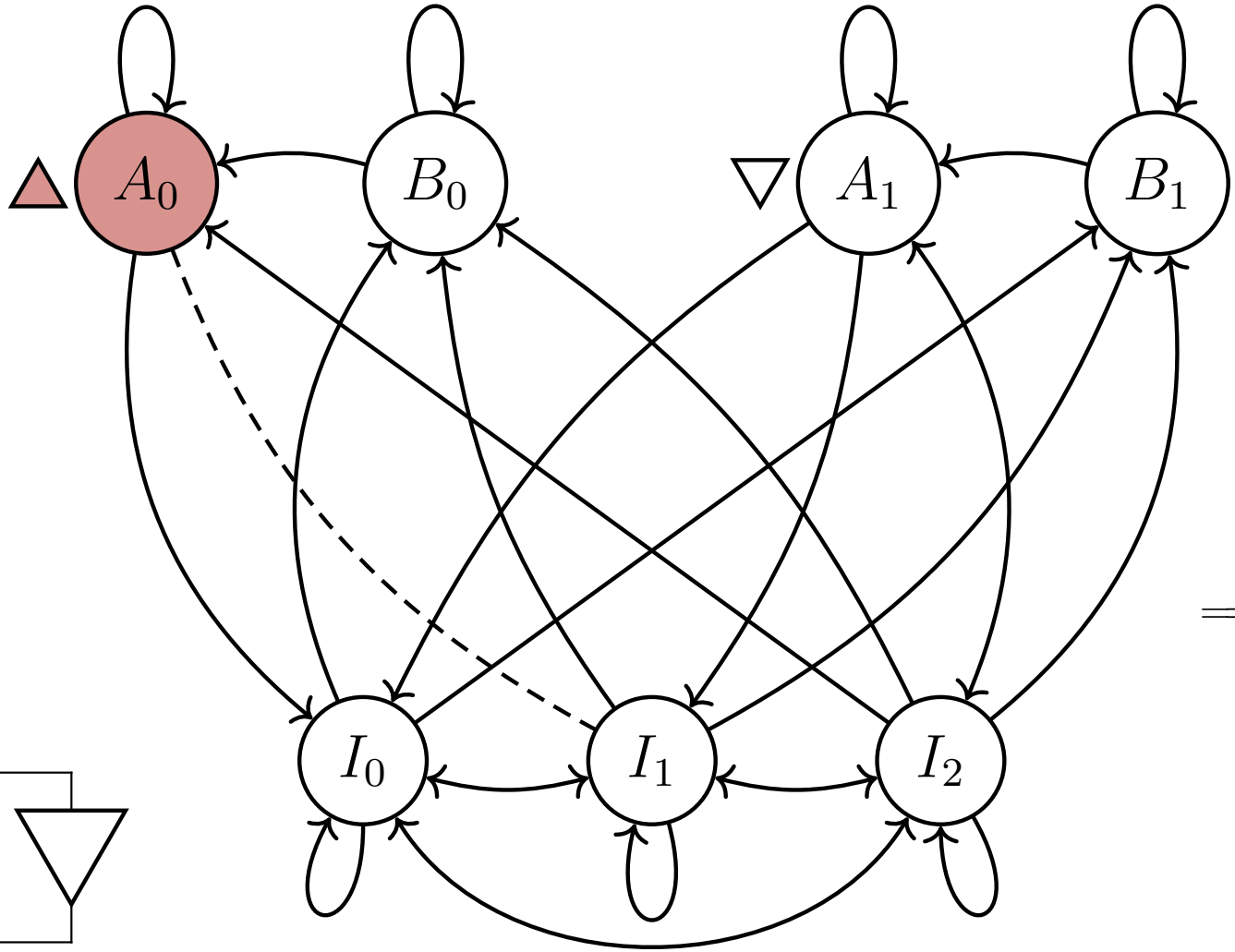
A Trivial Motivating Example

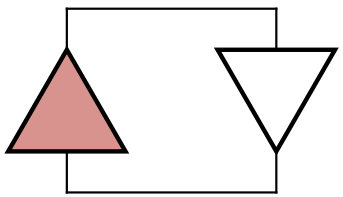


$$\Phi = 1.0$$



A Causal Model Based on Flip-Flops

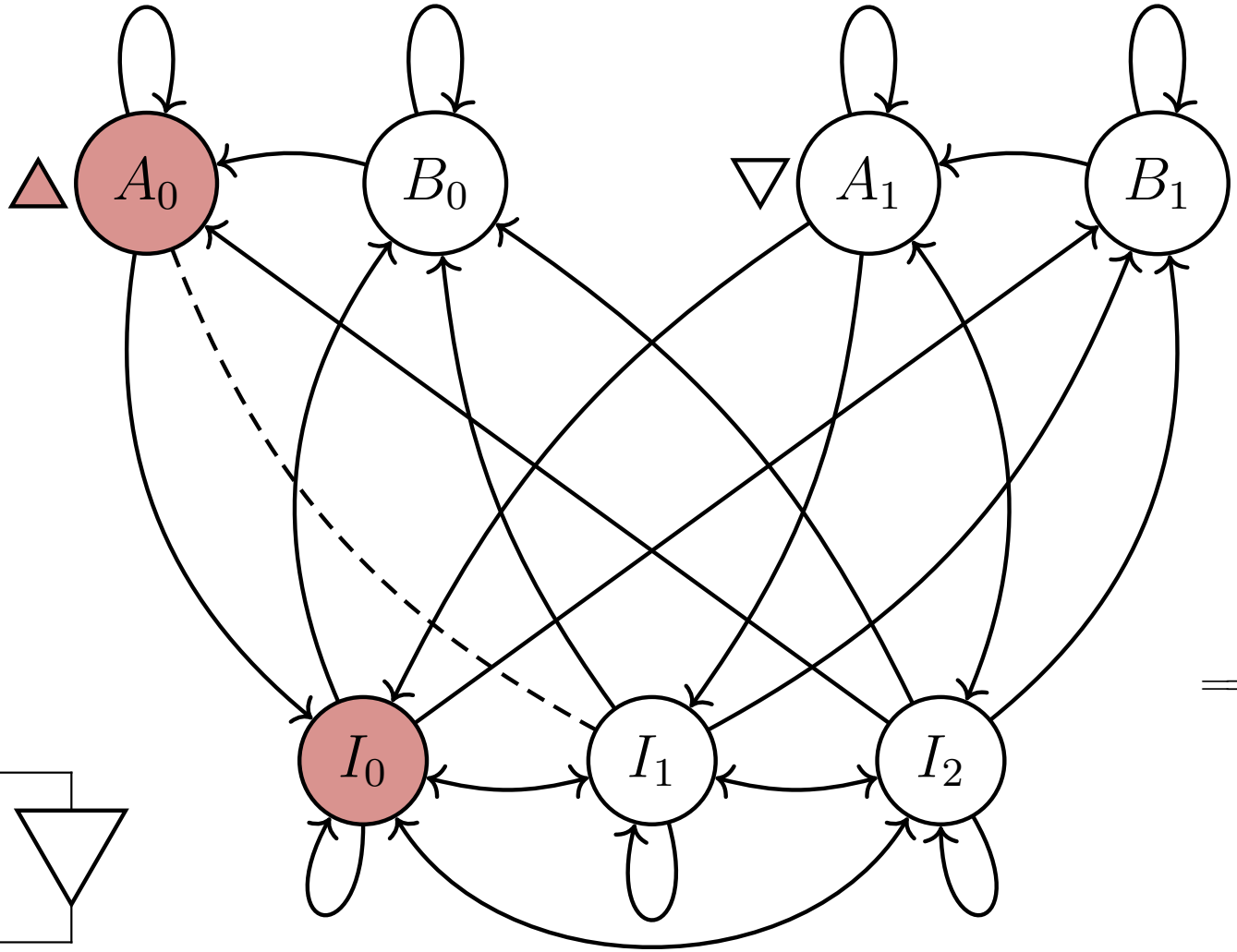


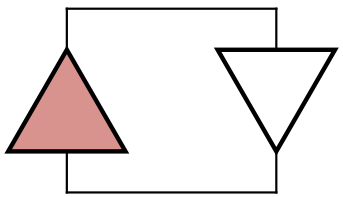


$\Longrightarrow$

```
0.  SKZ  #0
1.  SET  #1
2.  SKZ  #1
3.  SET  #0
4.  END
```

