Engineering of Chemical Register Machines

Prague International Workshop on Membrane Computing 2008

R. Fassler, T. Hinze, T. Lenser and P. Dittrich
{raf,hinze,thlenser,dittrich}@minet.uni-jena.de

2. June 2008

Outline

Motivation

Goal:

- Chemical system with Turing completeness
- Use mass action kinetics only
- Binary chemical information encoding
- No assumptions like molecule structure or a hierarchy of membranes

Motivation

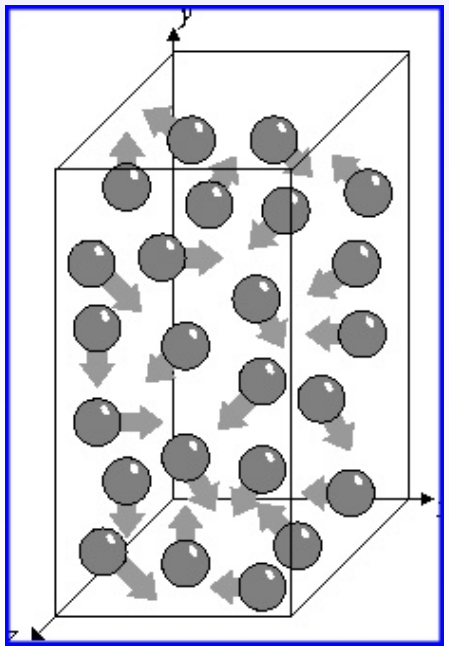
Realization:

- Artificial chemistry as P-System
- Registers composed of flip flops store information
- Chemical clock orders operations on registers
- P-System expands registers if necessary

P-Systems as Artificial Chemistries

Similarities:

- Many autonomous units like molecules or agents
- Decentralization
- Nondeterminism
- Stochasticity
- Data parallelism



Register Machine

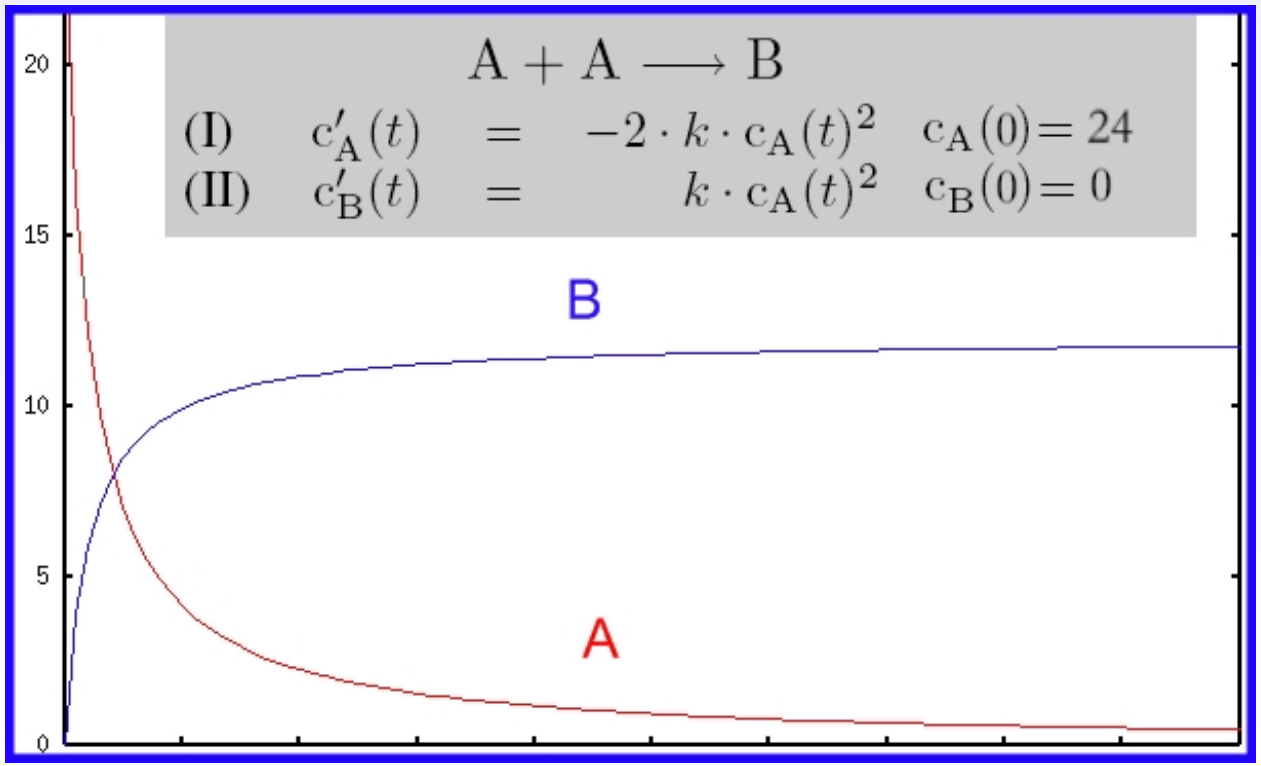
Definition: $M = (R, L, P, \#_0)$ with

- Registers $R = \{\mathrm{R}_1, \ldots, \mathrm{R}_m\}$
- Labels (addresses) $L = \{\#_0, \ldots, \#_n\}$
- Instructions P
- Initial label $\#_0$
- Registers store binary values for natural numbers

Mass Action Kinetics

Definition:

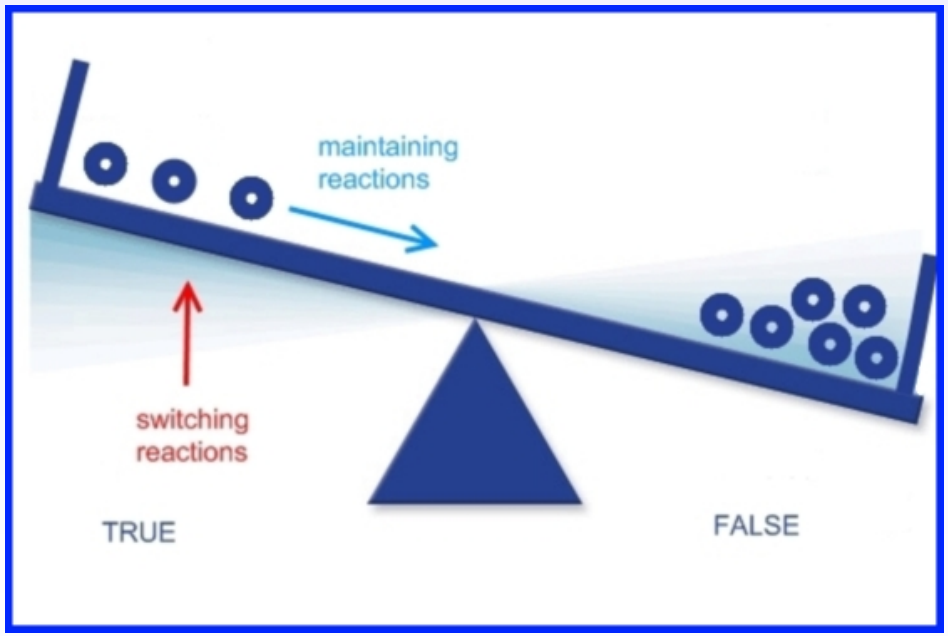
- Speed of an reaction depends on reactants concentrations and a constant factor
- No saturation
- No inhibitors



Chemical Implementation of Logical Variables

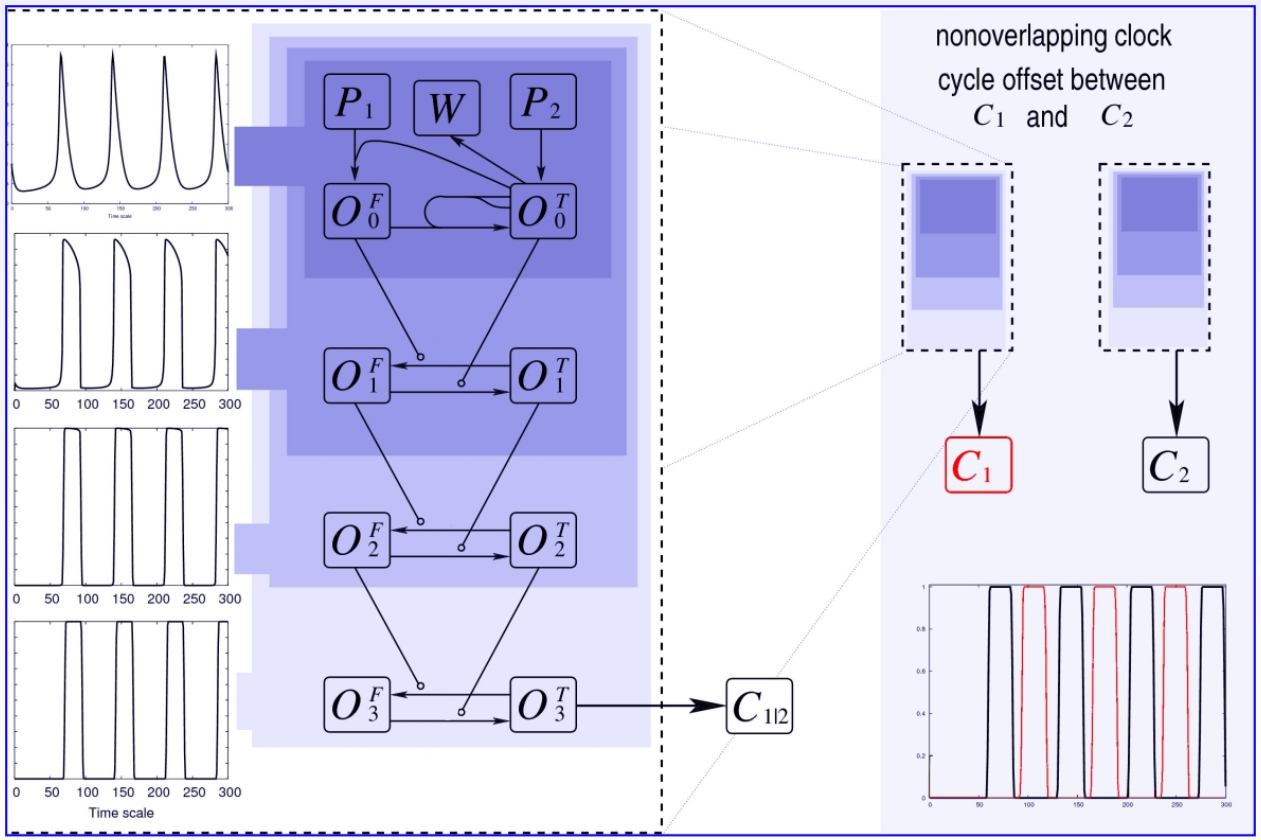
Variables:

- Two species representing "true" and "false"
- Correlated concentrations
- Reactions to switch between "true" and "false"
- Reactions to maintain consistency



A Chemical Clock

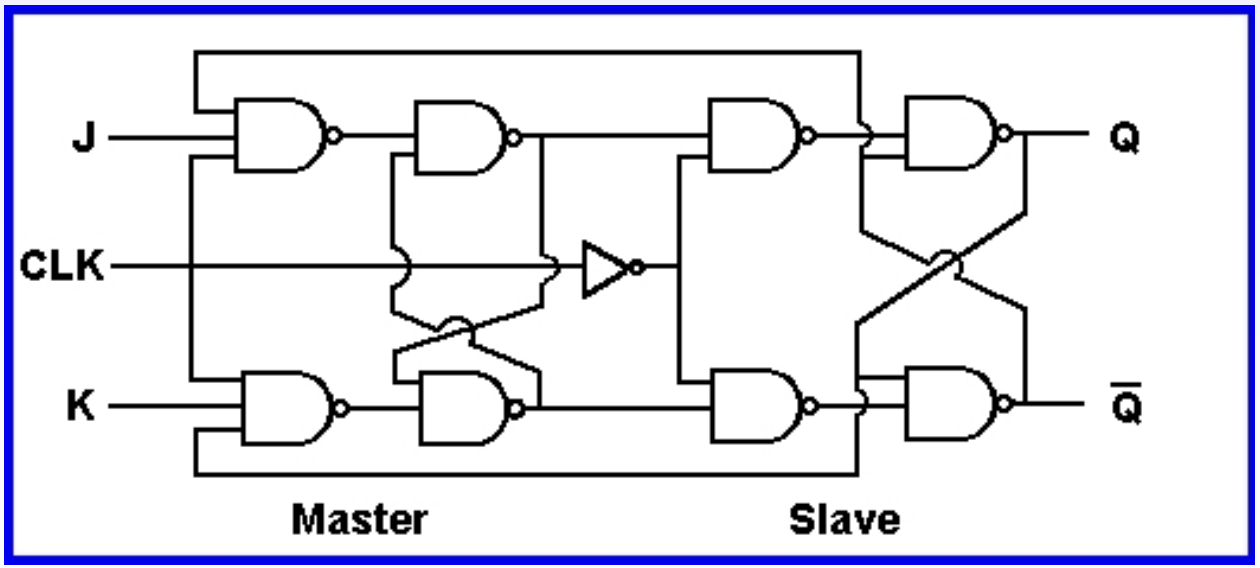
- Based on Belousov- Zhabotinsky reactions
- Cascade of variables derives high/low signals
- Two offset oscillators form the clock signal



Master Slave - Flipflop (MS-FF)

MS-FF: - technical role model

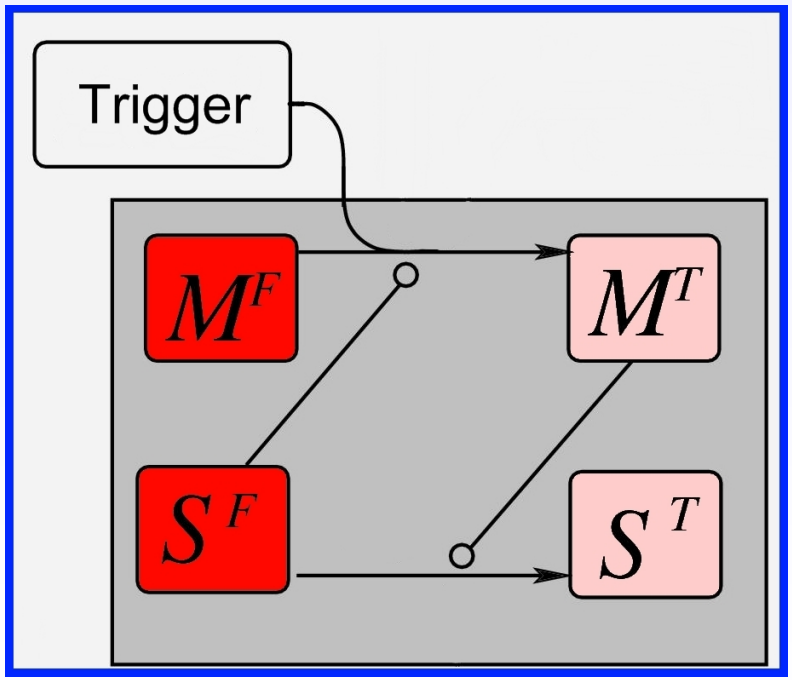
- Two-part device storing one bit
- Clocked to switch with offset
- Short switches at intended time
- Only one switch per positive edge



