Multiple Side Linear Equations and Circuit Lattices

Igor Semaev
University of Bergen
Norway

CECC, 25 June 2009

Problem

- X Boolean variable set, size n
- f_i Boolean polynomials in $X_i \subseteq X$
- Find all $0,1$-solutions to

$$f_1(X_1) = 0, \ldots, f_m(X_m) = 0$$

- were $|X_i| \leq l$ for small $l = 3, 4, \ldots$
- No other restrictions
- E.g. TRIVIUM: 951 Boolean variables and equations
- each depends on 6 variables

Zakrevskij-Raddum Representation of Equations

- In [Zakrevskij,1999]
- Independently [Raddum,2004]
- $f_i(X_i) = 0 \Leftrightarrow$ solutions V_i in variables $X_i \Leftrightarrow E_i = (X_i, V_i)$
- E.g.

$$x_1x_2 + x_3 \equiv 0 \mod 2 \Leftrightarrow \begin{array}{lcccc} x_1 & & 0 & 0 & 1 & 1 \\ x_2 & = & 0 & 1 & 0 & 1 \\ x_3 & & 0 & 0 & 0 & 1 \end{array}$$

- Solve $E_1, \ldots, E_m$ by guessing some variable values and check by Pairwise Agreeing (a kind of simplification, called local reduction by Zakrevskij and graph algorithm by Raddum)
- Combine equations by Gluing [Semaev,2005]

Agreeing-Gluing family

- Expected complexity much lower than worst case bounds, [Semaev,2007-08]
- Practically, Linear Algebra variant(MRHS) is far better than F4 in Magma
- E.g., more than 20000 times faster on AES-type random eqations with 48 Boolean variables, [Raddum-Semaev,2007]. MiniSat should be slow here, as the problem is not sparse in common sense
- Overcomes MiniSat in small sparsity, as 3,4,5, for randomly generated common sparse equations, [Schilling, in progress]
- Reasons for further development

Contribution Outline

Part I: New Gluing

- Coset Covers
- New representation: Multiple side linear equations
- New Gluing Algorithm

Part II: Hardware Agreeing

- Fast pairwise agreeing [Raddum-Semaev,2007]
- Circuit Lattice Implementation
- DES and TripleDES equation systems

Part I
New Gluing Algorithm

Motivating Example: Quadratics in Trivium

- $xy + u + v + w + z = 0 \Rightarrow$ Gröbner Basis Algorithms
- 32 solutions $\Rightarrow$ Agreeing-Gluing or Sat-solvers
- MRHS lin. equation $\Rightarrow$ MRHS routines [Raddum-Semaev,07]

$$\begin{array}{rcl} x & & 0\ 0\ 1\ 1 \\ y & = & 0\ 1\ 0\ 1 \\ u+v+w+z & & 0\ 0\ 0\ 1 \end{array}$$

- MS linear equation [This talk]

$$\begin{array}{rcl} x & & 0 \\ u+v+w+z & = & 0 \end{array}$$

$$\begin{array}{rcl} x & & 1 \\ y+u+v+w+z & = & 0 \end{array}$$

Coset Cover Problem

- n-bit Boolean vector space S_n
- Vector b , subspace U, coset $b + U \subseteq S_n$,
- V any subset in S_n
- Problem: find a coset cover

$$V = \{b_1 + U_1\} \cup \ldots \cup \{b_t + U_t\}$$

  with minimal $t = t(V)$. Seems hard to solve for large n

- 1. $t(V) \leq \lceil \frac{|V|}{2} \rceil$
  2. $t(V) \geq 2^{n/2-1}$ for some V, where n even
  3. $n = 4$, always $t(V) \leq 4$
  4. and mean of $t(V)$ is 2.71

Represent Coset by Linear Equations

- $B = b + \langle u_1, \ldots, u_r \rangle$ coset, $\dim B = r$
- n-variable column vector X
- B all solutions to $AX = a$
- Matrix A: $n - r$ rows and n columns
- $rank(A) = n - r$
- Point:
  1. $AX = a$ more compact than B
  2. easy to operate with linear algebra routines

MS Linear Equation

- $f(X_i) = 0$ Boolean equation with low $|X_i| = l$
- V all solutions in variables X_i
- Compute coset cover

$$V = \{b_1 + U_1\} \cup \ldots \cup \{b_t + U_t\}$$

- Get MS linear equation

$$\begin{aligned} A_1 X_i &= a_1 \\ \ldots & \\ A_t X_i &= a_t \end{aligned}$$

Example: one Boolean equation in 4 variables X_i

solutions in X_i		cosets		MS linear equation
		$\begin{matrix}0&0&0&0\\0&0&0&1\\0&1&0&0\\0&1&0&1\end{matrix}$	$\Leftrightarrow \begin{pmatrix}1&0&0&0\\0&0&1&0\end{pmatrix} X_i = \begin{pmatrix}0\\0\end{pmatrix}$
$\begin{matrix}0&0&0&0\\0&0&0&1\\0&1&0&0\\0&1&0&1\\0&1&1&1\\1&1&0&1\\1&0&1&1\\1&1&1&1\end{matrix}$	$\Leftrightarrow$	$\begin{matrix}0&1&1&1\\1&0&1&1\end{matrix}$	$\Leftrightarrow \begin{pmatrix}1&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix} X_i = \begin{pmatrix}1\\1\\1\end{pmatrix}$
		$\begin{matrix}1&1&0&1\\1&1&1&1\end{matrix}$	$\Leftrightarrow \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\end{pmatrix} X_i = \begin{pmatrix}1\\1\\1\end{pmatrix}$

Standard Gluing Algorithm

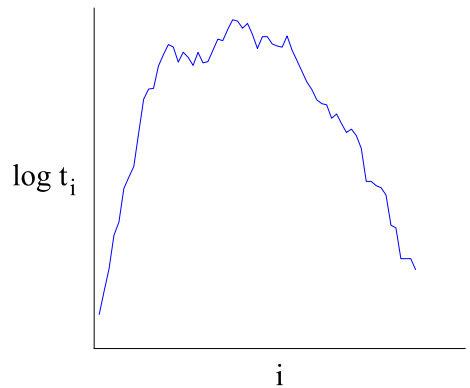
- $f_1(X_1) = 0$ with solutions V_1 in X_1
- $f_2(X_2) = 0$ with solutions V_2 in X_2
- Compute common solutions V to the system

$$f_1(X_1) = 0$$
$$f_2(X_2) = 0$$

  in variables $X' = X_1 \cup X_2$
- Apply to solve the whole system

Gluing Running Time

- Characterized by $T = (t_1, t_2, \ldots, t_m)$
- t_i -number of solutions to $f_1(X_1) = 0, \ldots, f_i(X_i) = 0$ in variables $X_1 \cup \ldots \cup X_i$, for $i = 1, 2, \ldots, m$