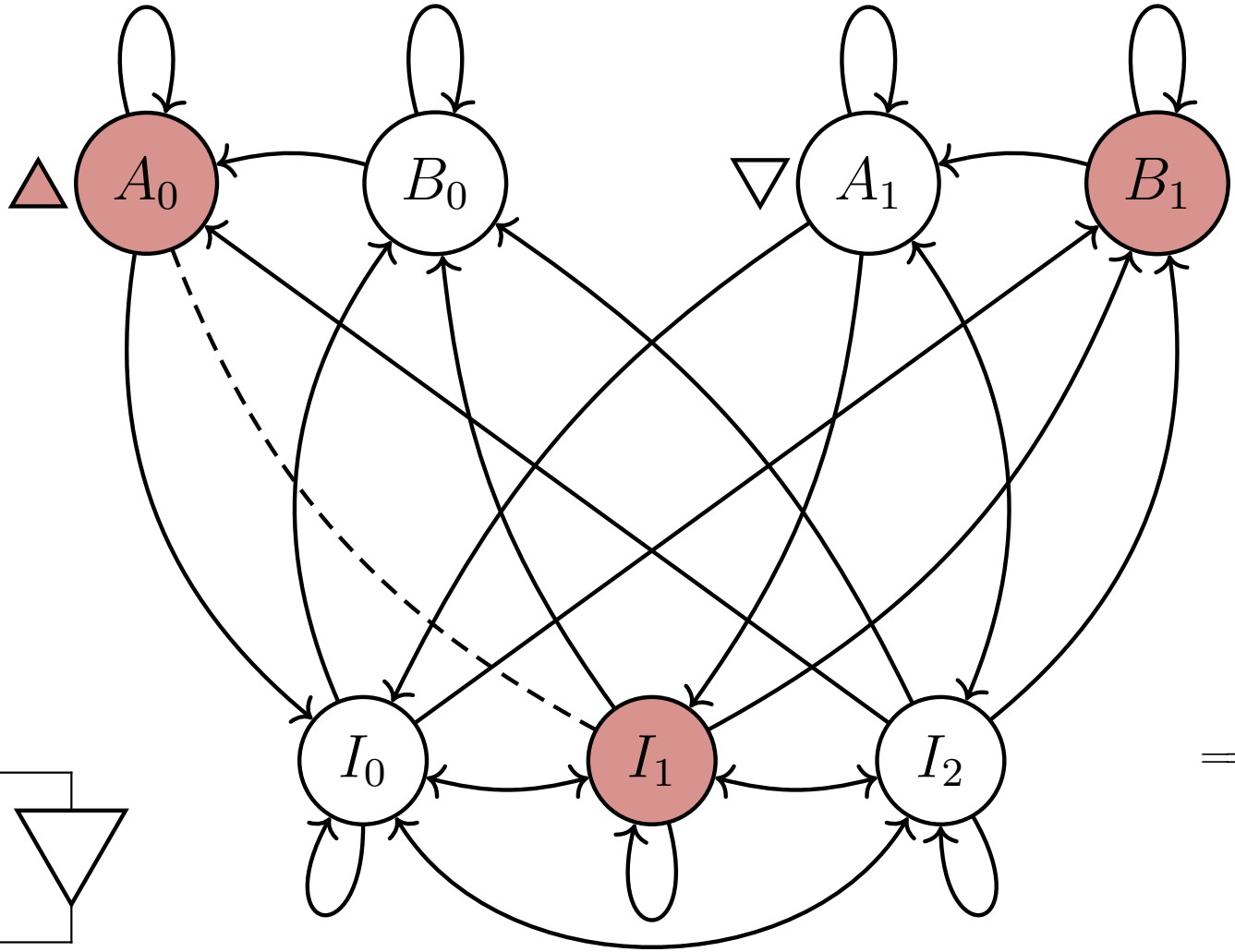
A Causal Model Based on Flip-Flops

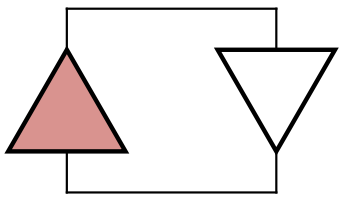




```
0. SKZ  #0
1. SET  #1
2. SKZ  #1
3. SET  #0
4. END
```