Master Slave - Flipflop

MS-FF: - chemical adaption

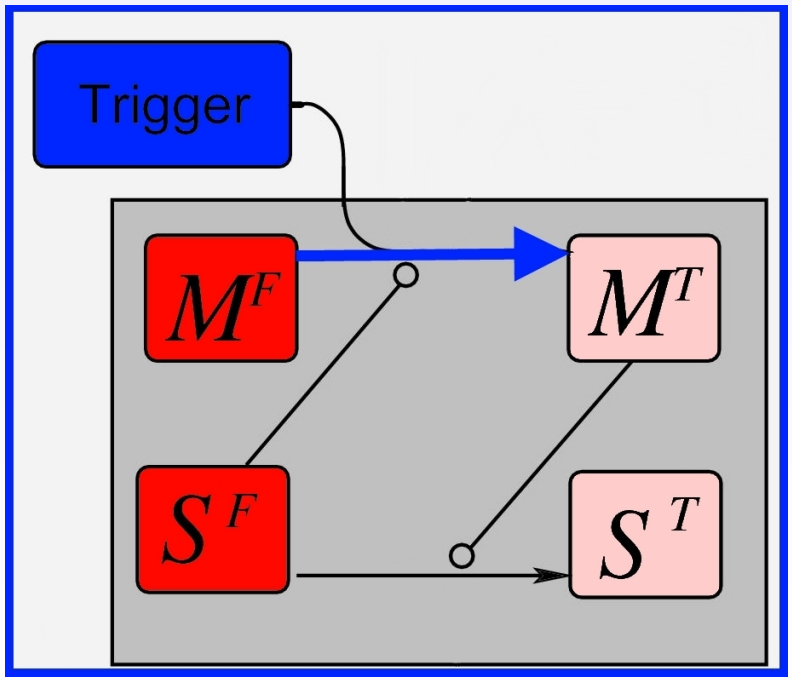
- Master part switches if triggered
- Slave part will adapt master part
- Encoded by two variables
- M/S-reactions separated by two clock signals



Master Slave - Flipflop

MS-FF: - chemical adaption

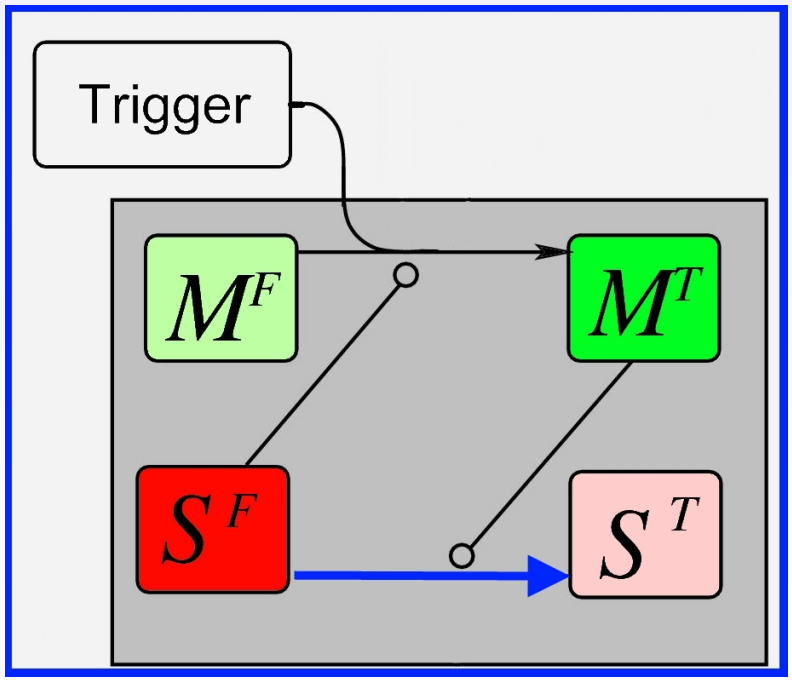
- Master part switches if triggered
- Slave part will adapt master part
- Encoded by two variables
- M/S-reactions separated by two clock signals



Master Slave - Flipflop

MS-FF: - chemical adaption

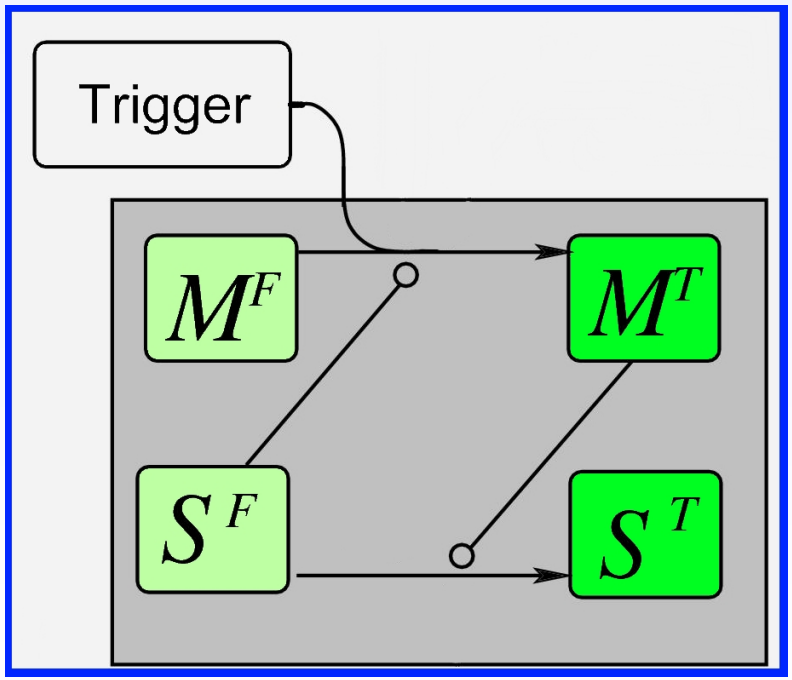
- Master part switches if triggered
- Slave part will adapt master part
- Encoded by two variables
- M/S-reactions separated by two clock signals



Master Slave - Flipflop

MS-FF: - chemical adaption

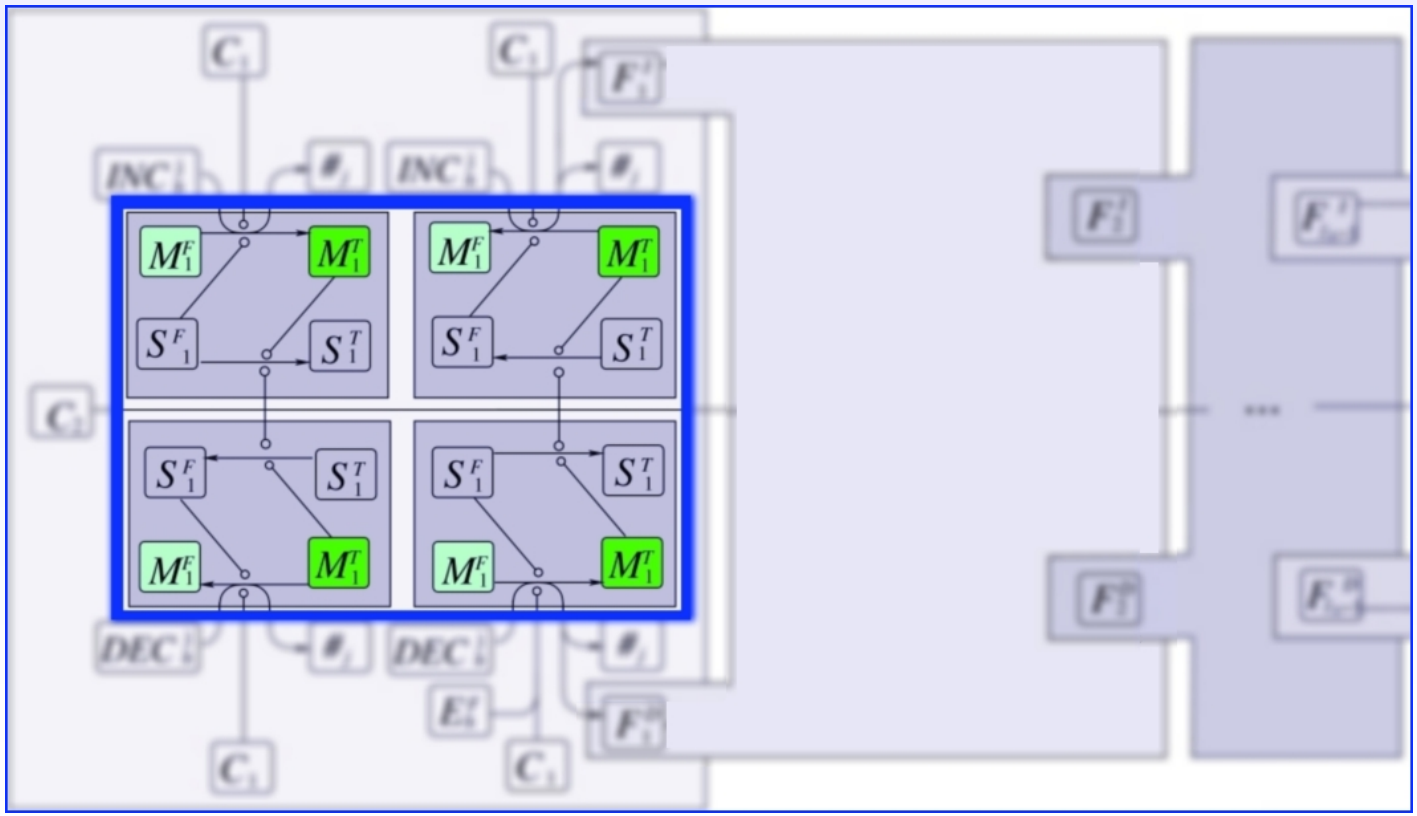
- Master part switches if triggered
- Slave part will adapt master part
- Encoded by two variables
- M/S-reactions separated by two clock signals



Chemical Registers

Registers:

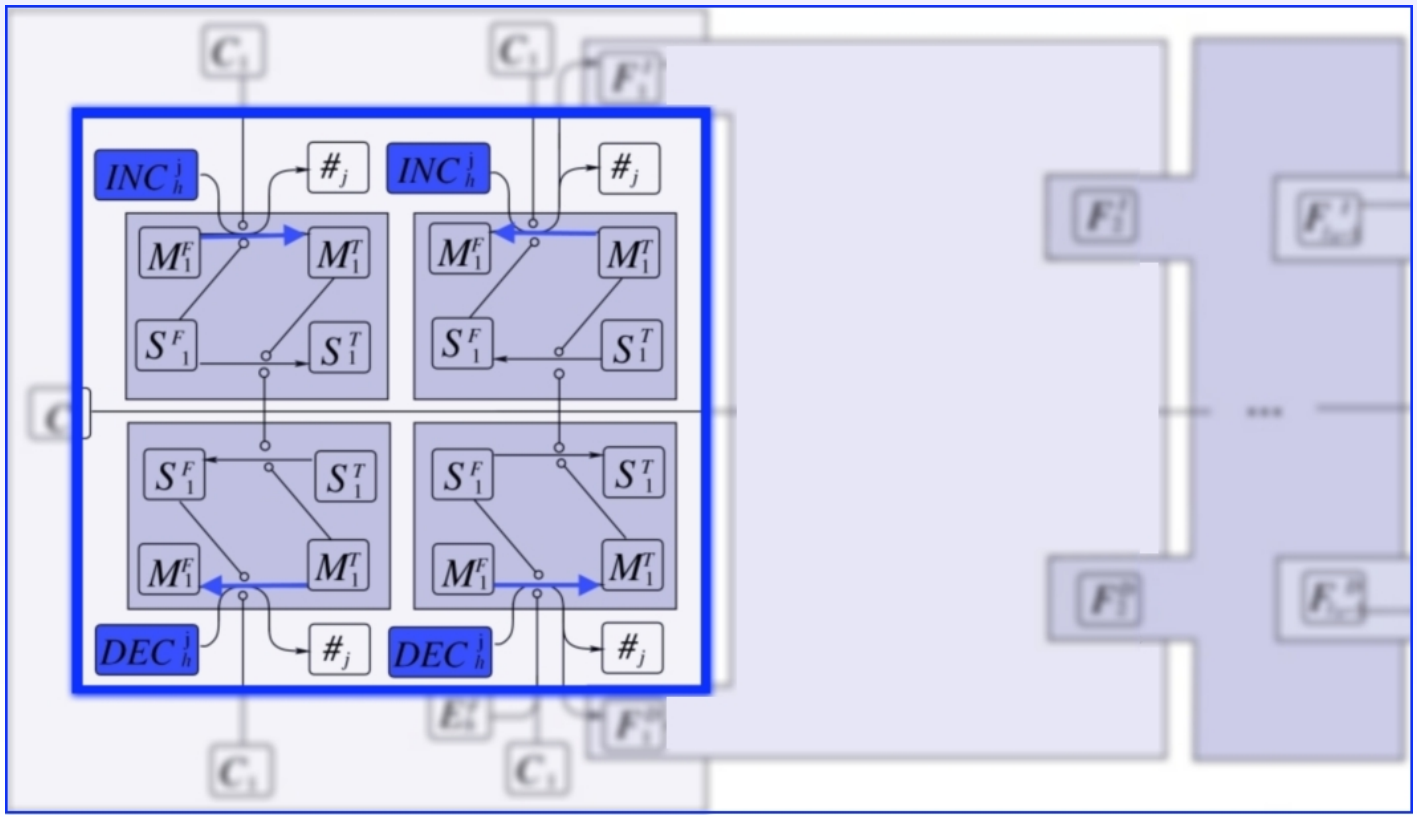
- One MS-FF for each bit, chained
- Extra variable to check if empty
- Only the first bit can be flipped manually
- Add carries will flip further bits