- Asymptotic complexity under uniform instances distribution, [Semaev,2007]

l	3	4	5	6
max t_i, expectation	1.262^n	1.355^n	1.425^n	1.479^n
Worst case [Iwama,2004]	1.324^n	1.474^n	1.569^n	1.637^n

New Gluing Algorithm

- $f_1(X_1) = 0$ with $A_1X_1 = a_1, \ldots, A_tX_1 = a_t$
- $f_2(X_2) = 0$ with $B_1X_2 = b_1, \ldots, B_sX_2 = b_s$
- For each i, j triangulate

$$\begin{aligned} A_iX_1 &= a_i \\ B_jX_2 &= b_j \end{aligned}$$

- Discard if inconsistent. Keep if consistent
- Produces the list

$$C_1X' = c_1, \ldots, C_kX' = c_k$$

  for common solutions in variables $X' = X_1 \cup X_2$
- Apply to solve the whole system

New Gluing Running Time

- Characterized by $S = (s_1, s_2, \ldots, s_m)$
- s_i -number of linear systems for

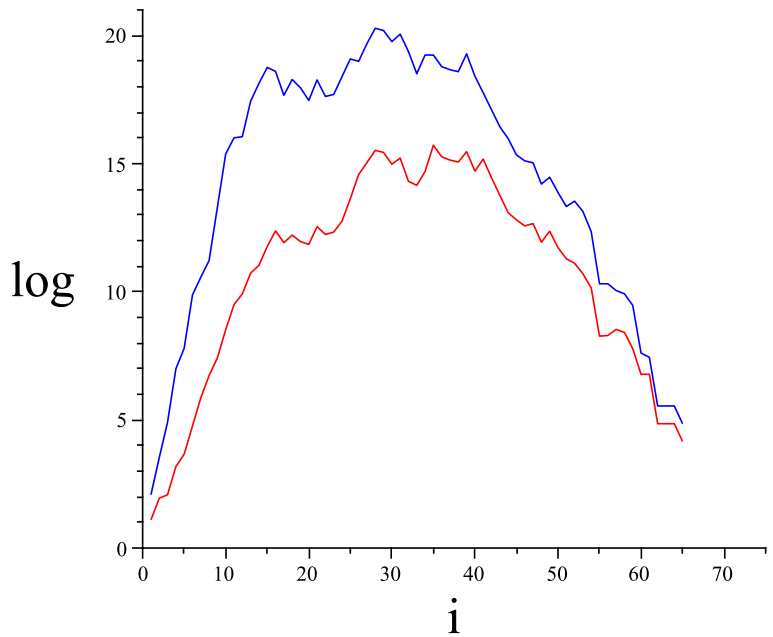
$$f_1(X_1) = 0, \ldots, f_i(X_i) = 0$$

  produced with New Gluing Algorithm, $i = 1, 2 \ldots, m$
- Running time roughly $\max s_i$. Difficult to estimate in theory
- Compare T and S experimentally
- $n = m = 75$ and $l = 4$, uniformly random instance
- With Maple linear algebra routines

Experimental Evidence

$$T = (8, 34, 136, 1088, 2432, 19456, 38912, .., \mathbf{645404416}, .., 0)$$
$$S = (3, 7, 8, 24, 117, 351, 837, .., \mathbf{6733966}, .., 0)$$

- Logarithmic scale; blue T-curve and red S-curve



- S-curve is generated 4 times faster than T-curve
- New Gluing is faster

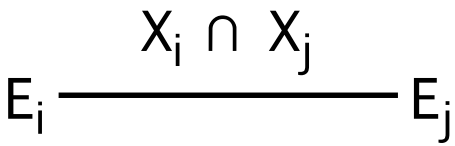
Research Directions

- Get asymptotic complexity bound
- Develop a combination with Agreeing

Part II
Agreeing with Circuit Lattices

Equation System Graph and Pairwise Agreeing

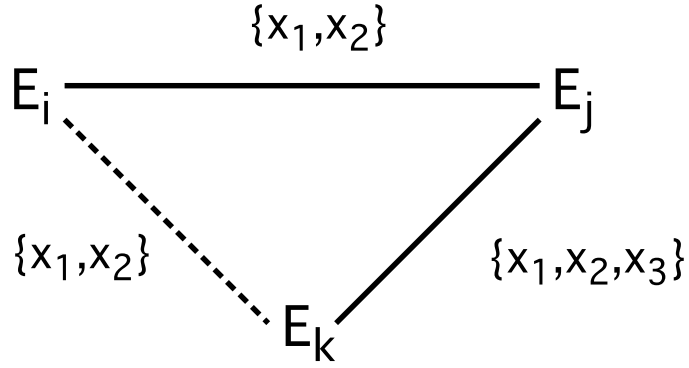
- **System**: $f_1(X_1) = 0, \ldots, f_m(X_m) = 0$
- $f_i(X_i) = 0 \Leftrightarrow$ solutions V_i in variables $X_i \Leftrightarrow E_i = (X_i, V_i)$
- Connect $E_i = (X_i, V_i)$ and $E_j = (X_j, V_j)$ by
- Edge labeled $X_i \cap X_j \neq \emptyset$



- **Pairwise Agreeing.** Let $Y \subseteq X_i \cap X_j$
- Learn ban $Y \neq a$ from E_i
- Expand to E_j through the edge
- Modify E_j accordingly