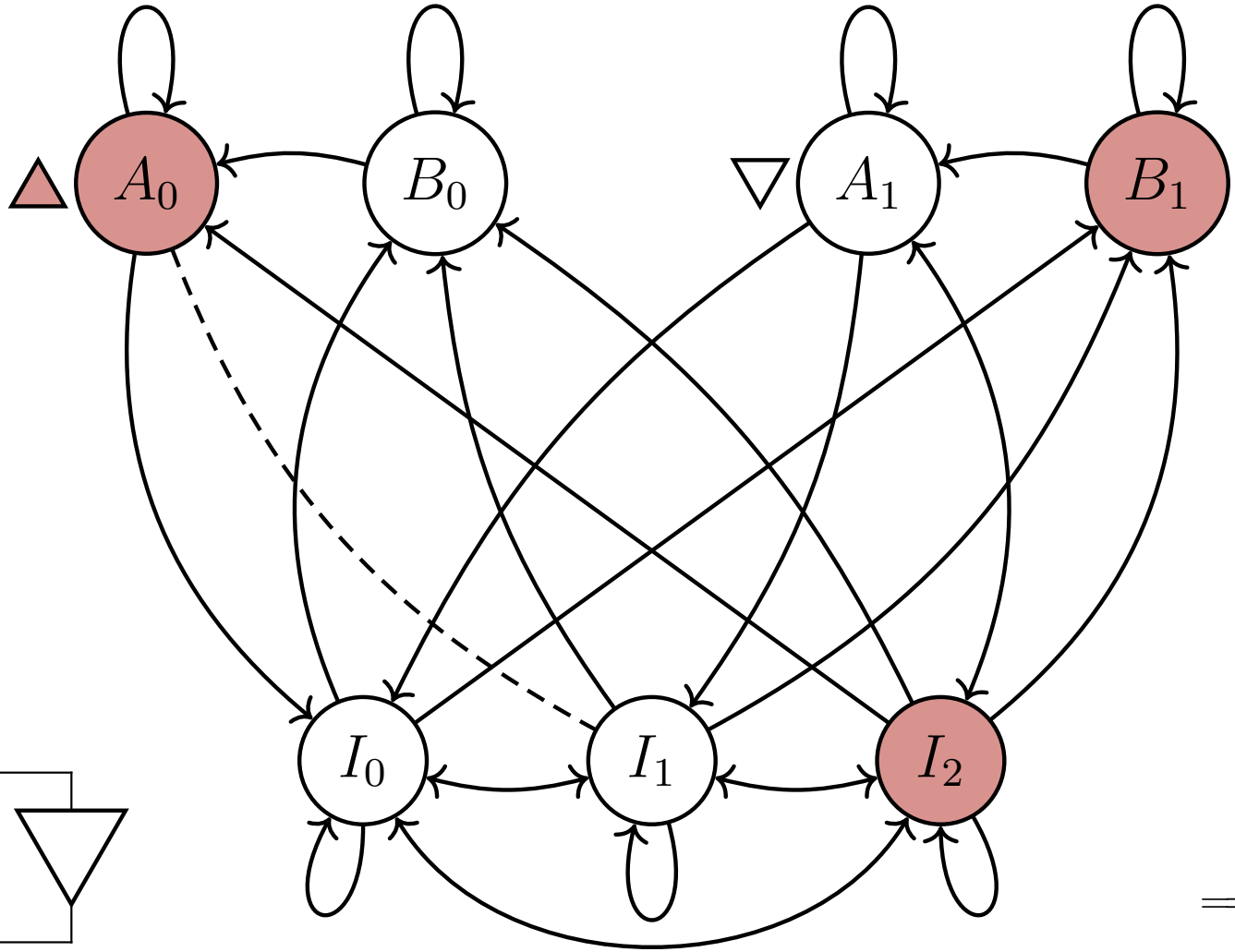
A Causal Model Based on Flip-Flops

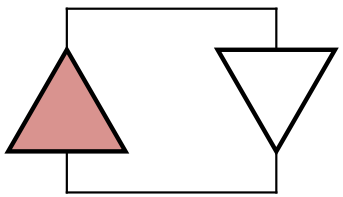




```
0. SKZ  #0
1. SET  #1
⟹ 2. SKZ  #1
3. SET  #0
4. END
```