Chemical Registers

Registers:

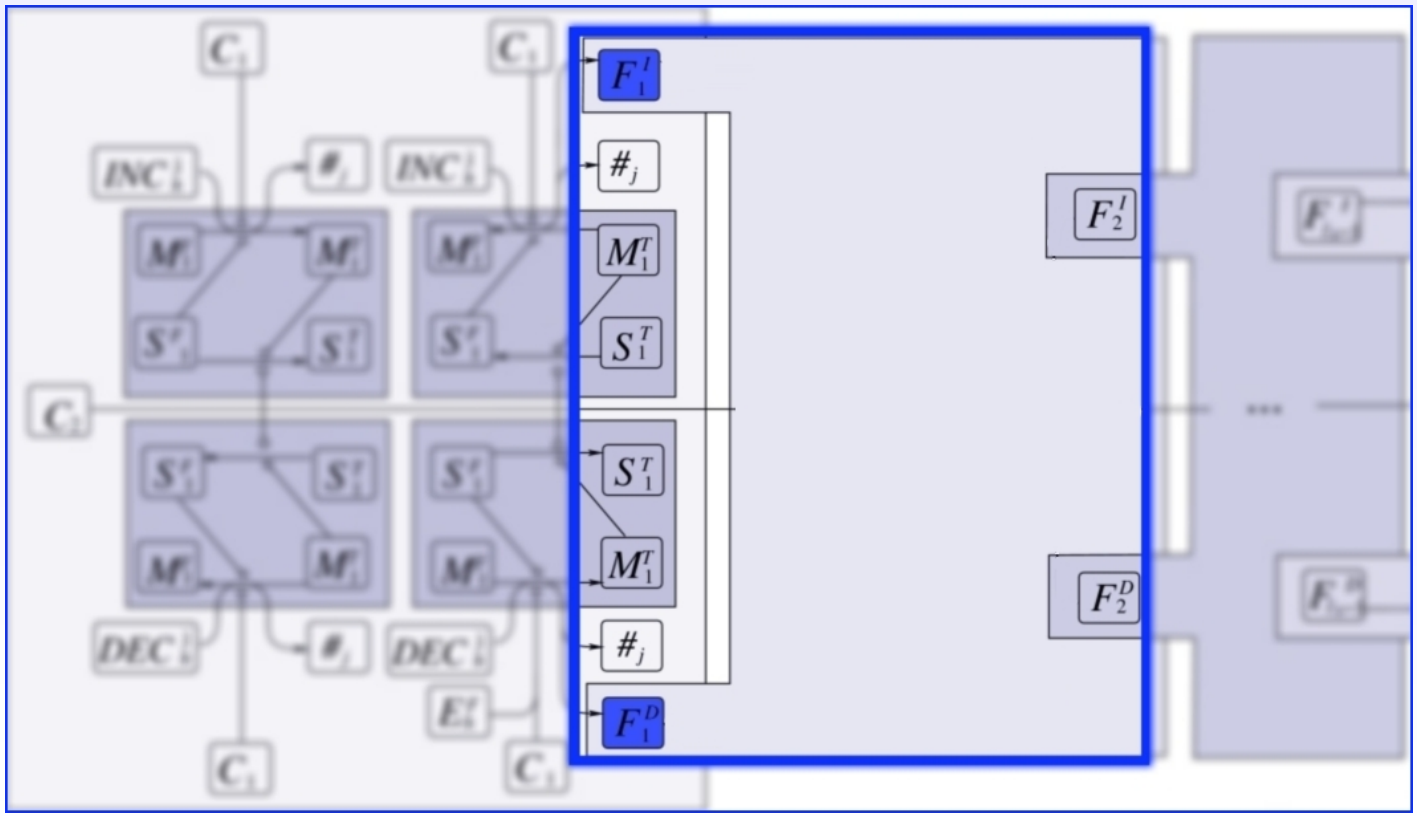
- One MS-FF for each bit, chained
- Extra variable to check if empty
- Only the first bit can be flipped manually
- Add carries will flip further bits



Chemical Registers

Registers:

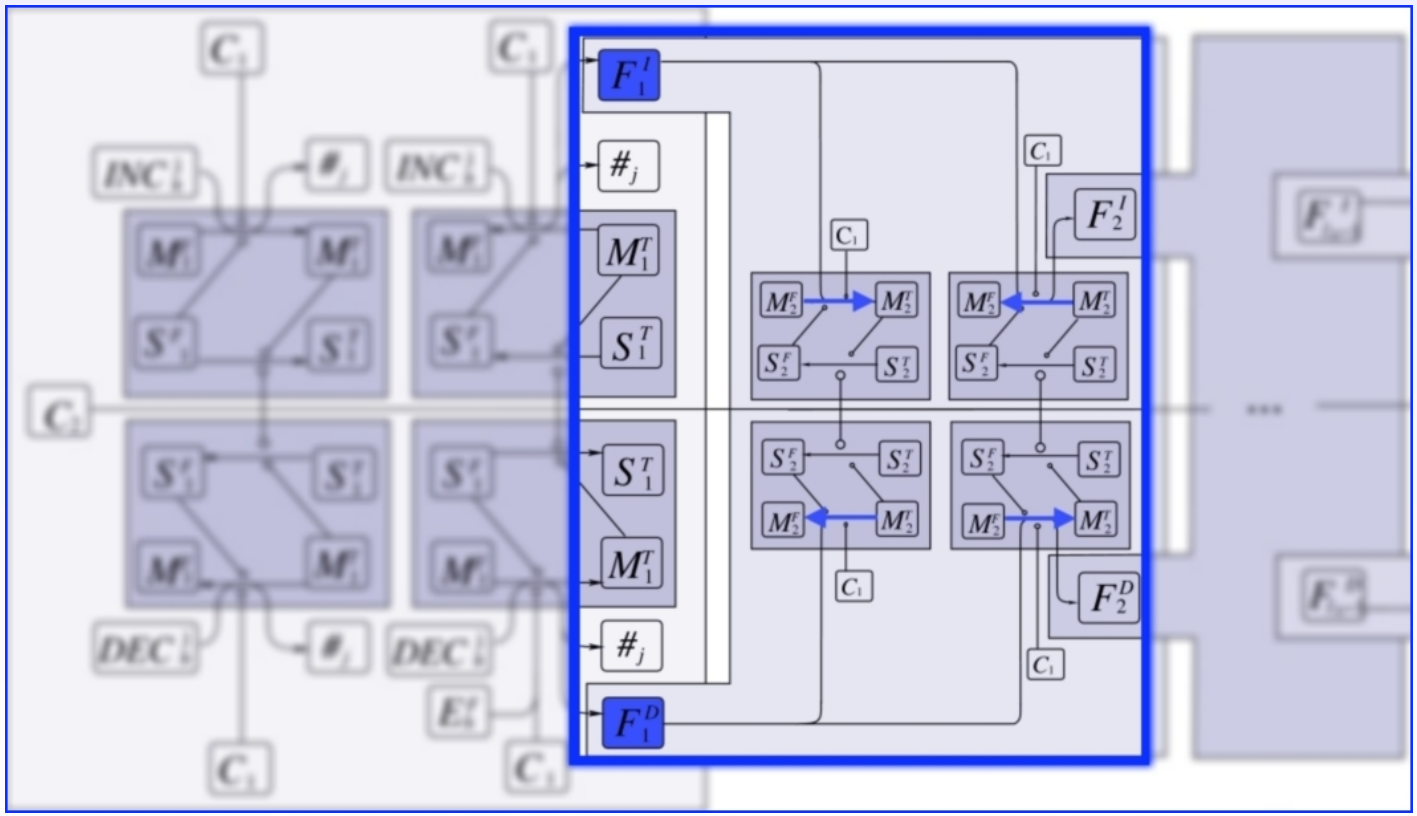
- One MS-FF for each bit, chained
- Extra variable to check if empty
- Only the first bit can be flipped manually
- Add carries will flip further bits



Chemical Registers

Registers:

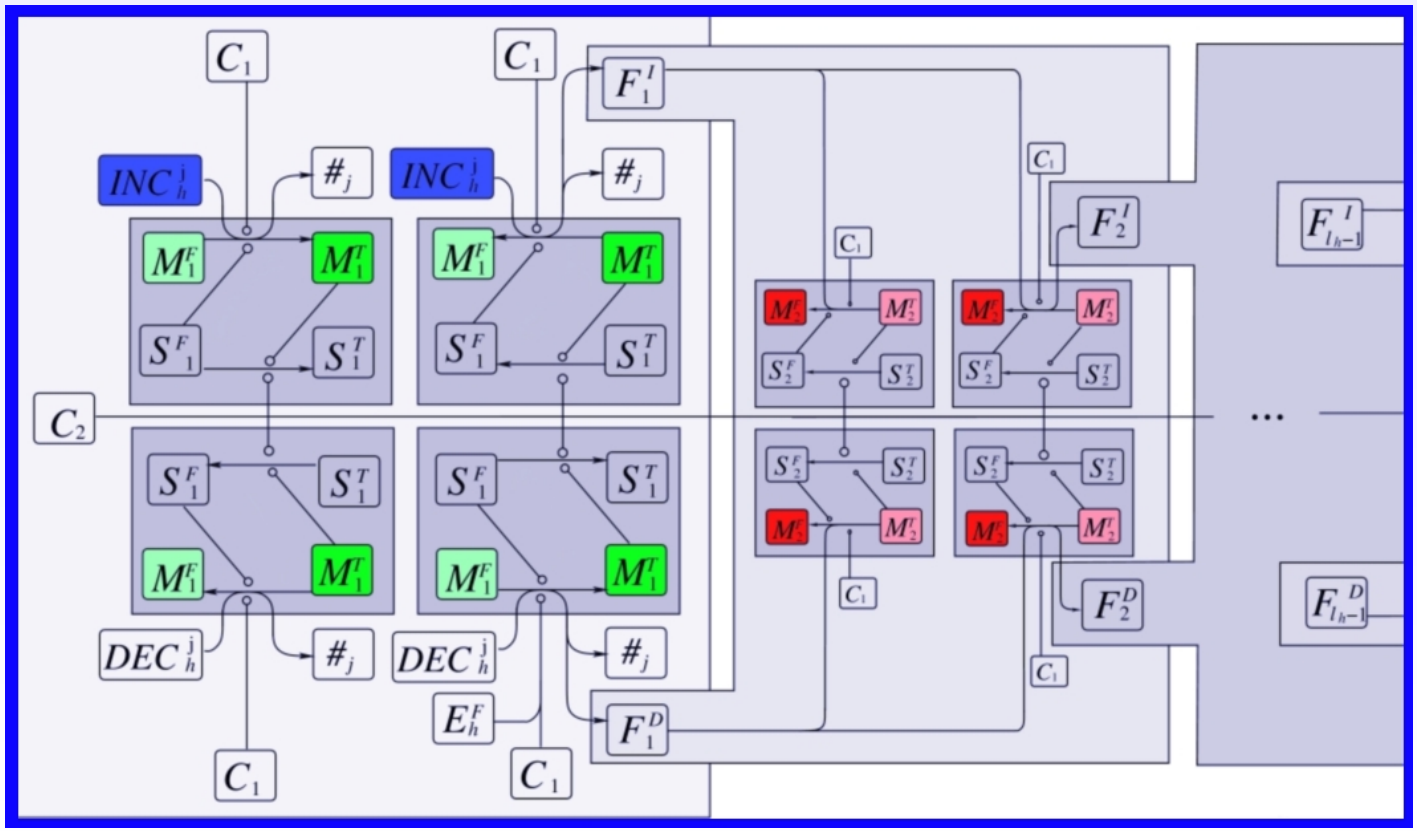
- One MS-FF for each bit, chained
- Extra variable to check if empty
- Only the first bit can be flipped manually
- Add carries will flip further bits



Chemical Registers

Registers:

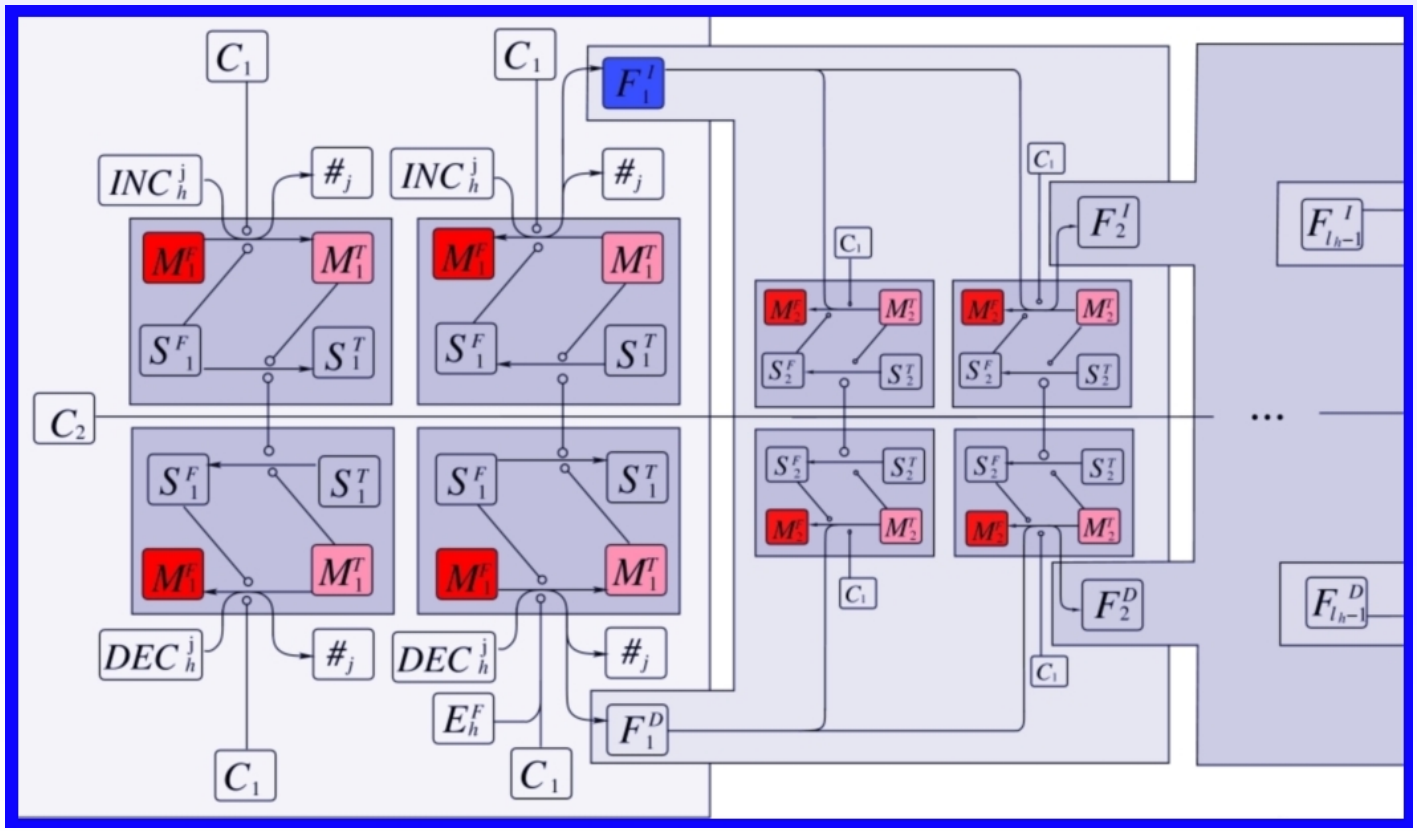
- One MS-FF for each bit, chained
- Extra variable to check if empty
- Only the first bit can be flipped manually
- Add carries will flip further bits