Obsolescent Edges

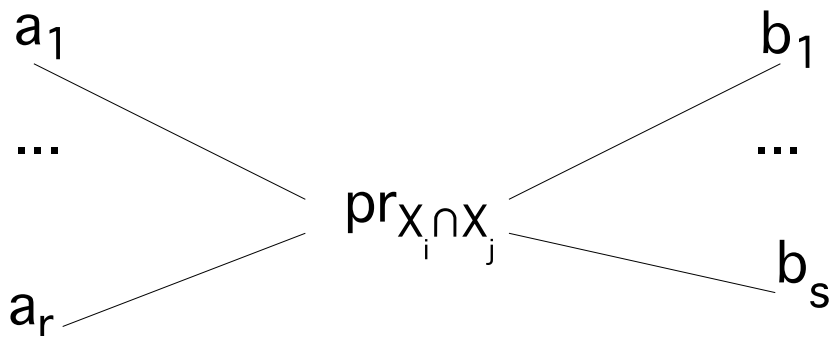
- Remove some edges and keep Algorithm's output
- E.g. $X_i = \{x_1, x_2, x_4\}, X_j = \{x_1, x_2, x_3, x_5\}, X_k = \{x_1, x_2, x_3\}$



- Left edges called maximal
- 16831 edges in Triple DES equation system initially
- 3929 maximal (left after removals)

Fast Pairwise Agreeing (Agreeing2 method)

- For maximal edges (E_i, E_j)
- $a_1, \ldots, a_r$ and $b_1, \ldots, b_s$ solutions to E_i and E_j with the same projection to $X_i \cap X_j$



- Pre-compute all $\{a_1, \ldots, a_r; b_1, \ldots, b_s\}$

Fast Pairwise Agreeing (Agreeing2 method)

- $a_i \neq$ part of a global solution $\Rightarrow$ mark $\bar{a}_i$
- For each tuple $\{a_1, \ldots, a_r; b_1, \ldots, b_s\}$:
- $\bar{a}_1, \ldots, \bar{a}_r \Rightarrow \bar{b}_1, \ldots, \bar{b}_s$ and vice versa $\bar{b}_1, \ldots, \bar{b}_s \Rightarrow \bar{a}_1, \ldots, \bar{a}_r$
- Solving the system:
- Introducing a guess $\equiv$ marking some of a_i
- Expand marking through the tuples

Example

- Equations

	a_1	a_2	a_3
x_1	0	0	1
x_2	0	1	1
x_3	1	1	0

,

	b_1	b_2
x_1	0	1
x_4	1	0

,

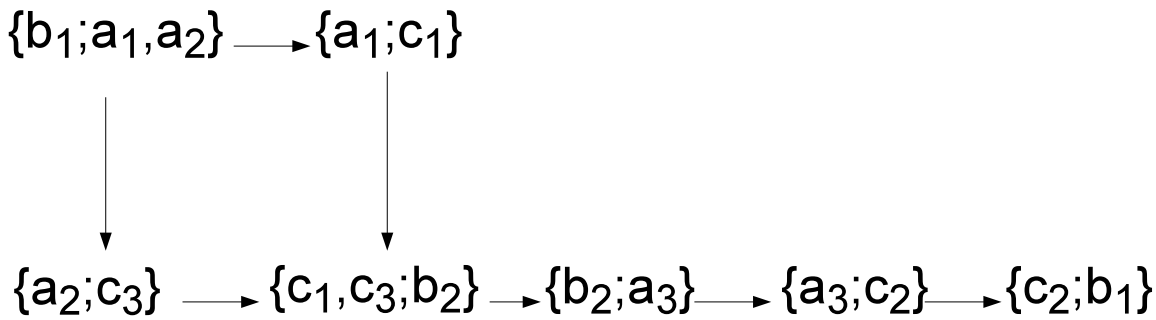
	c_1	c_2	c_3
x_2	0	1	1
x_3	1	0	1
x_4	1	1	0

- Tuples

$$\{a_1, a_2; b_1\}, \quad \{a_3; b_2\}, \quad \{b_1; c_2\},$$
$$\{b_2; c_1, c_3\}, \quad \{a_1; c_1\}, \quad \{a_2; c_3\},$$
$$\{a_3; c_2\}$$

Example

- Assume $x_4 = 0 \Rightarrow b_1$ should be marked(not a solution part)
- Implies marking expansion



- All instances(b_2 at early stage) got marked
- The system is inconsistent for $x_4 = 0$

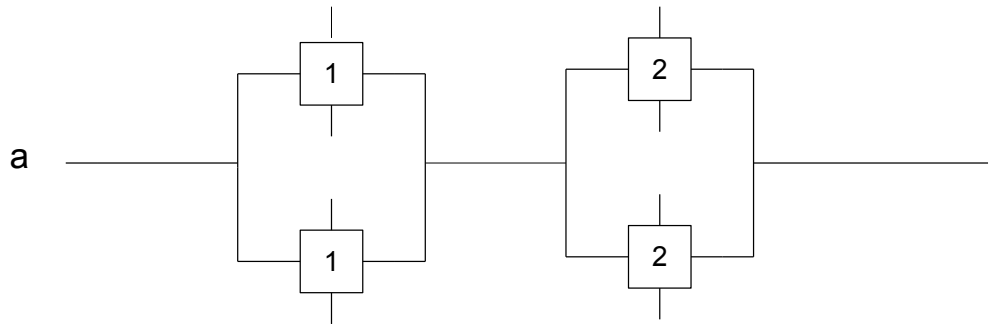
Circuit Lattice

- Hardware implementation of Agreeing2 method (different from hardware MRHS by Steinwandt, Geiselmann and Matheis)
- Circuit Lattice is a combination of switches and wires
- Two types of switches:

- Type-1 switch controls vertical circuit by the horizontal
- Type-2 switch controls horizontal circuit by the vertical