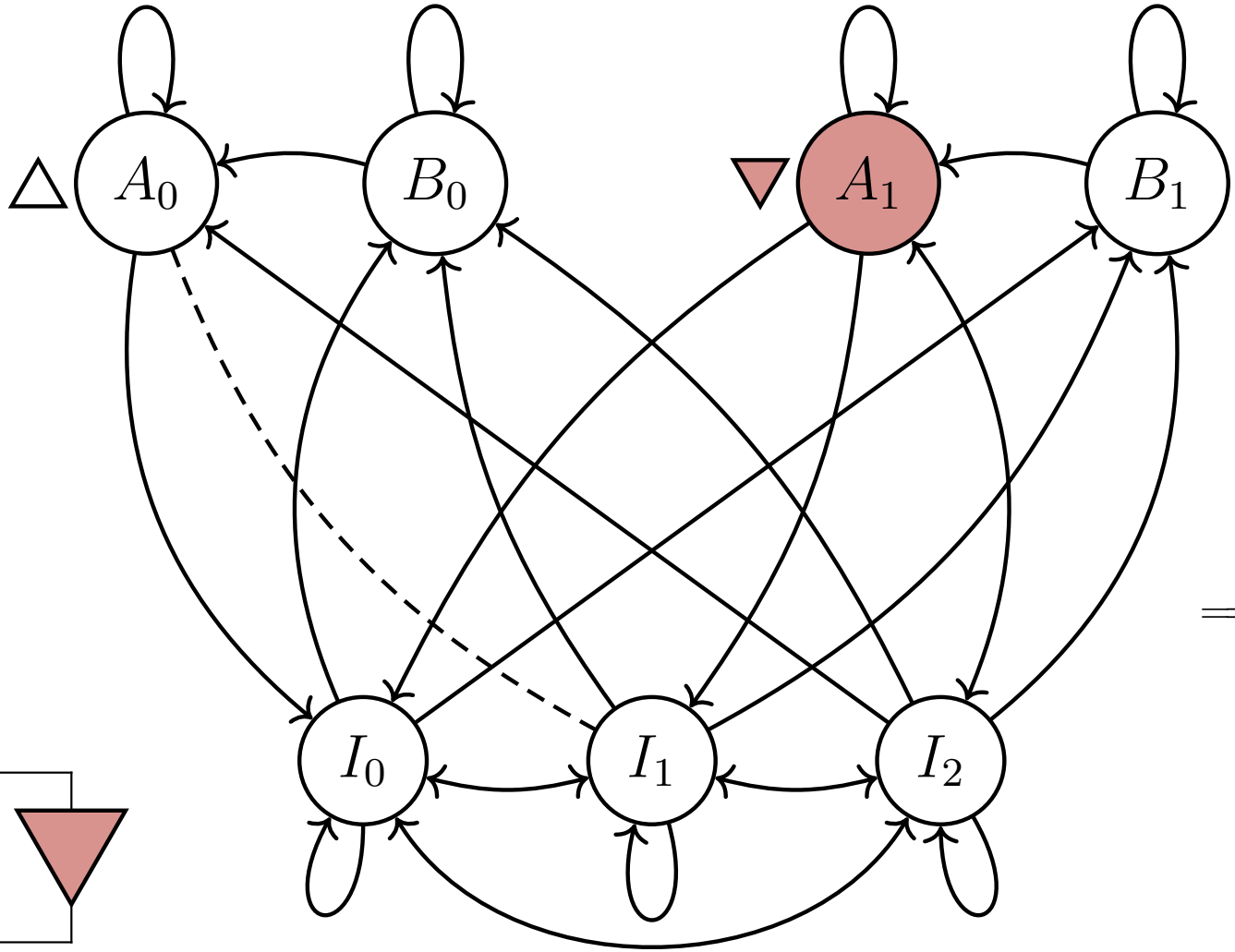
A Causal Model Based on Flip-Flops

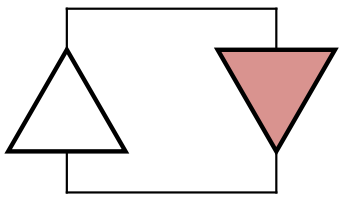




```
0.  SKZ  #0
1.  SET  #1
2.  SKZ  #1
3.  SET  #0
⟹ 4.  END
```