Chemical Registers

Registers:

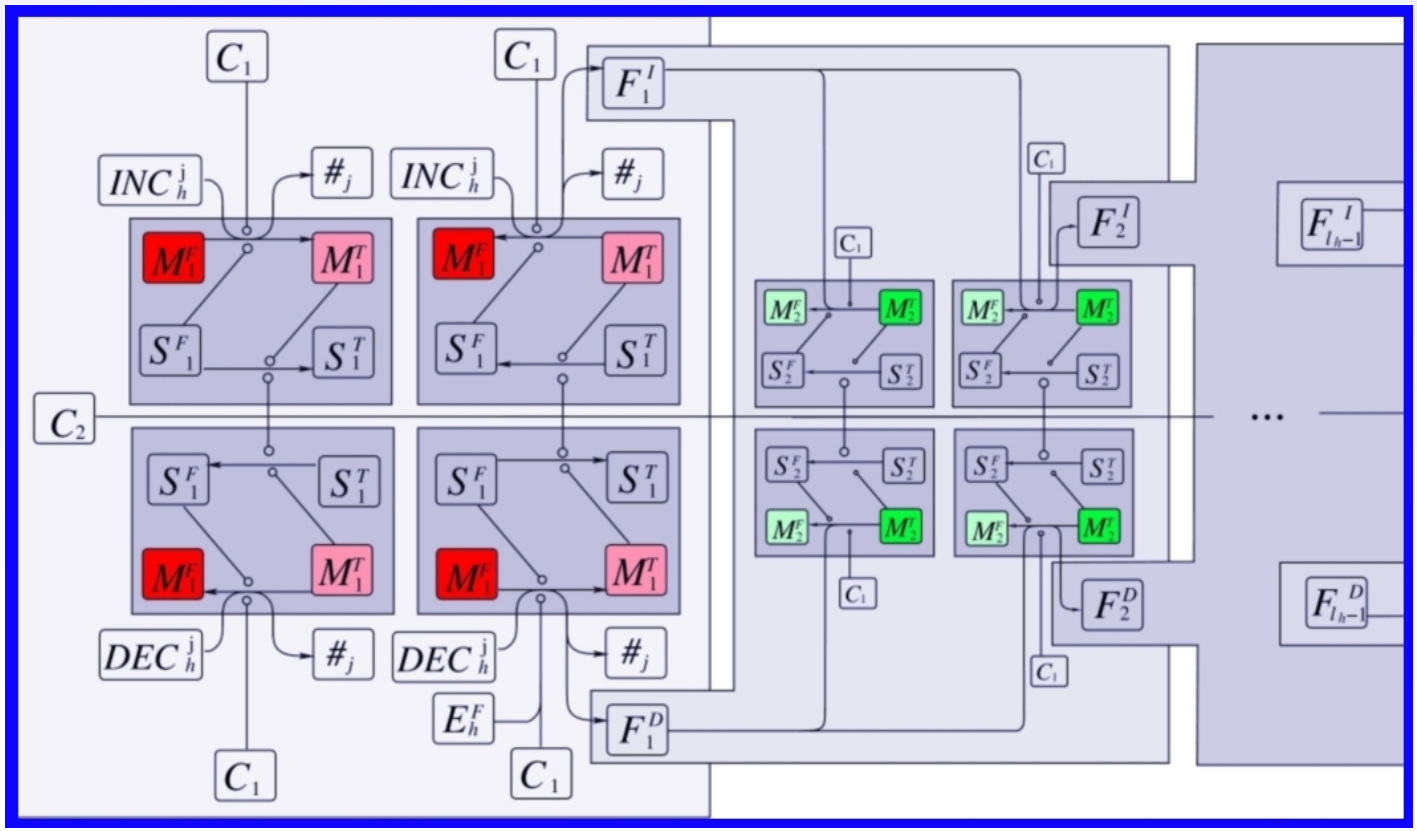
- One MS-FF for each bit, chained
- Extra variable to check if empty
- Only the first bit can be flipped manually
- Add carries will flip further bits



Chemical Registers

Registers:

- One MS-FF for each bit, chained
- Extra variable to check if empty
- Only the first bit can be flipped manually
- Add carries will flip further bits