Horizontal circuits

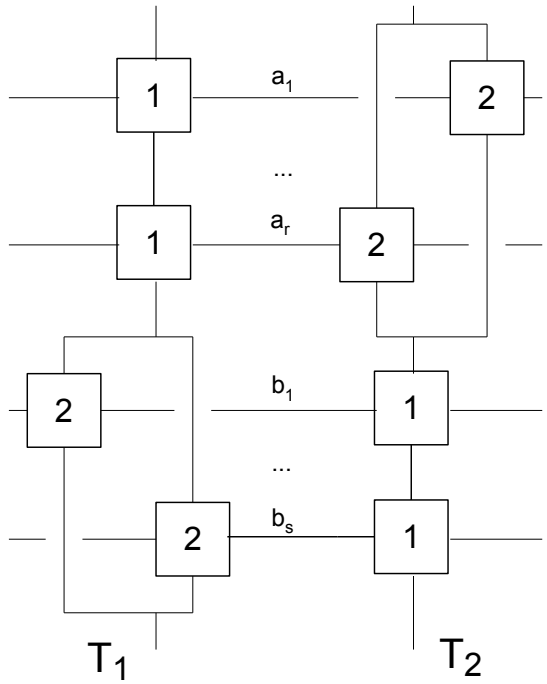
- Each local solution $a \in E_i$ determines one horizontal circuit
- Type-1 and Type-2 switches are connected in parallel



- Endings are connected with a battery

Vertical circuits

- Tuple $T = \{a_1, \ldots, a_r; b_1, \ldots, b_s\}$ defines two vertical circuits



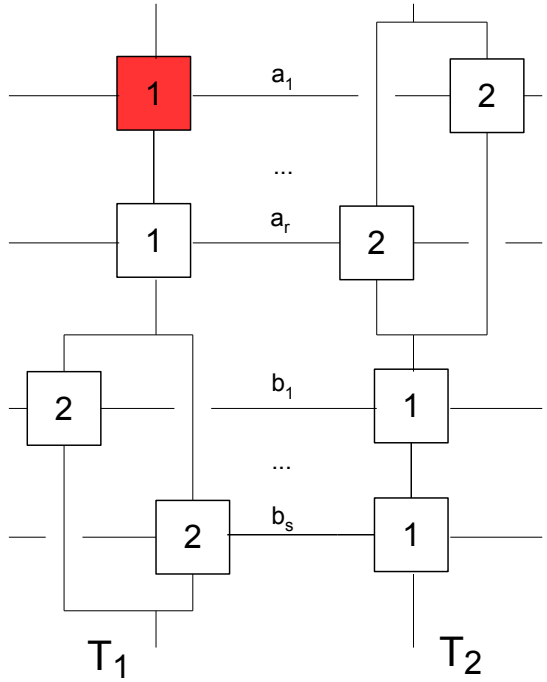
- Implement implications

$$\bar{a}_1, \ldots, \bar{a}_r \Rightarrow \bar{b}_1, \ldots, \bar{b}_s, \quad \bar{b}_1, \ldots, \bar{b}_s \Rightarrow \bar{a}_1, \ldots, \bar{a}_r$$

- All Type-1 switches closed $\Rightarrow$ Voltage appears in a vertical circuit

Vertical circuits

- Tuple $T = \{a_1, \ldots, a_r; b_1, \ldots, b_s\}$ defines two vertical circuits



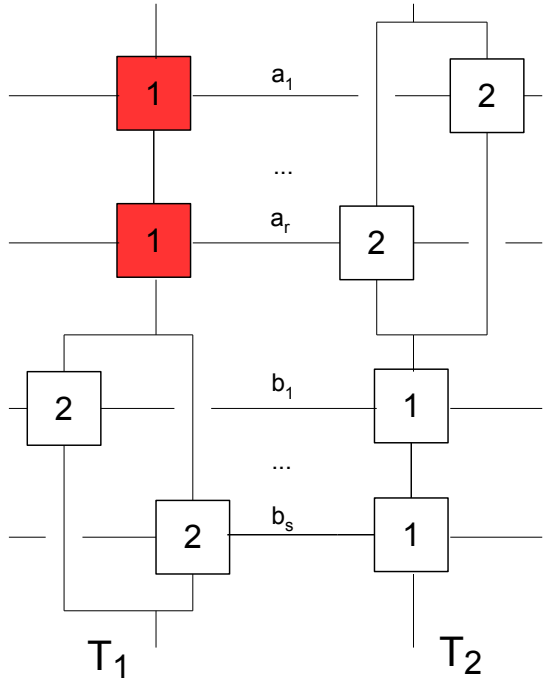
- Implement implications

$$\bar{a}_1, \ldots, \bar{a}_r \Rightarrow \bar{b}_1, \ldots, \bar{b}_s, \quad \bar{b}_1, \ldots, \bar{b}_s \Rightarrow \bar{a}_1, \ldots, \bar{a}_r$$

- All Type-1 switches closed $\Rightarrow$ Voltage appears in a vertical circuit

Vertical circuits

- Tuple $T = \{a_1, \ldots, a_r; b_1, \ldots, b_s\}$ defines two vertical circuits



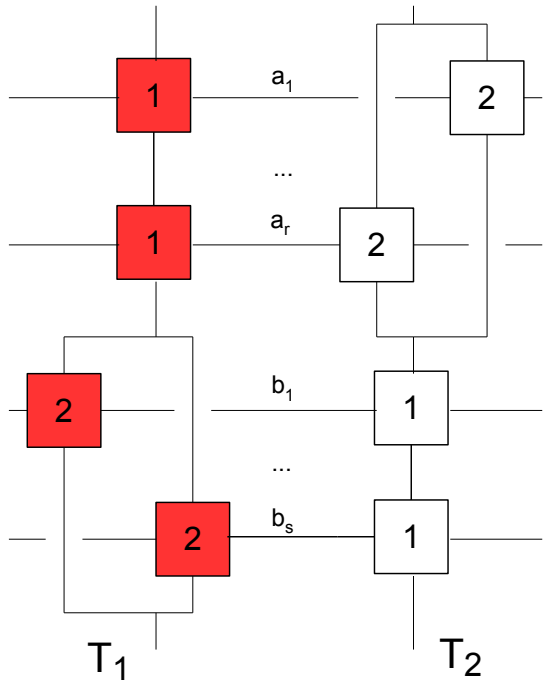
- Implement implications

$$\bar{a}_1, \ldots, \bar{a}_r \Rightarrow \bar{b}_1, \ldots, \bar{b}_s, \quad \bar{b}_1, \ldots, \bar{b}_s \Rightarrow \bar{a}_1, \ldots, \bar{a}_r$$

- All Type-1 switches closed $\Rightarrow$ Voltage appears in a vertical circuit

Vertical circuits

- Tuple $T = \{a_1, \ldots, a_r; b_1, \ldots, b_s\}$ defines two vertical circuits