A Causal Model Based on Flip-Flops

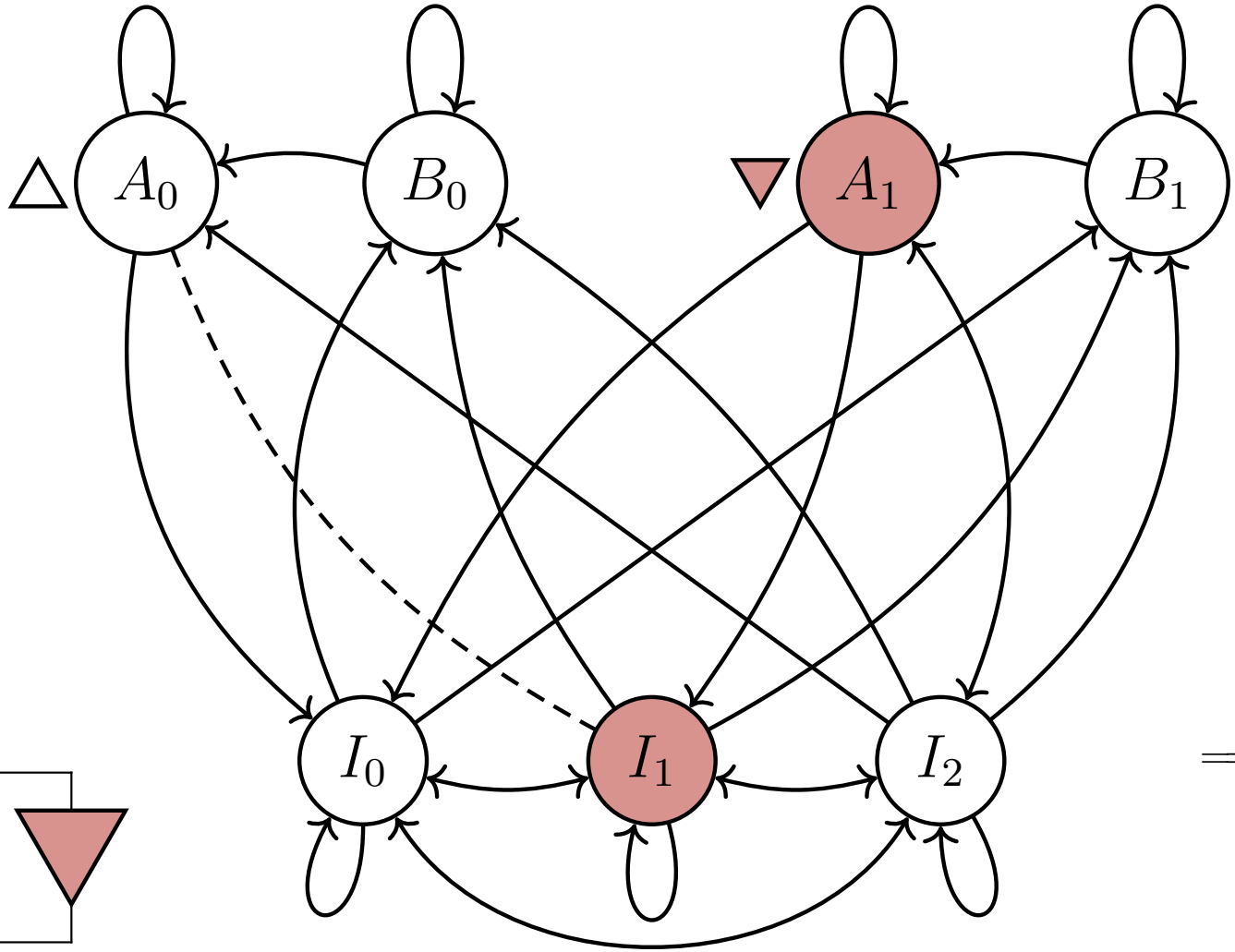


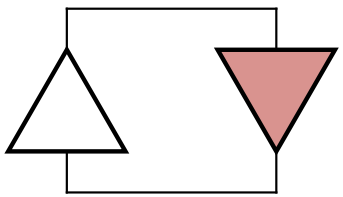


$\Longrightarrow$

```
0.  SKZ  #0
1.  SET  #1
2.  SKZ  #1
3.  SET  #0
4.  END