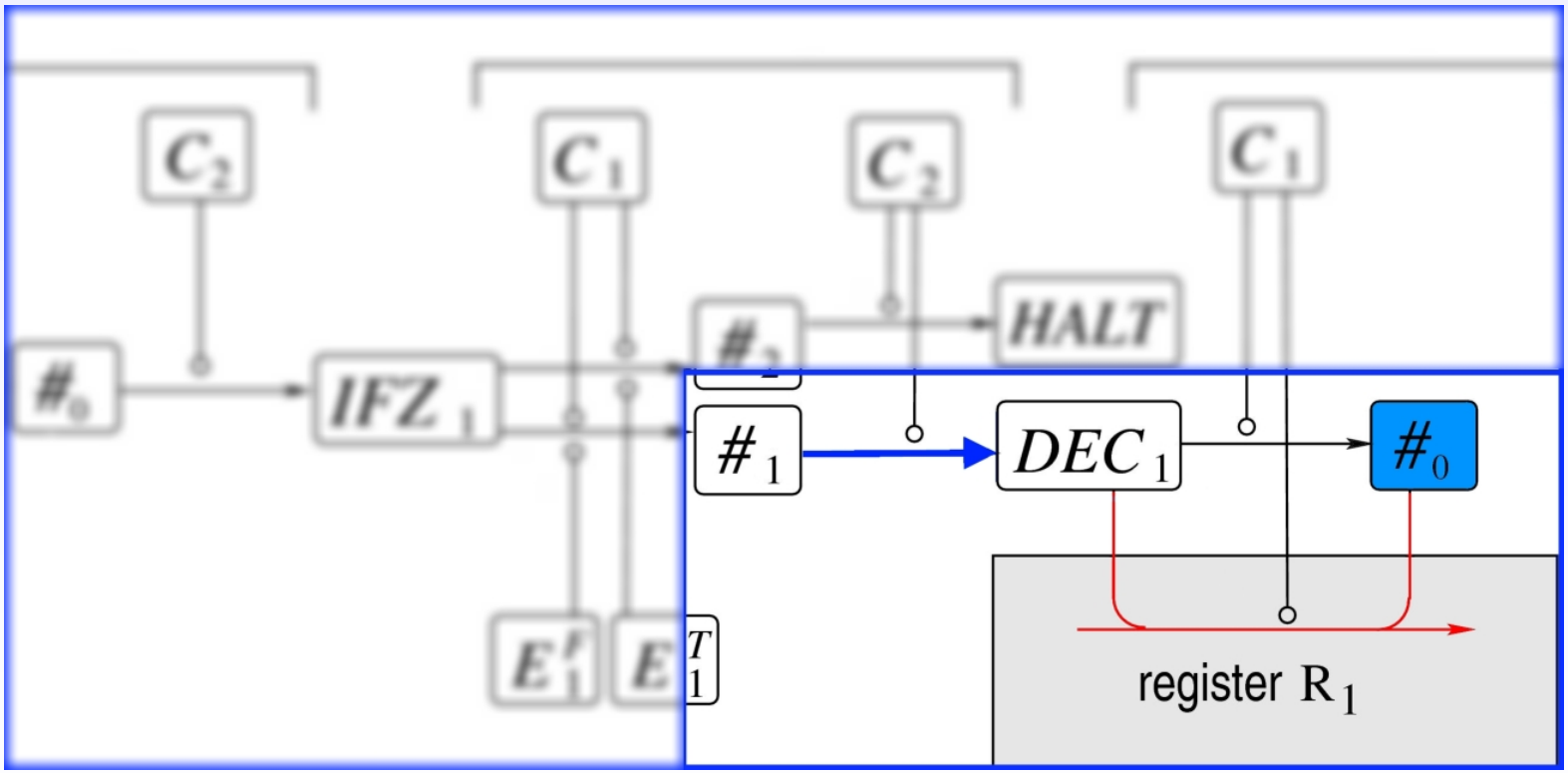
Chemical Program Control

3 Elements per Instruction: $\#_1 : \mathrm{DEC}\ \mathrm{R}_1\ \#_0$

Address - reaction that produces instruction species

Action - reaction that alters/tests register

Follow up - species encoding the next label



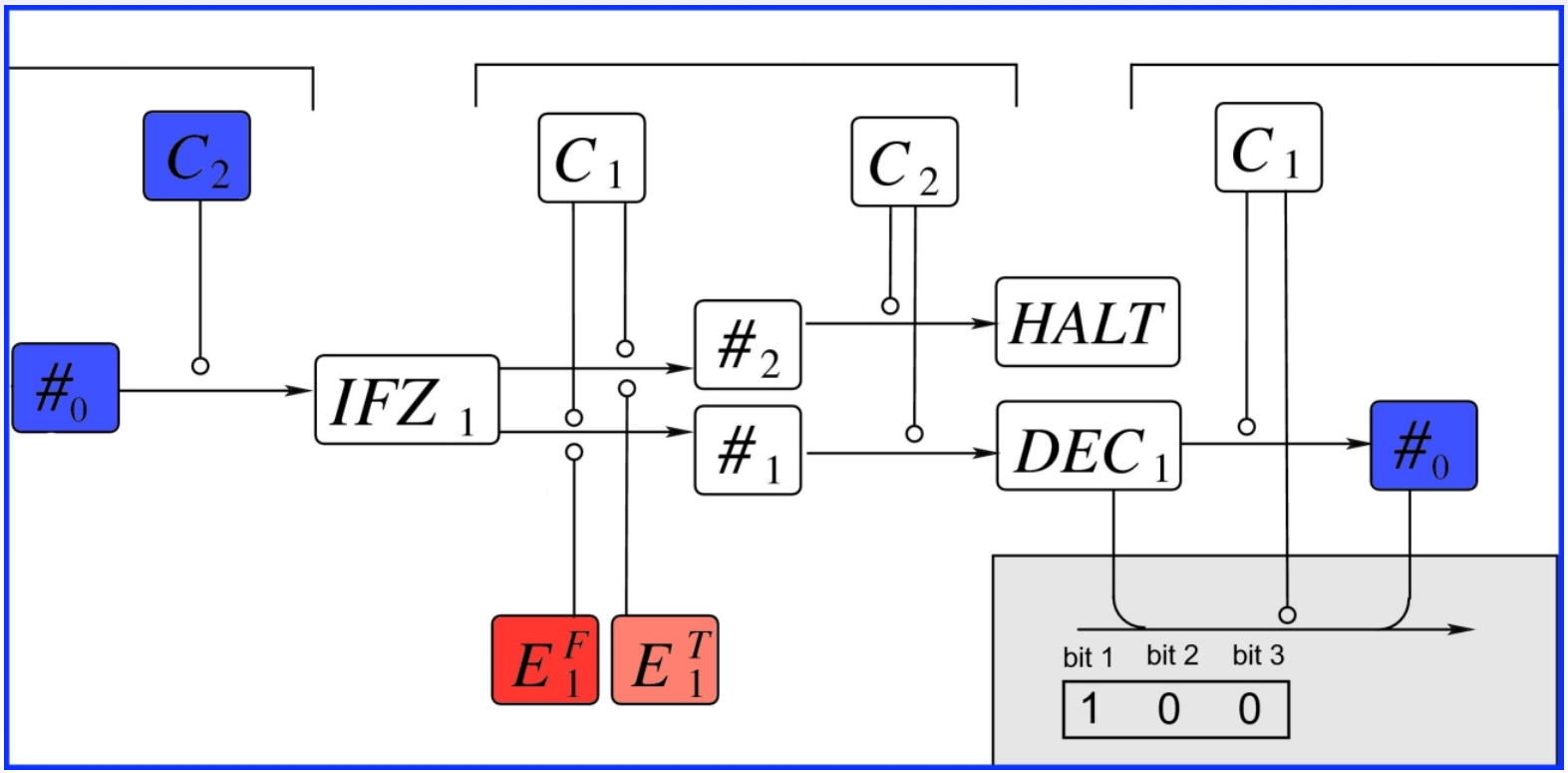
Chemical Program Control

3 Elements per Instruction: $\#_1 : \text{DEC } \text{R}_1 \ \#_0$

Address - reaction that produces instruction species

Action - reaction that alters/tests register

Follow up - species encoding the next label



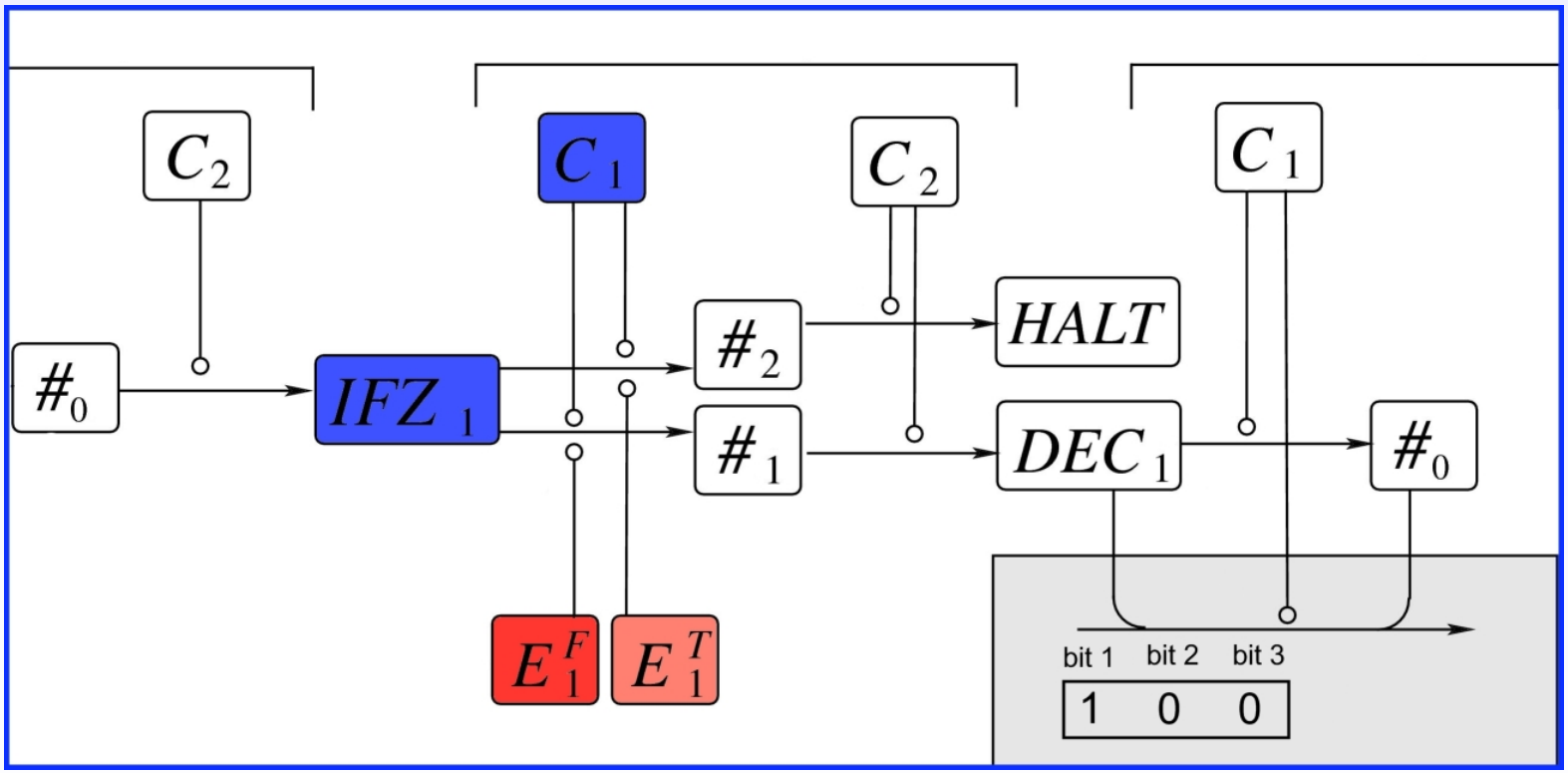
Chemical Program Control

3 Elements per Instruction: $\#_1 : \text{DEC } \text{R}_1 \ \#_0$

Address - reaction that produces instruction species

Action - reaction that alters/tests register

Follow up - species encoding the next label



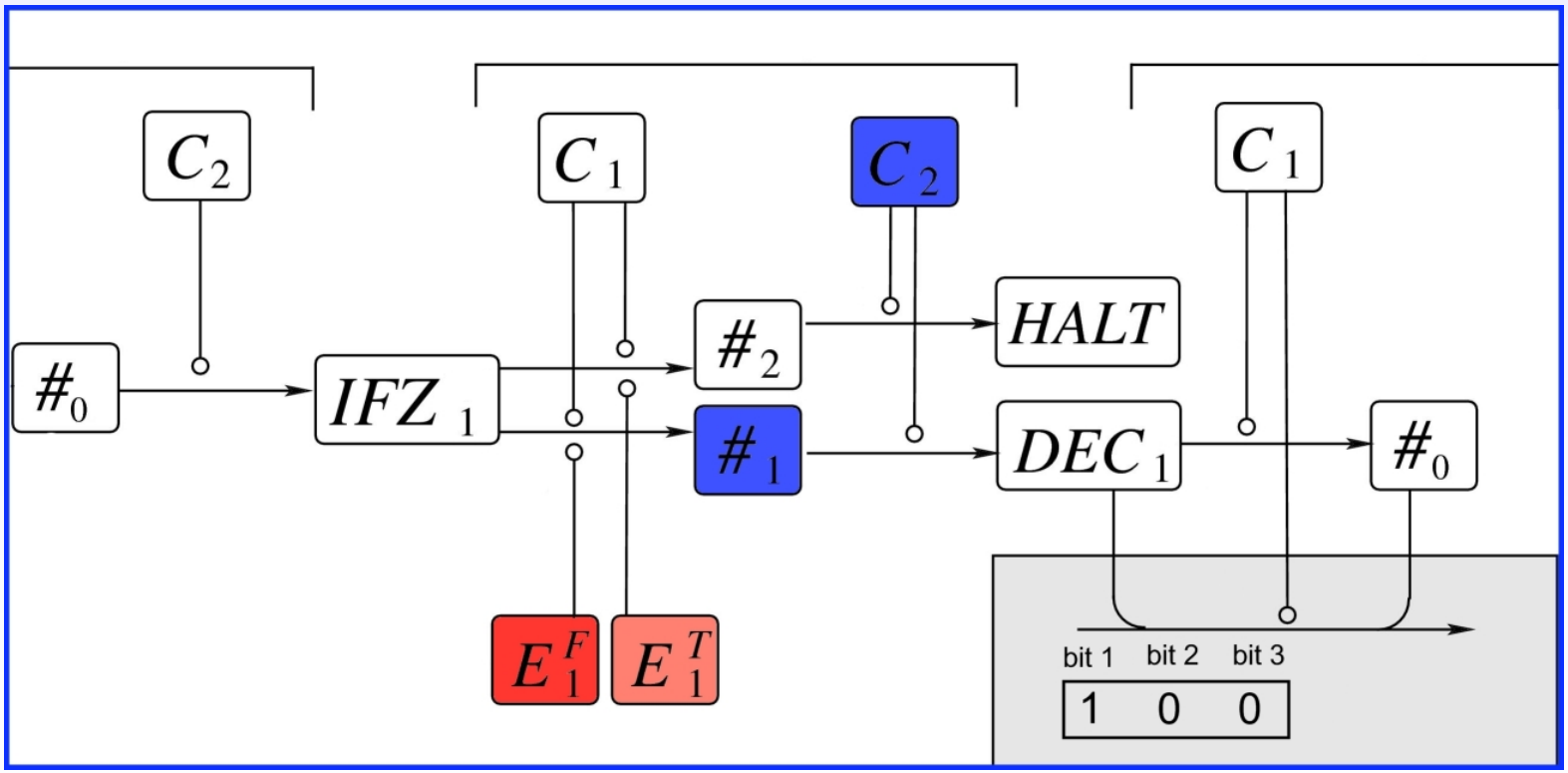
Chemical Program Control

3 Elements per Instruction: $\#_1 : \text{DEC } \text{R}_1 \ \#_0$

Address - reaction that produces instruction species

Action - reaction that alters/tests register

Follow up - species encoding the next label



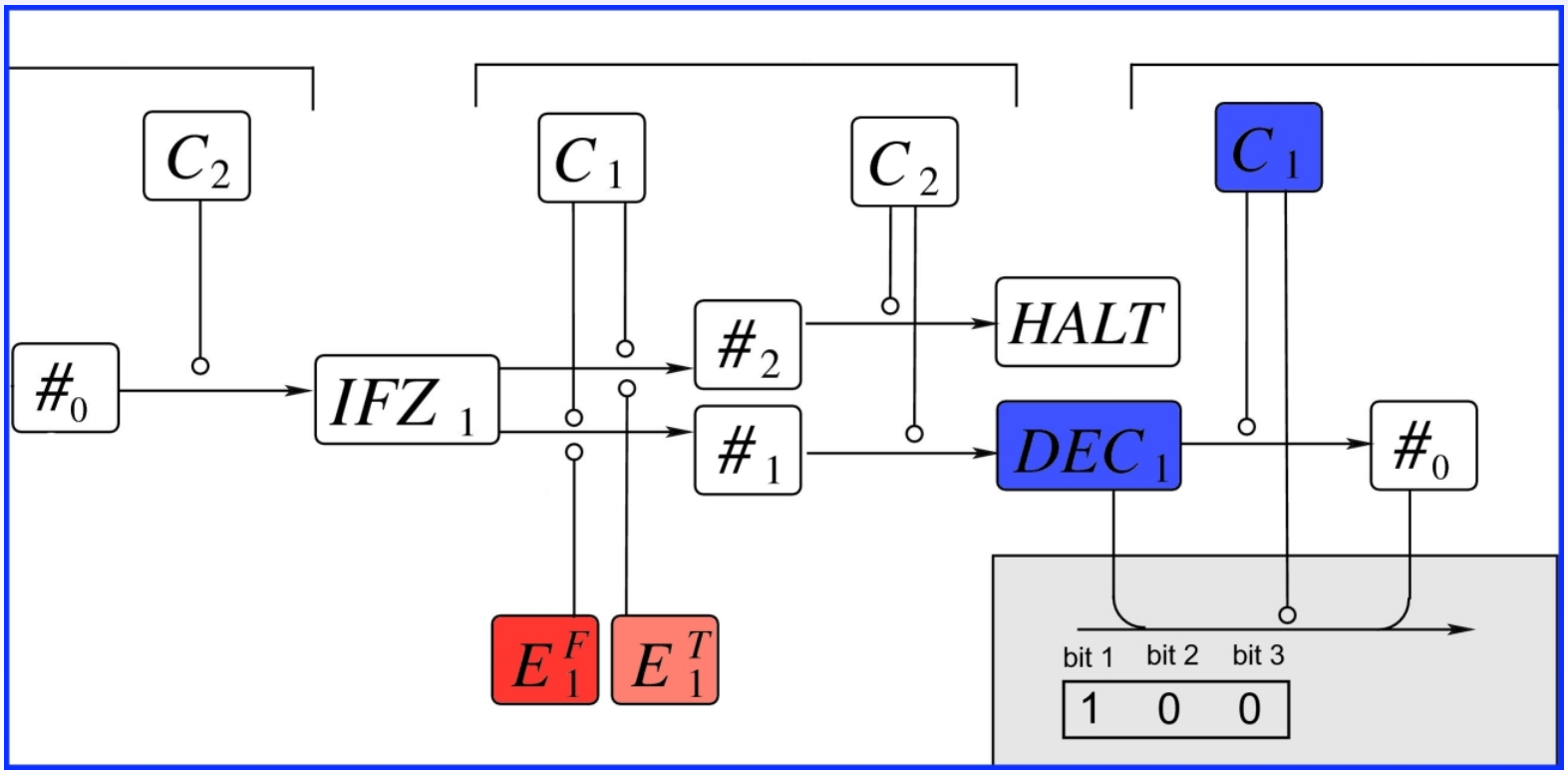
Chemical Program Control

3 Elements per Instruction: $\#_1 : \text{DEC } \mathrm{R}_1 \ \#_0$

Address - reaction that produces instruction species