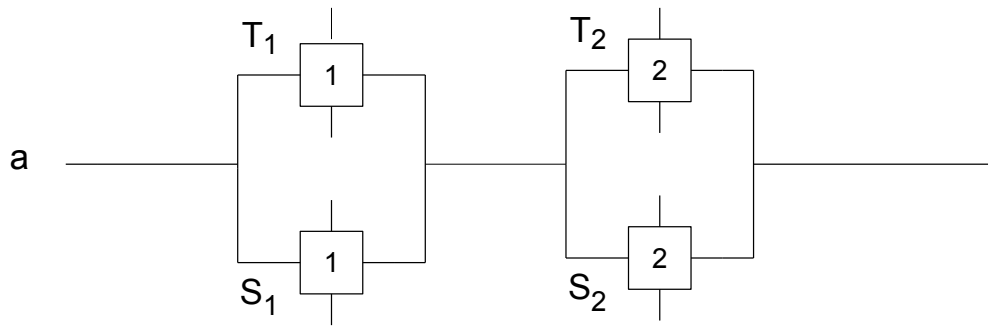
- Implement implications

$$\bar{a}_1, \ldots, \bar{a}_r \Rightarrow \bar{b}_1, \ldots, \bar{b}_s, \quad \bar{b}_1, \ldots, \bar{b}_s \Rightarrow \bar{a}_1, \ldots, \bar{a}_r$$

- All Type-1 switches closed $\Rightarrow$ Voltage appears in a vertical circuit

How horizontal circuit works

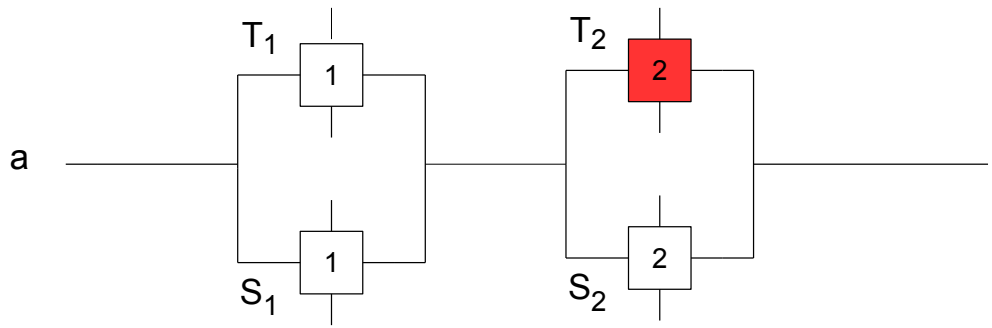
- $T = \{a, a_1, \ldots, a_2; b_1, \ldots, b_2\}$, $S = \{a, a_3, \ldots, a_4; c_1, \ldots, c_2\}$
- Define



- Voltage in vertical $T_2 \Rightarrow$ Type-2 switch closed
- Voltage in circuit $a \equiv$ marking a
- $\Rightarrow$ Type-1 switches closed
- Voltage(?) in new vertical circuit S_1

How horizontal circuit works

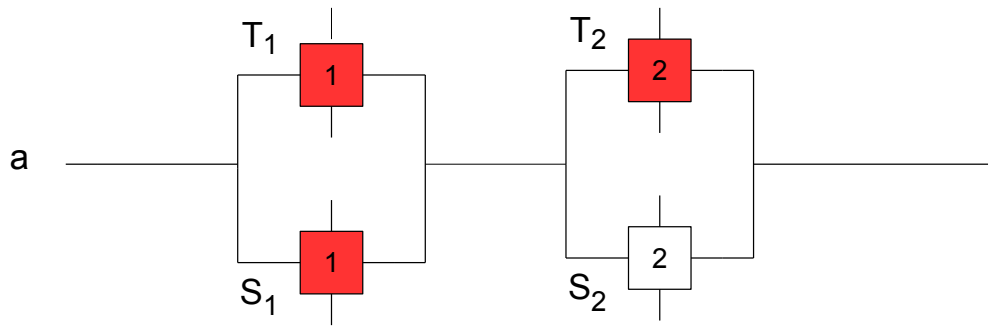
- $T = \{a, a_1, \ldots, a_2; b_1, \ldots, b_2\}$, $S = \{a, a_3, \ldots, a_4; c_1, \ldots, c_2\}$
- Define



- Voltage in vertical $T_2 \Rightarrow$ Type-2 switch closed
- Voltage in circuit $a \equiv$ marking a
- $\Rightarrow$ Type-1 switches closed
- Voltage(?) in new vertical circuit S_1

How horizontal circuit works

- $T = \{a, a_1, \ldots, a_2; b_1, \ldots, b_2\}$, $S = \{a, a_3, \ldots, a_4; c_1, \ldots, c_2\}$
- Define