```

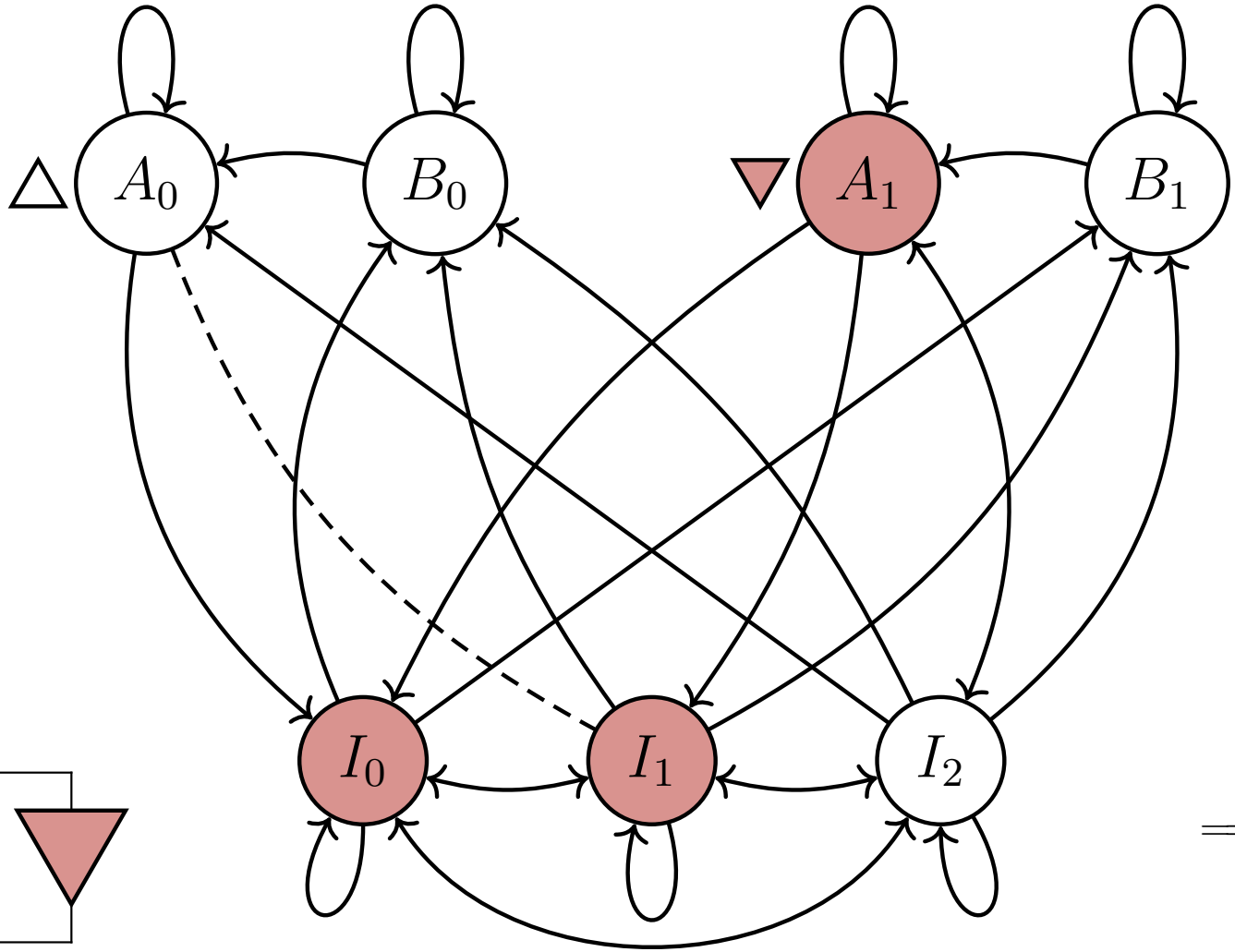
A Causal Model Based on Flip-Flops

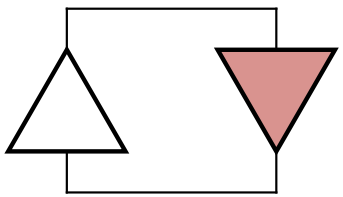




```
0.  SKZ  #0
1.  SET  #1
⟹ 2.  SKZ  #1
3.  SET  #0
4.  END
```