Action - reaction that alters/tests register

Follow up - species encoding the next label



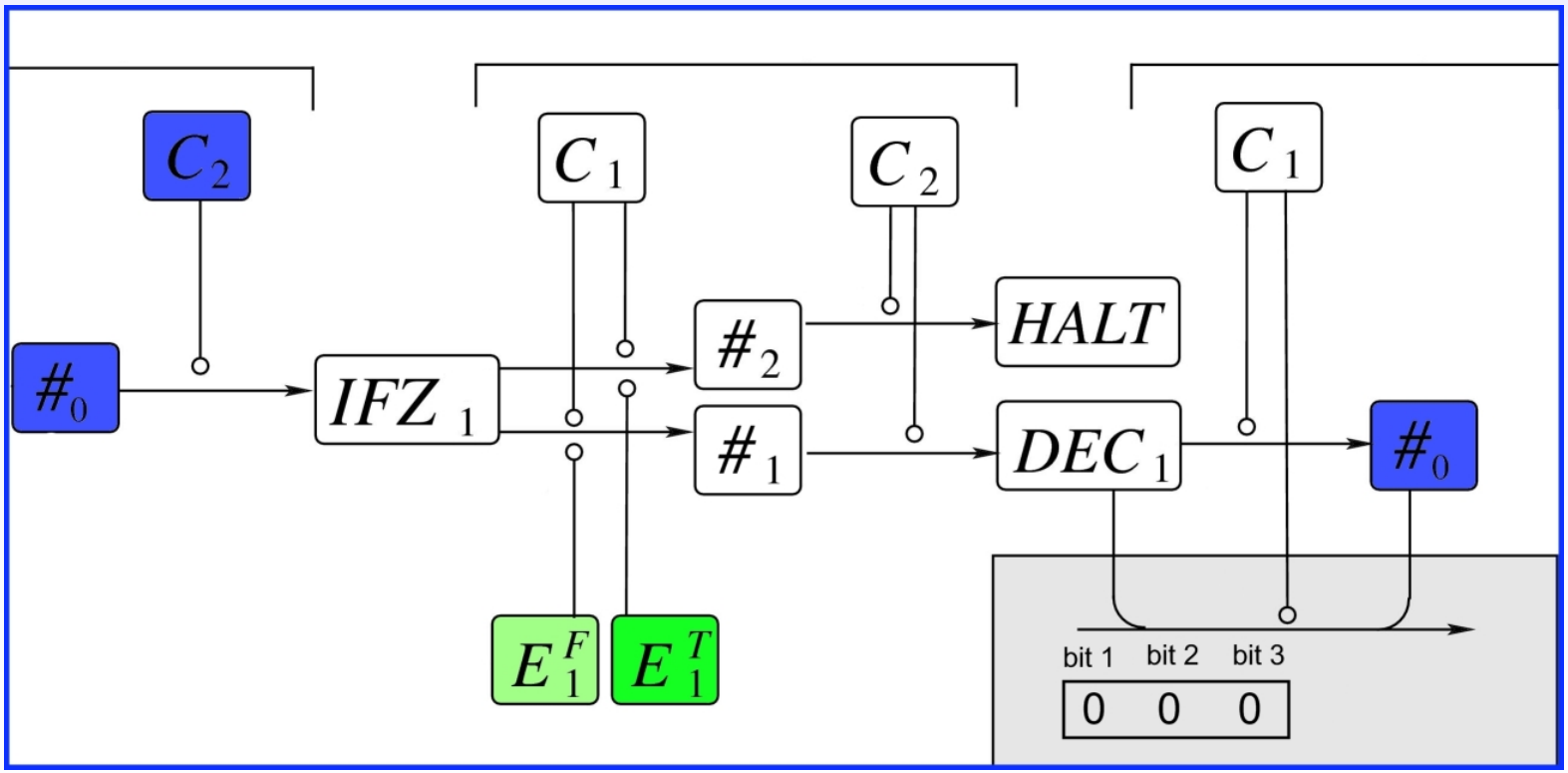
Chemical Program Control

3 Elements per Instruction: $\#_1 : \text{DEC } \text{R}_1 \ \#_0$

Address - reaction that produces instruction species

Action - reaction that alters/tests register

Follow up - species encoding the next label



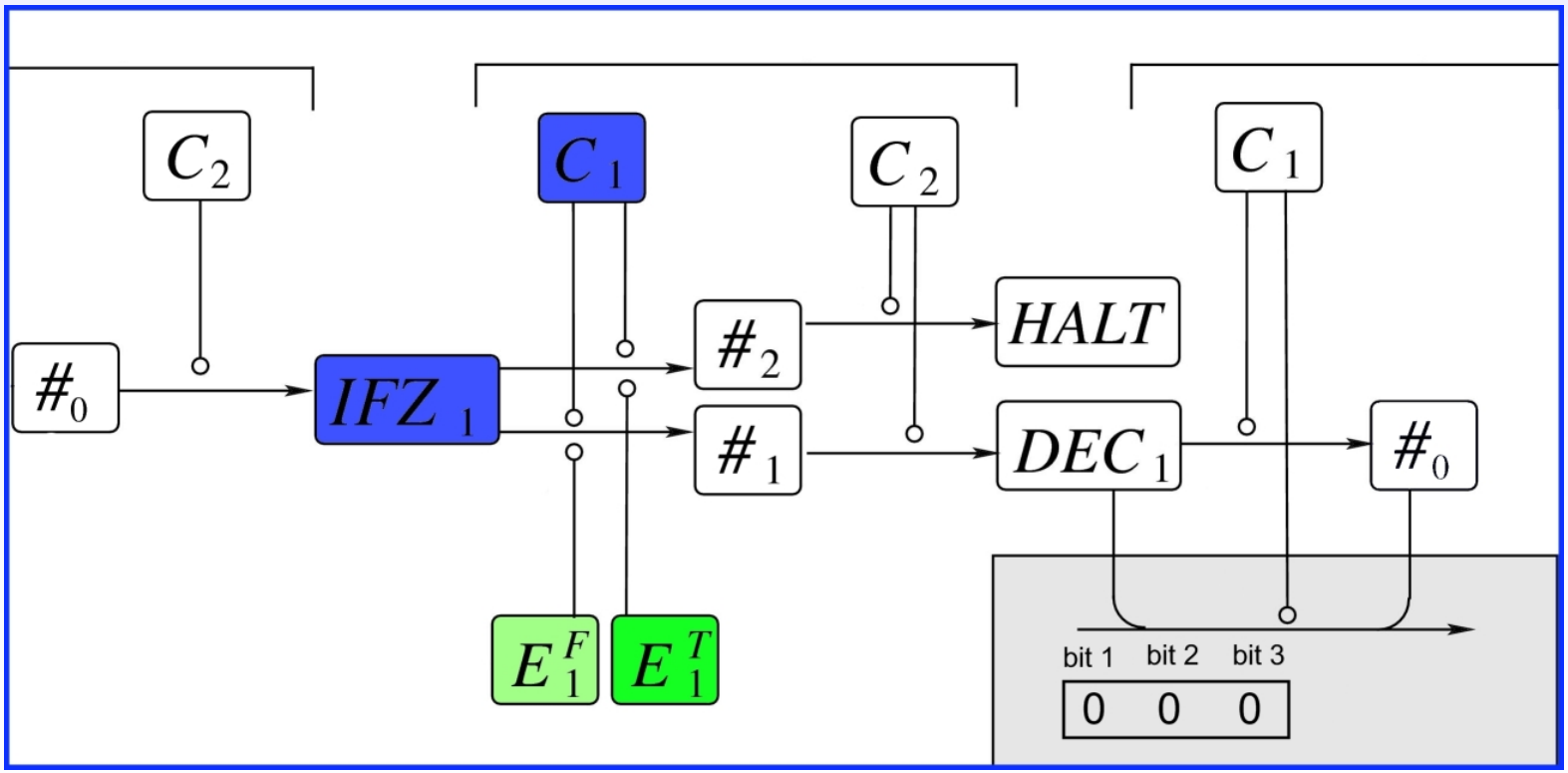
Chemical Program Control

3 Elements per Instruction: $\#_1 : \text{DEC } \text{R}_1 \ \#_0$

Address - reaction that produces instruction species

Action - reaction that alters/tests register

Follow up - species encoding the next label



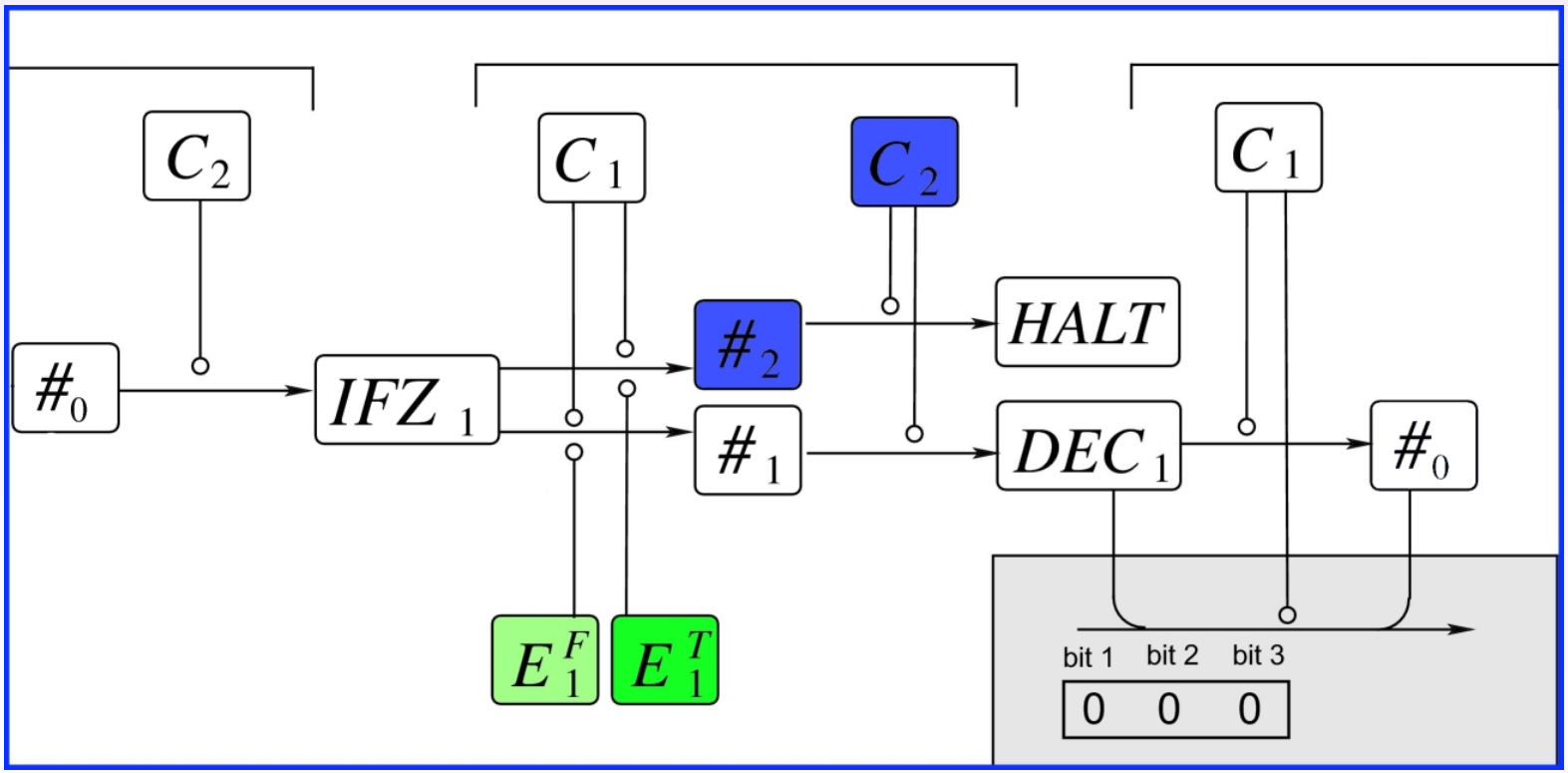
Chemical Program Control

3 Elements per Instruction: $\#_1 : \text{DEC } \text{R}_1 \ \#_0$

Address - reaction that produces instruction species

Action - reaction that alters/tests register

Follow up - species encoding the next label



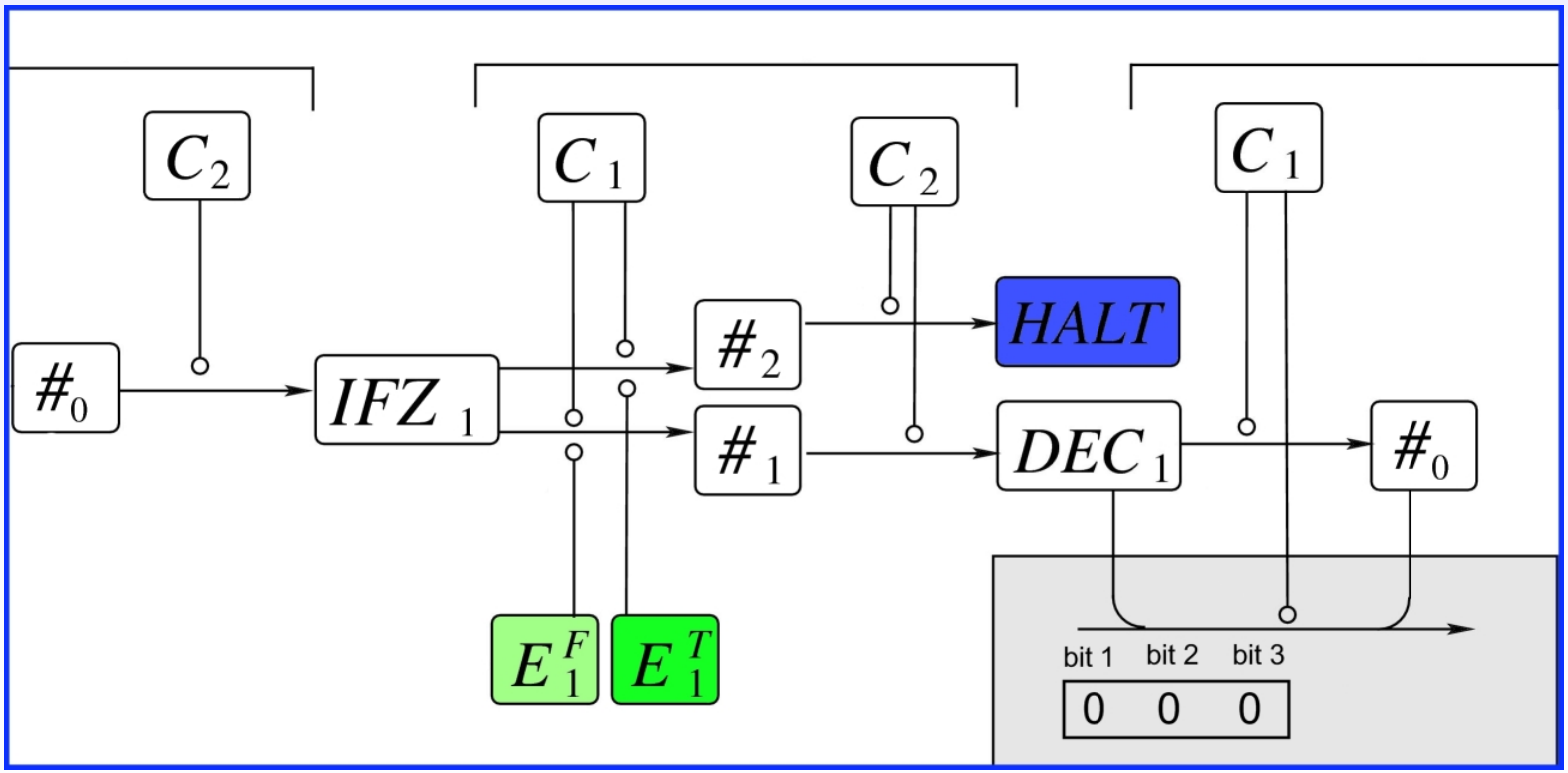
Chemical Program Control

3 Elements per Instruction: $\#_1 : \text{DEC } \text{R}_1 \ \#_0$

Address - reaction that produces instruction species

Action - reaction that alters/tests register

Follow up - species encoding the next label



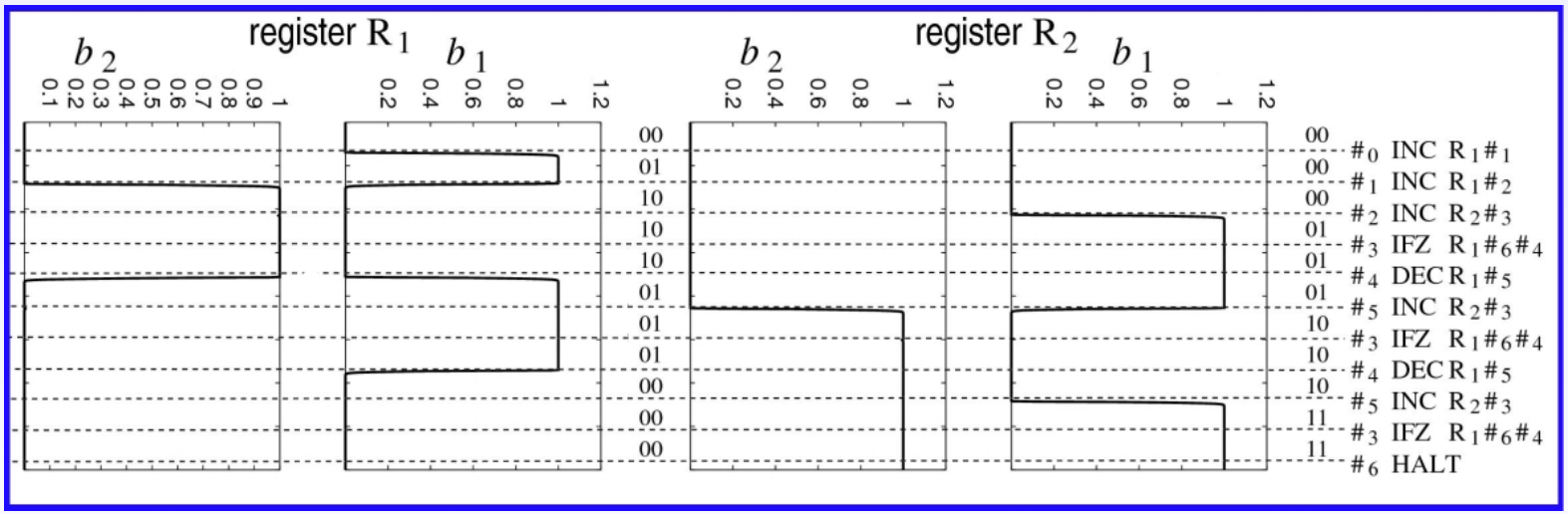
Case Study

Implemented Examples:

- Counter mod 4
- Integer addition
- Max(x,y,z)

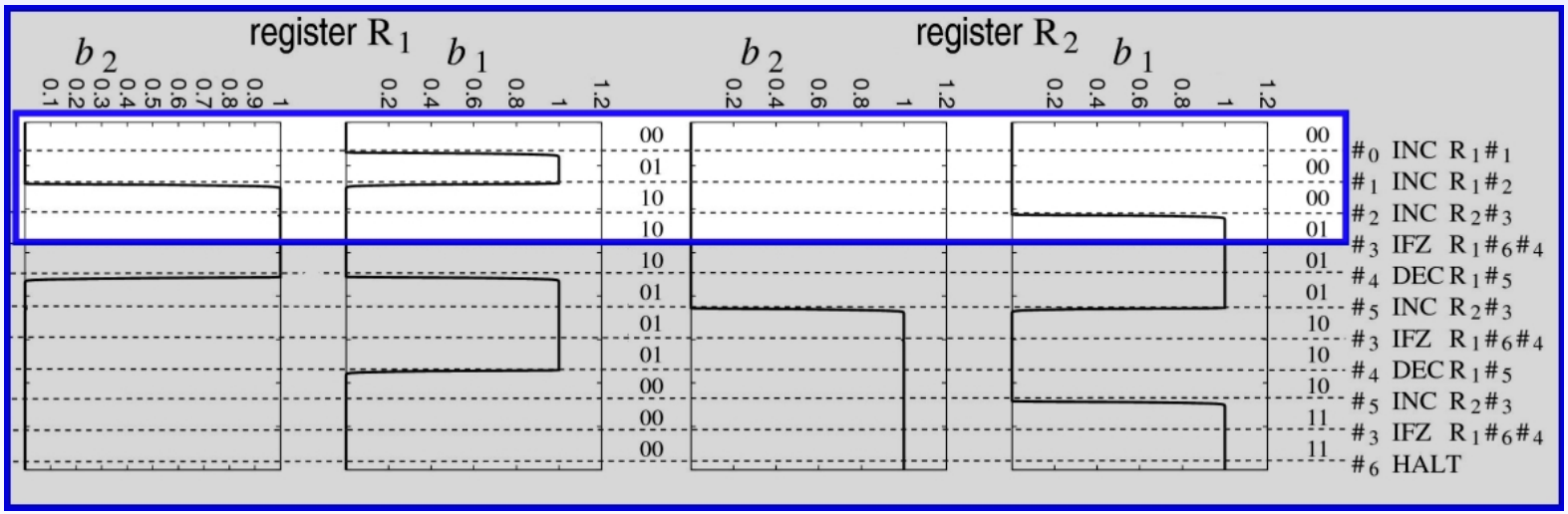
Case Study

Integer Addition: $2 + 1 = 3$



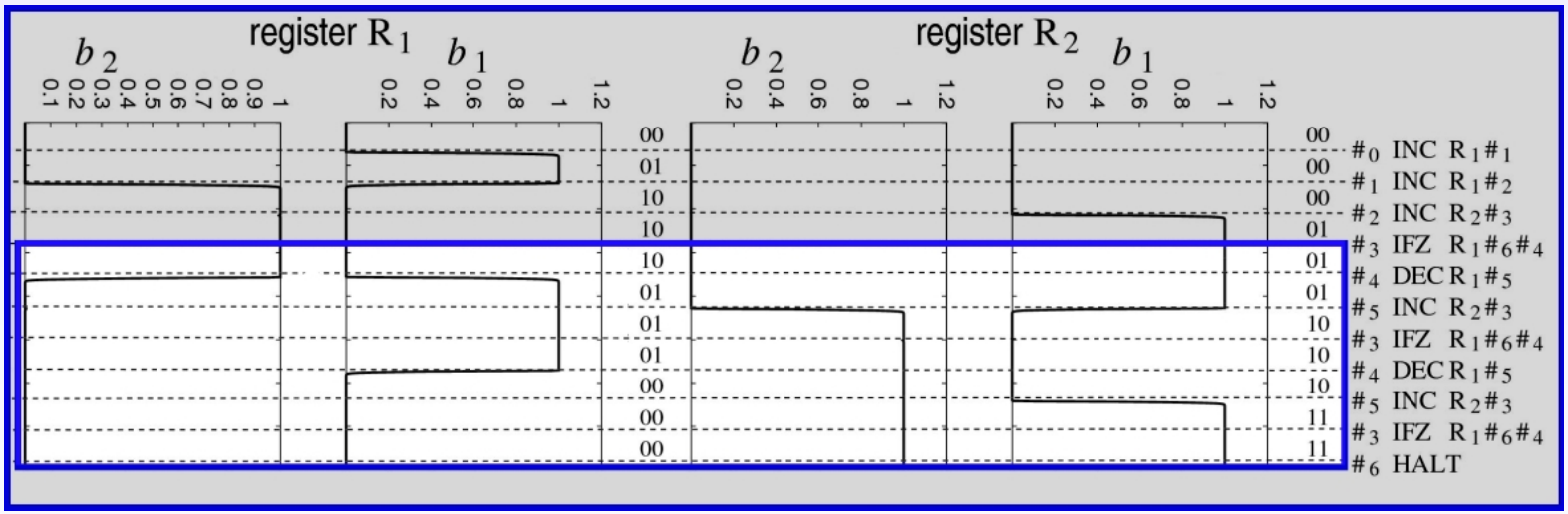
Integer Addition

Initialization: $2 + 1 = 3$



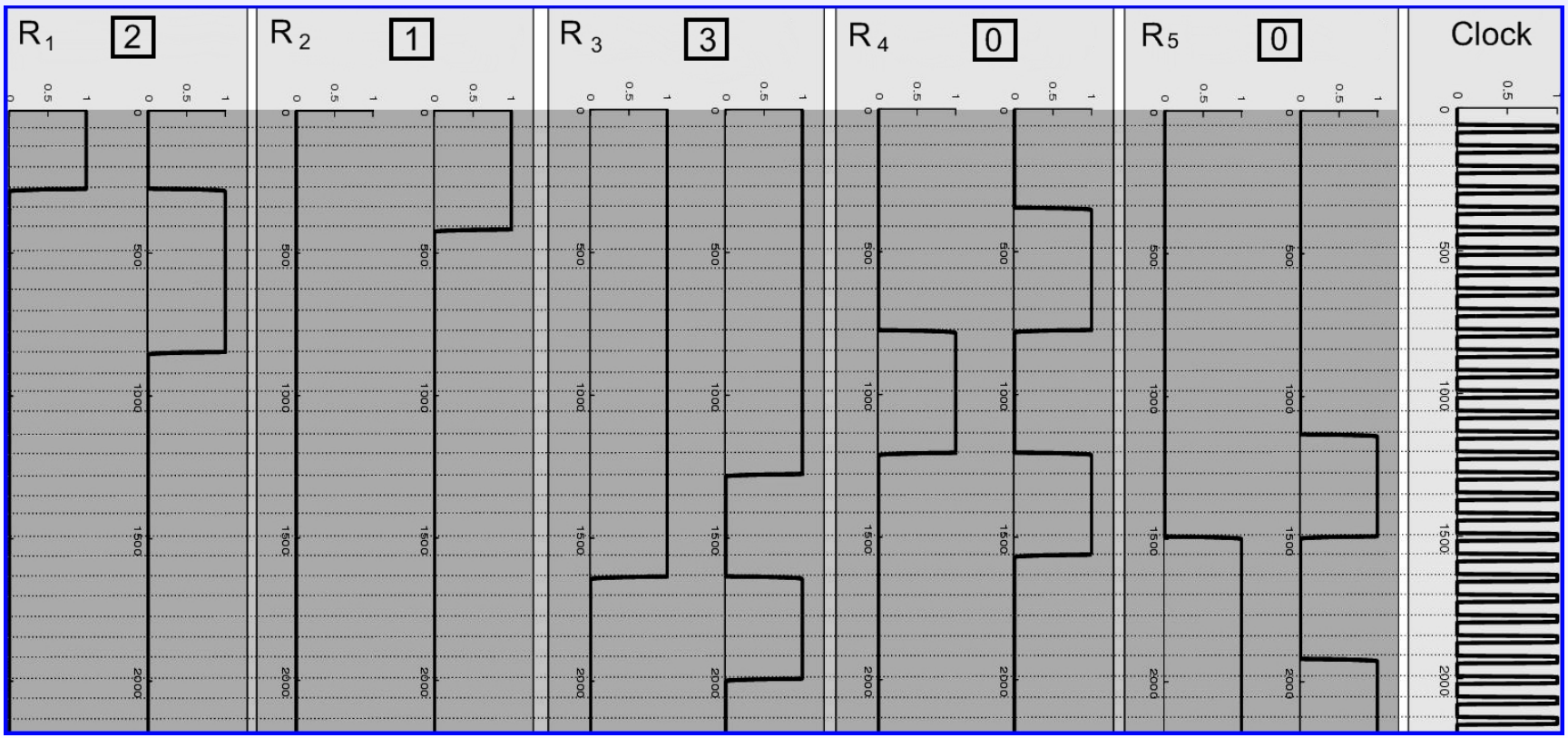
Integer Addition

Computation: $2 + 1 = 3$



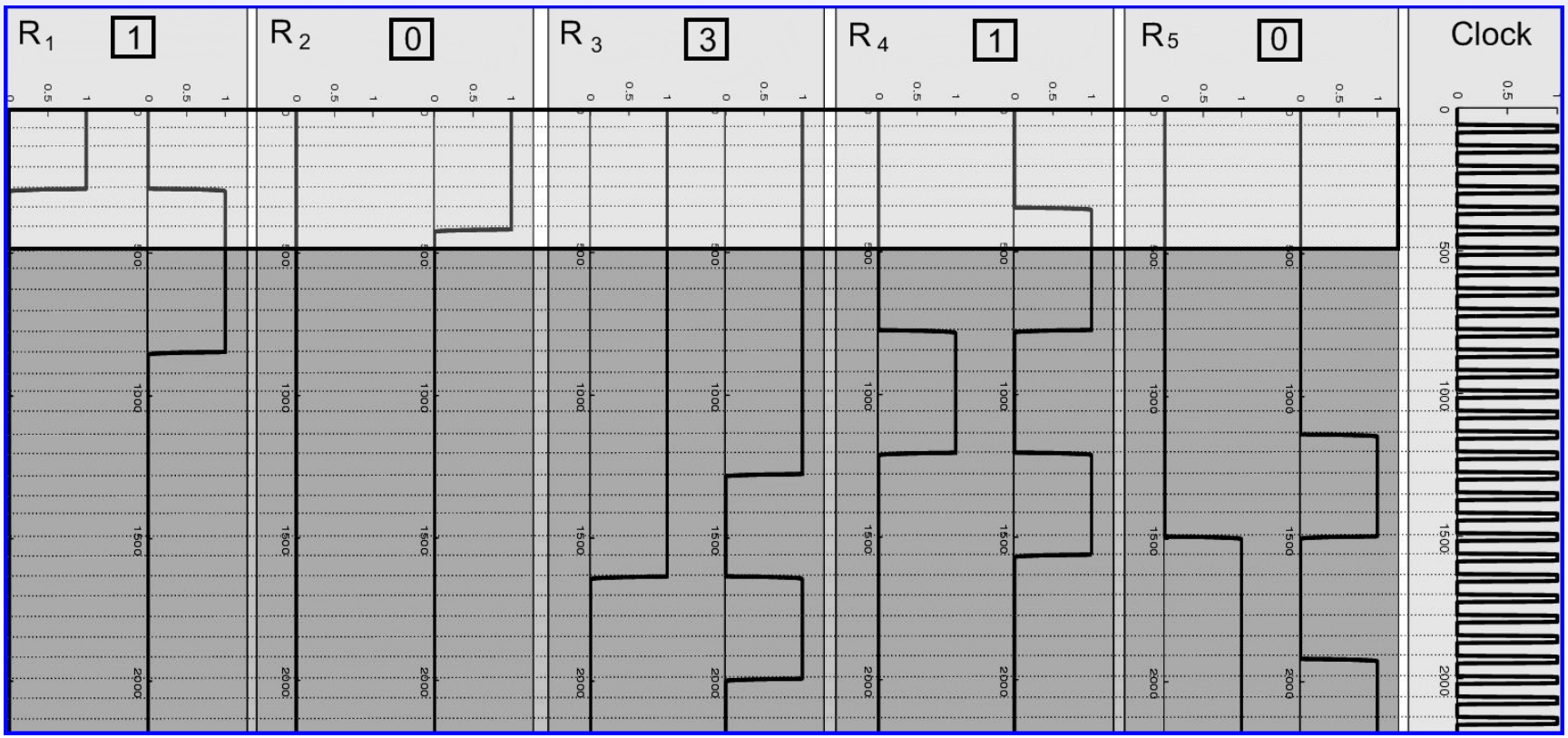
Maximum of Three Natural Numbers

Computation: $max(2, 1, 3)$



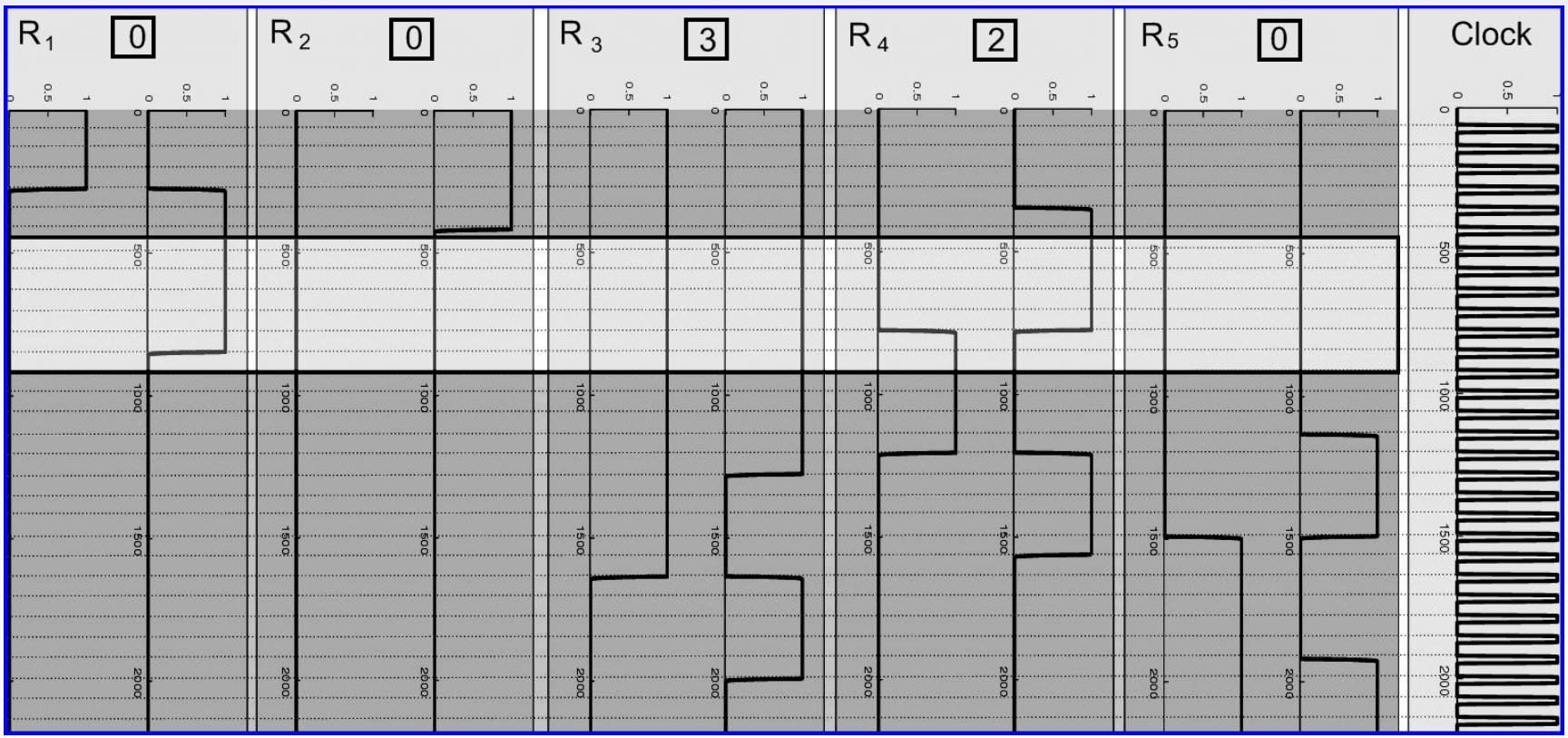
Maximum of Three Natural Numbers

Computation: $max(2, 1, 3)$