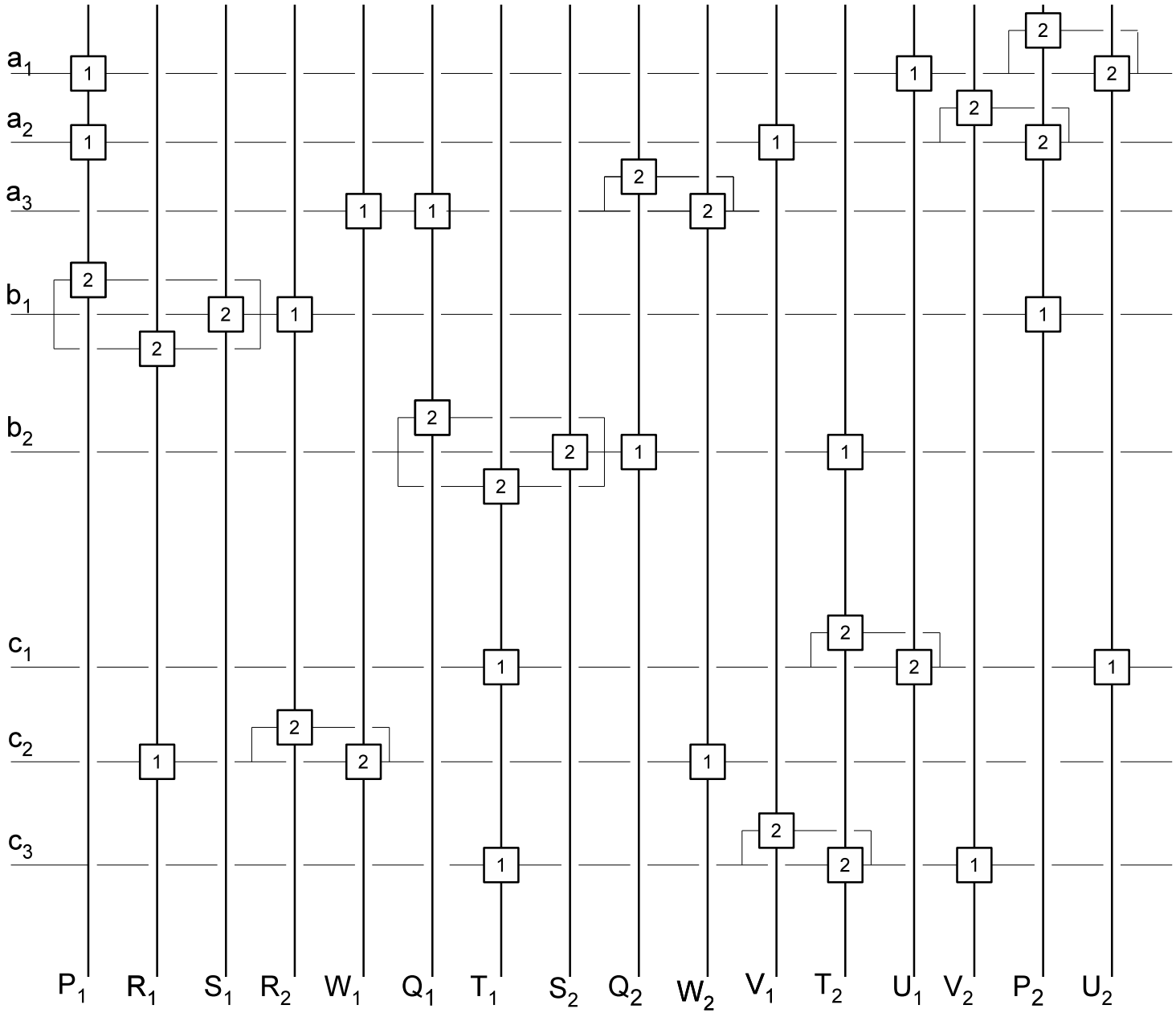


- Voltage in vertical $T_2 \Rightarrow$ Type-2 switch closed
- Voltage in circuit $a \equiv$ marking a
- $\Rightarrow$ Type-1 switches closed
- Voltage(?) in new vertical circuit S_1

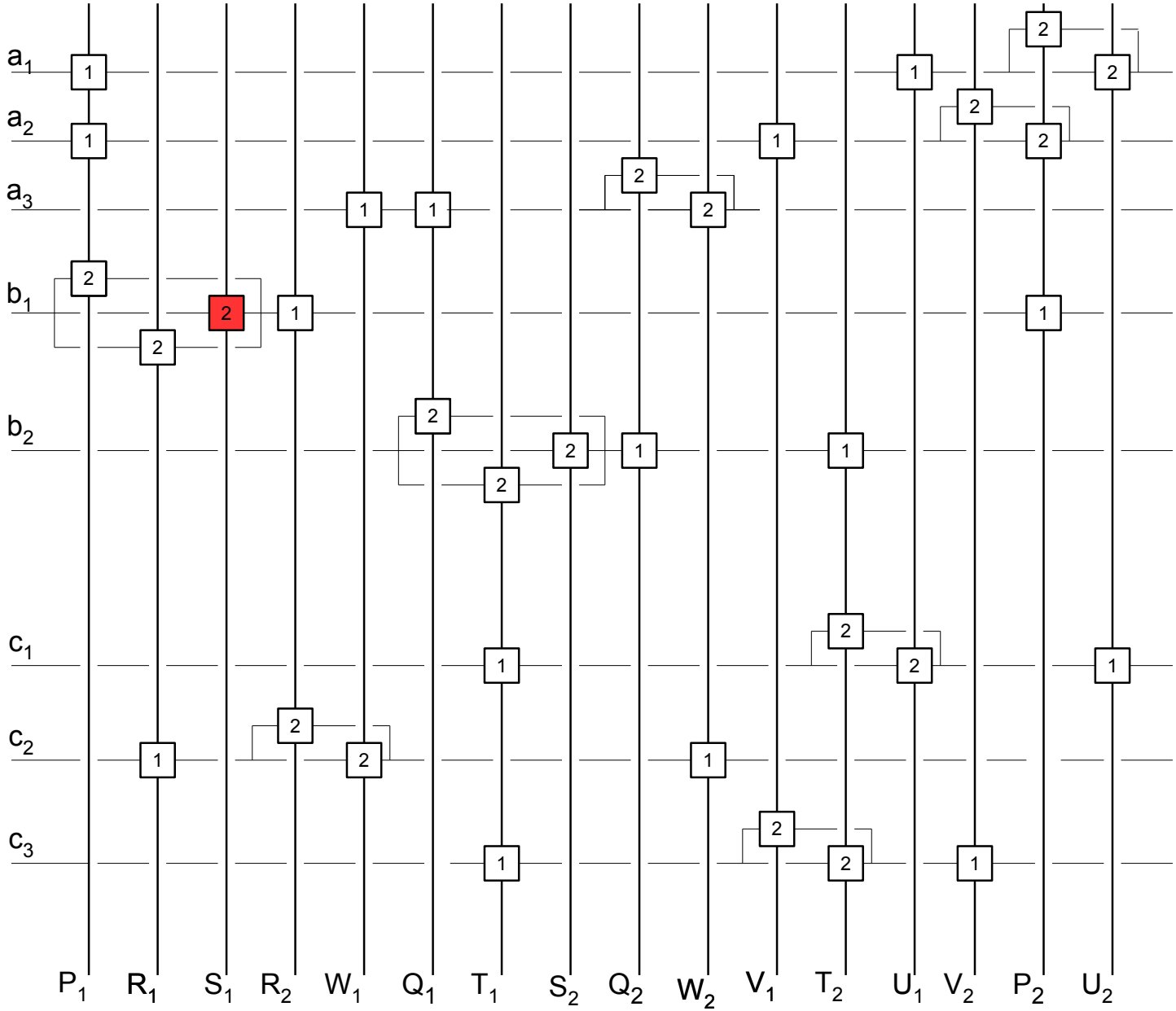
Introduce the guess

- Generally, no voltage in initial circuit lattice
- Assume E_i depends on x_j
- $\{a_1, \ldots, a_2\}$ solutions to E_i, where $x_j = 0$
- Add Type-2 switch to each $\{a_1, \ldots, a_2\}$, connect them
- Guessing $x_j = 0$ is inducing voltage in new circuit
- Similarly, guessing $x_j = 1$
- s-variable guess - $2s$ new vertical circuits