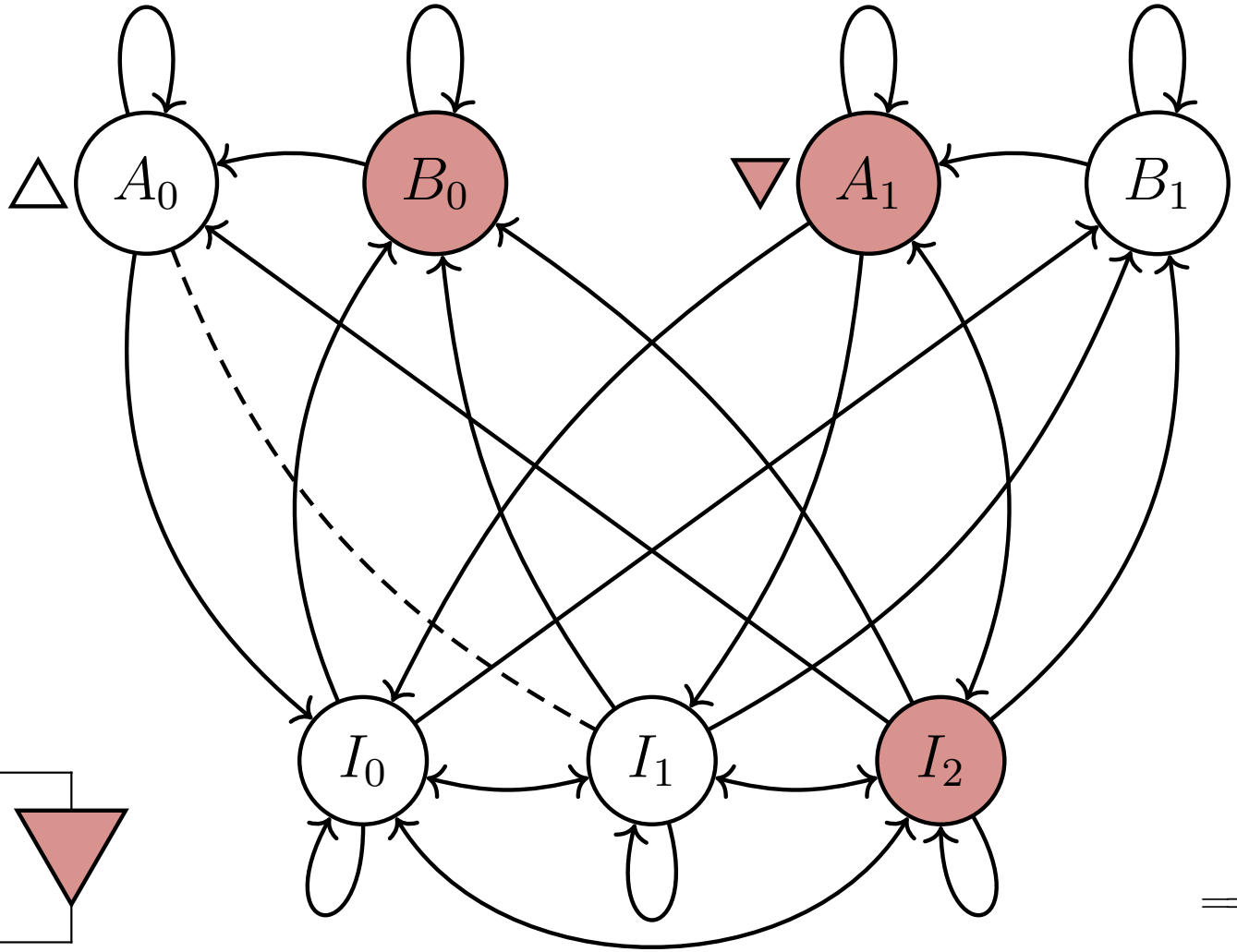
A Causal Model Based on Flip-Flops

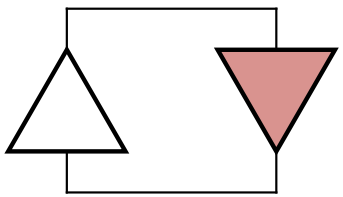




```
0. SKZ  #0
1. SET  #1
2. SKZ  #1
⟹ 3. SET  #0
4. END
```