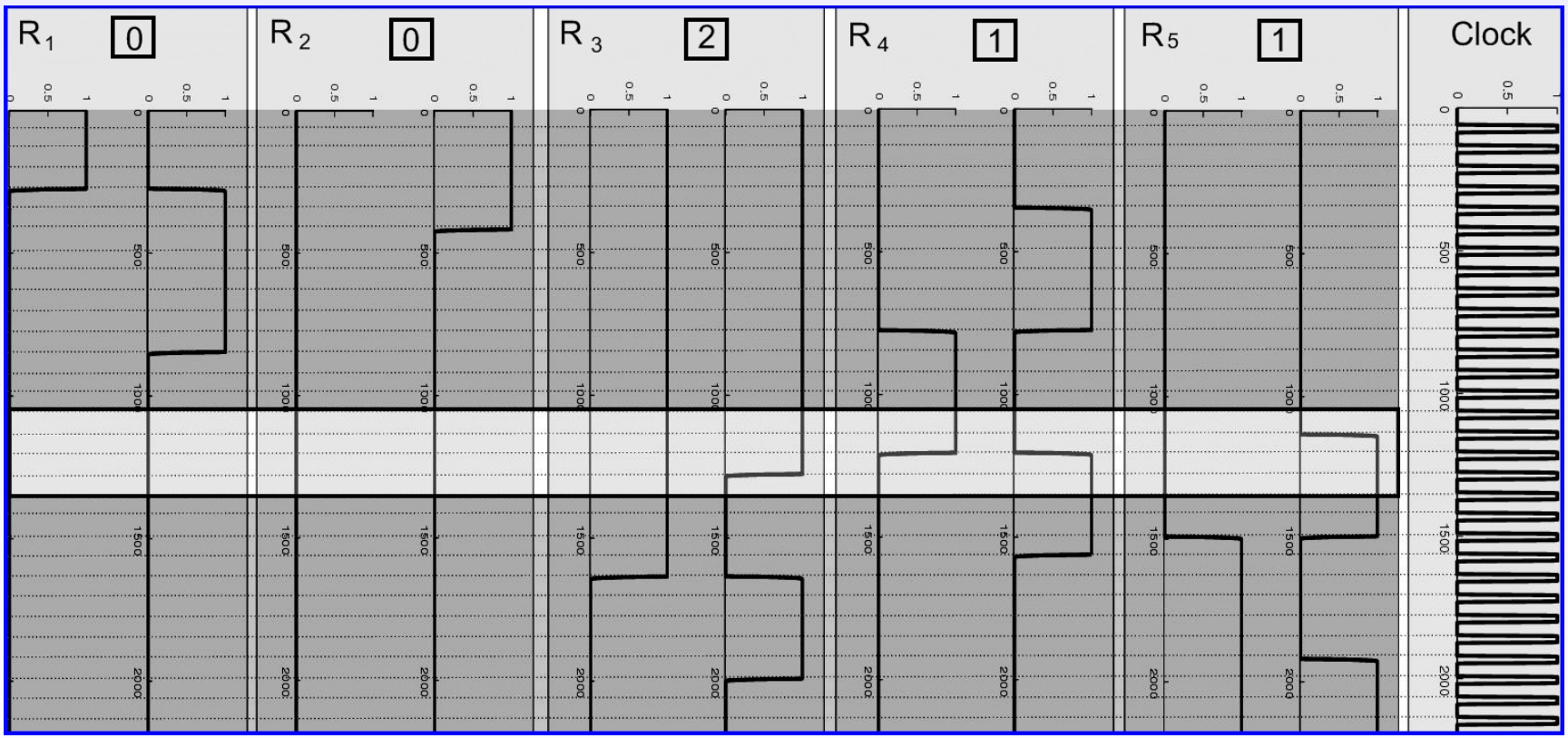
Maximum of Three Natural Numbers

Computation: $max(2, 1, 3)$



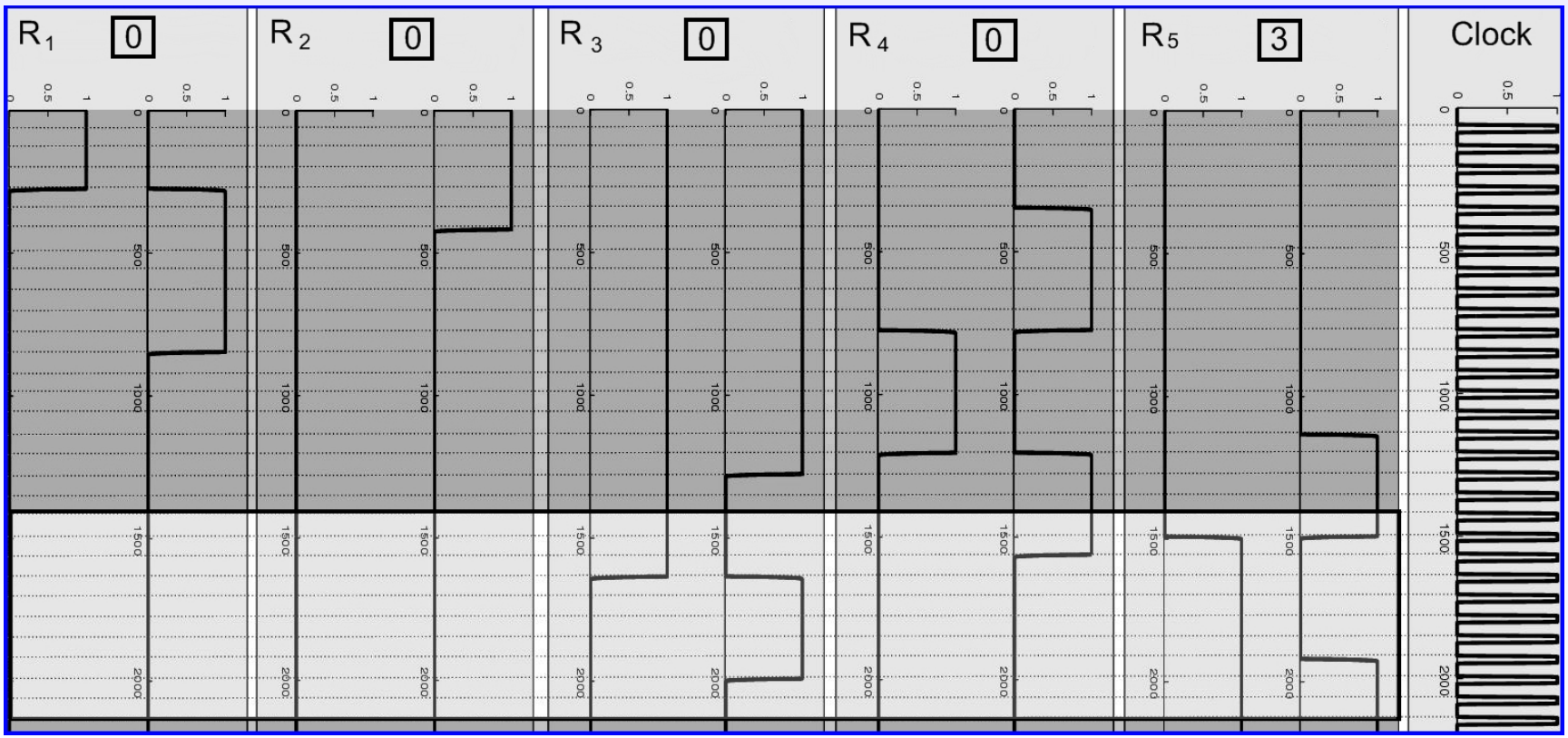
Maximum of Three Natural Numbers

Computation: $max(2, 1, 3)$



Maximum of Three Natural Numbers

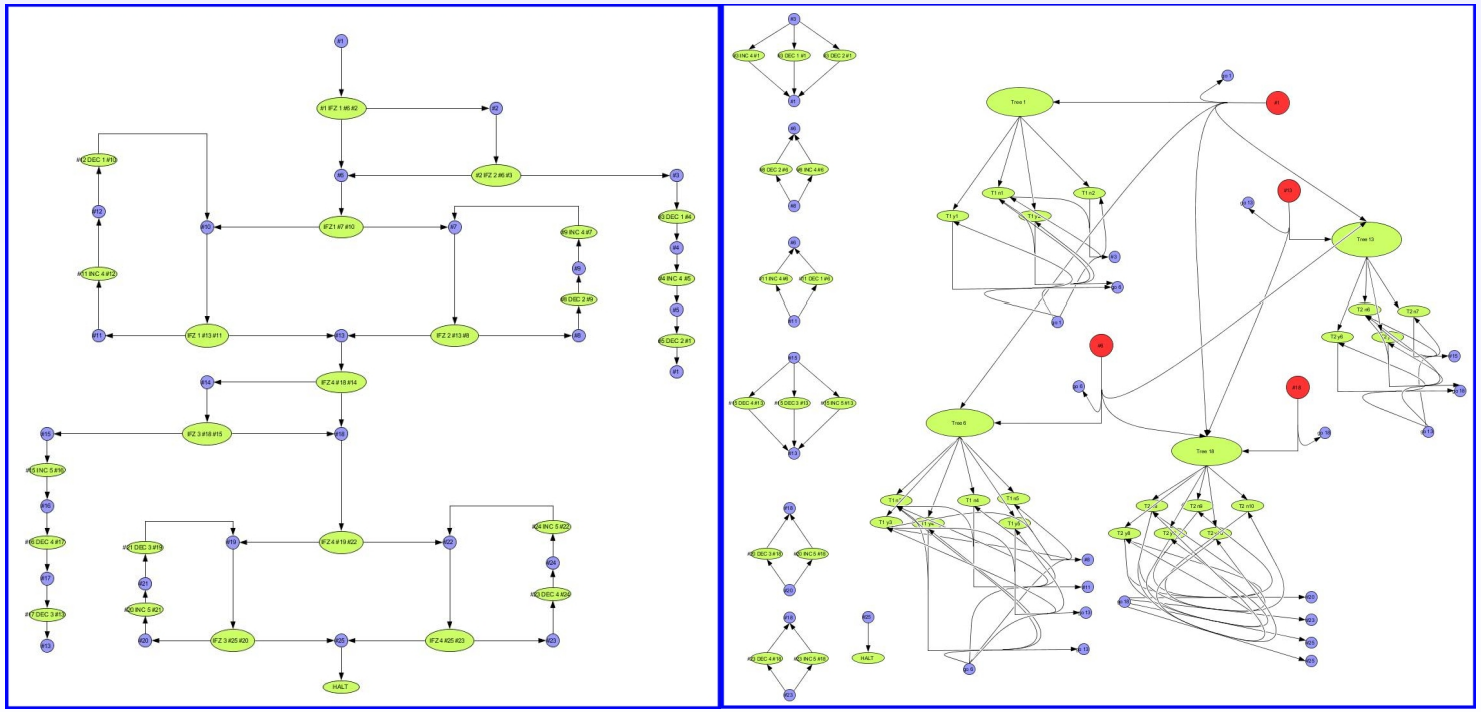
Computation: $max(2, 1, 3)$



Future Work

Parallelization:

- Consecutive read/write operations are merged into threads
- Operations in an thread are performed parallel
- Threads take the same time as a single operation



Thank you for your attention!