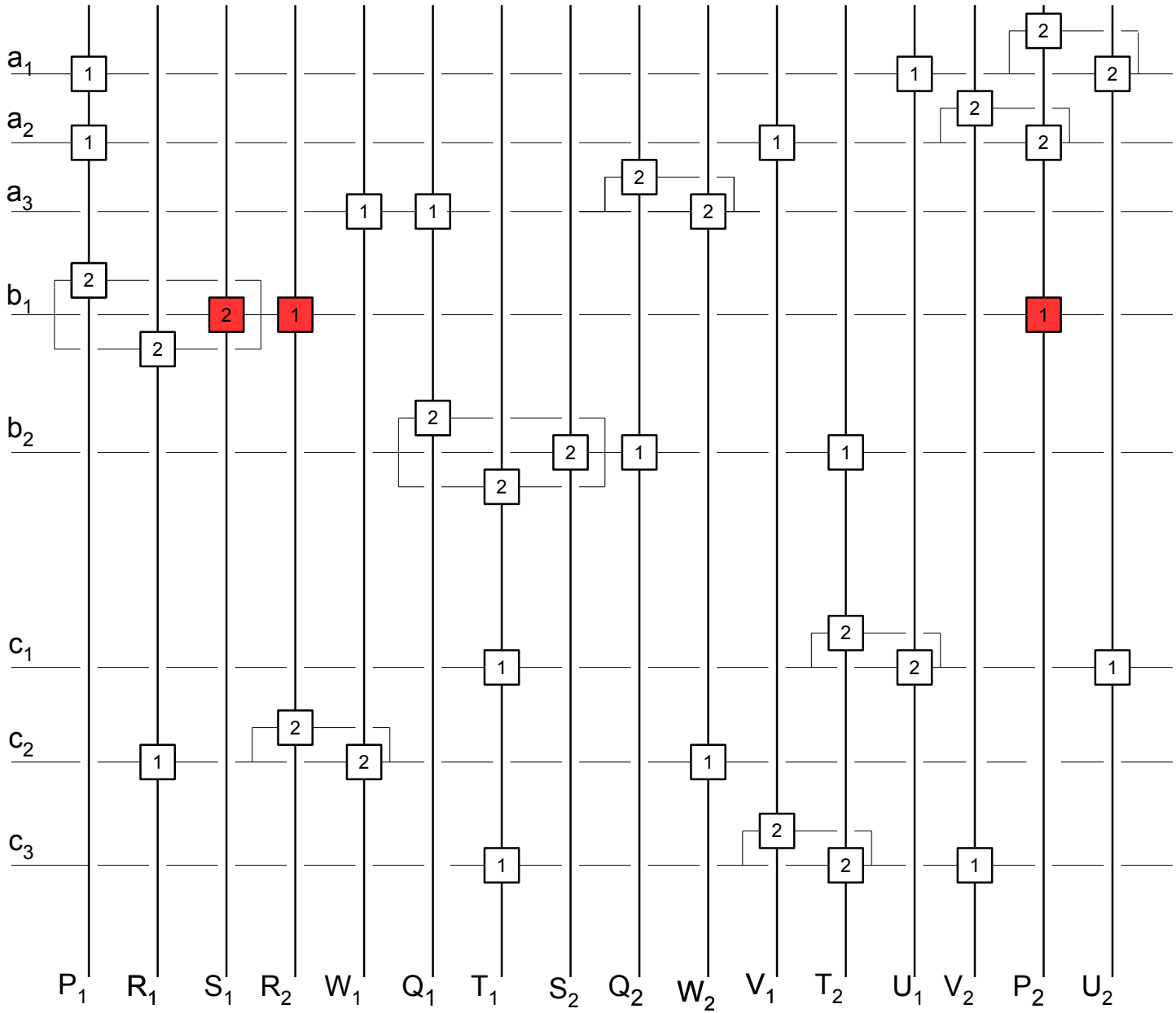
Exemplary circuit



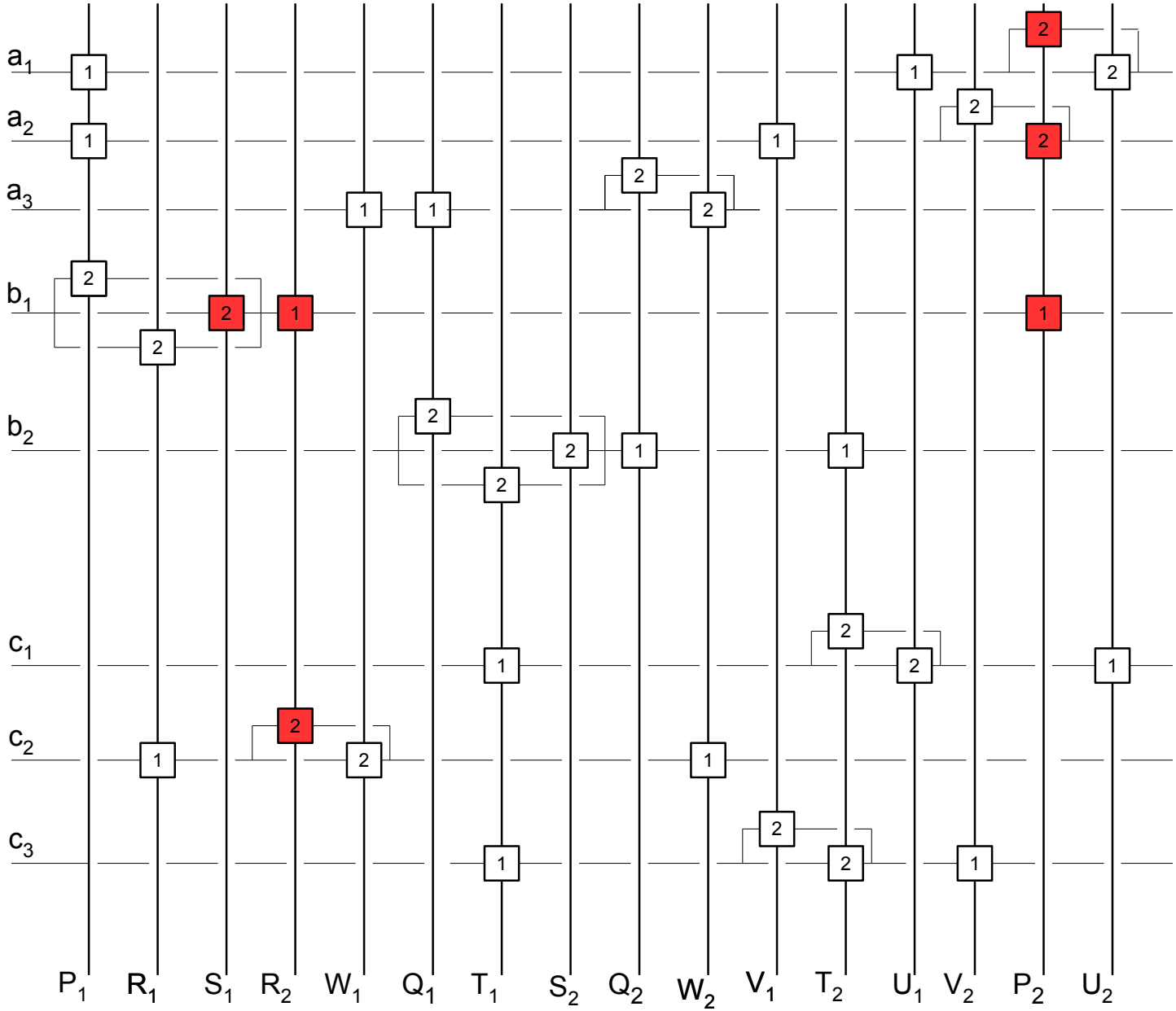