A Causal Model Based on Flip-Flops

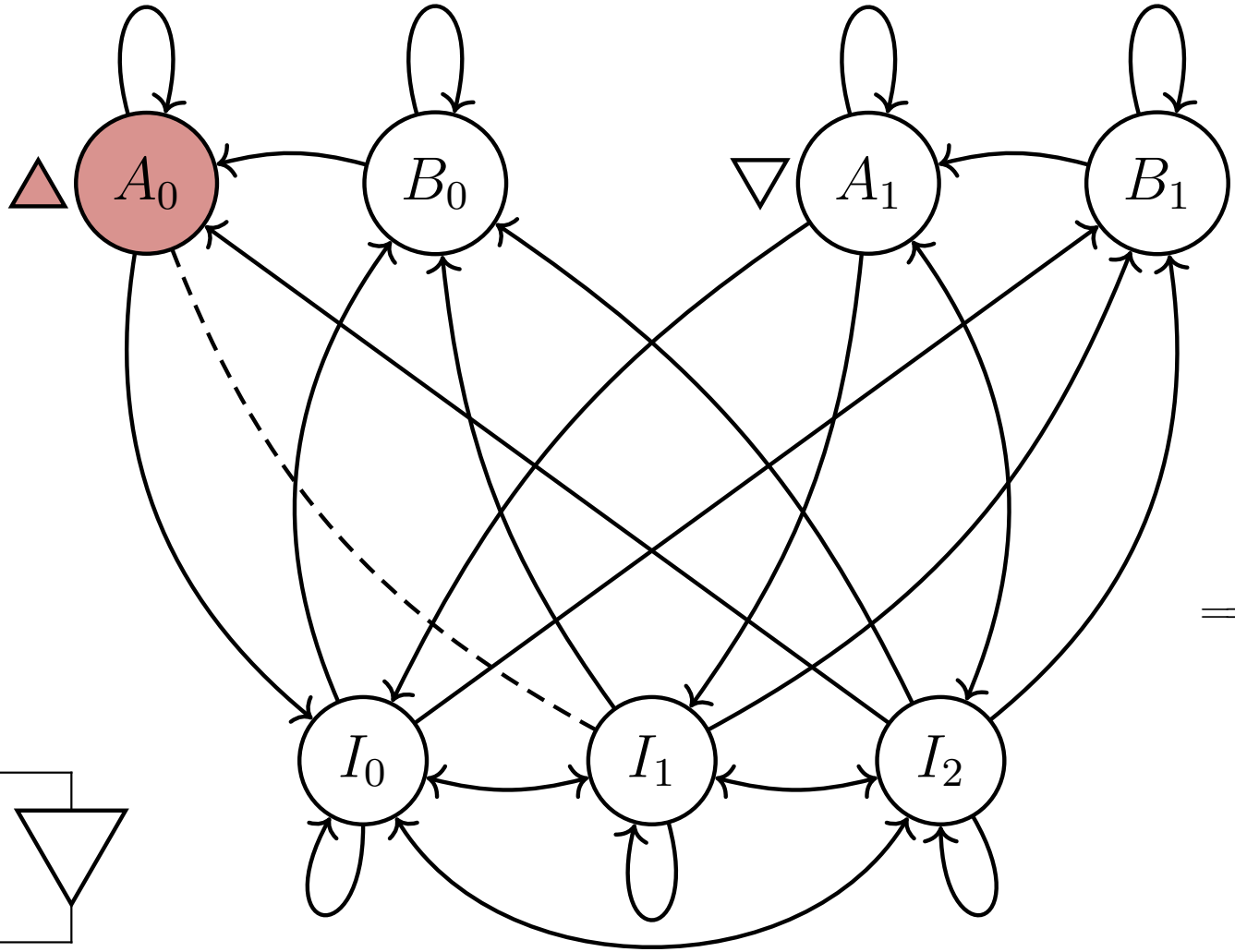


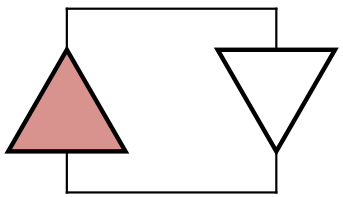


0. `SKZ #0`
1. `SET #1`
2. `SKZ #1`
3. `SET #0`

$\Longrightarrow$ 4. `END`

A Causal Model Based on Flip-Flops





$\Longrightarrow$

```
0.  SKZ  #0
1.  SET  #1
2.  SKZ  #1
3.  SET  #0
4.  END
```

A Computer Returns to Well-Formed States

Insight #3

- Elements of the simulated system correspond directly to a subset of elements in the computer
- In a *well-formed state*, states of all *other* elements of the computer are constrained *independently* of simulated system state
- Any well-formed state eventually leads to another well-formed state for the corresponding next simulated system state
- $\therefore$ Well-formed states prevent causally-significant state from "hiding" inside the computer

Takeaways

- A program *does* impact a computer's causal structure
- A computer's circuits are *multiplexed* among multiple black boxes according to the program's instructions
- A computer's causal structure reveals these insights:
  1. The clock's *rate* is a background condition
  2. The stateful elements that matter are *flip-flops*
  3. The computer returns to well-formed states such that causally-relevant state is not hidden
- ∴ As long as a program's data has the same structure as the system being simulated, the computer running it should be causally (and phenomenally) equivalent

Justin T. Sampson
sampsonetics@acm.org
https://www.linkedin.com/in/sampsonetics