1st turn: introduce guess $x_4 = 0$



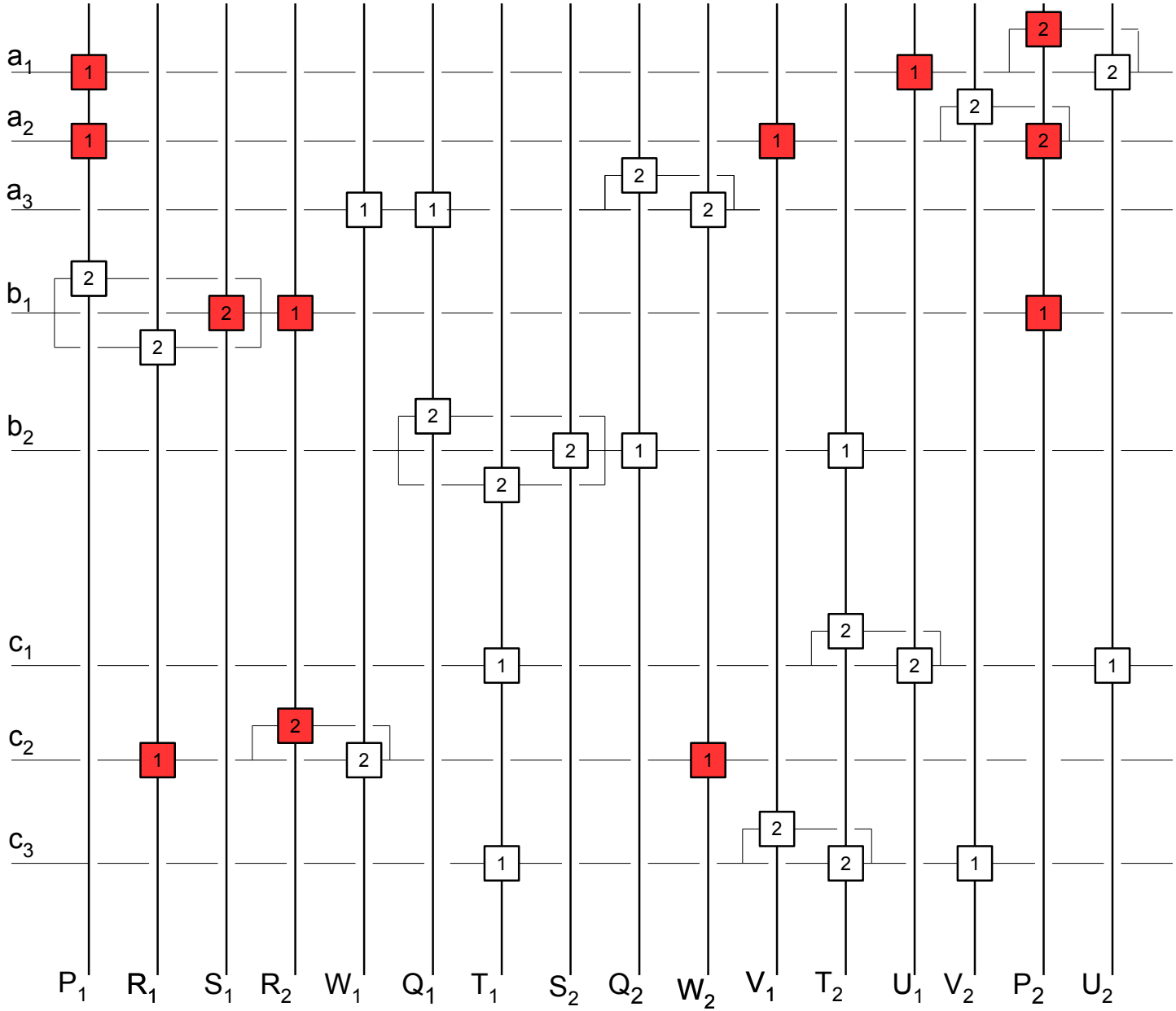
2nd turn



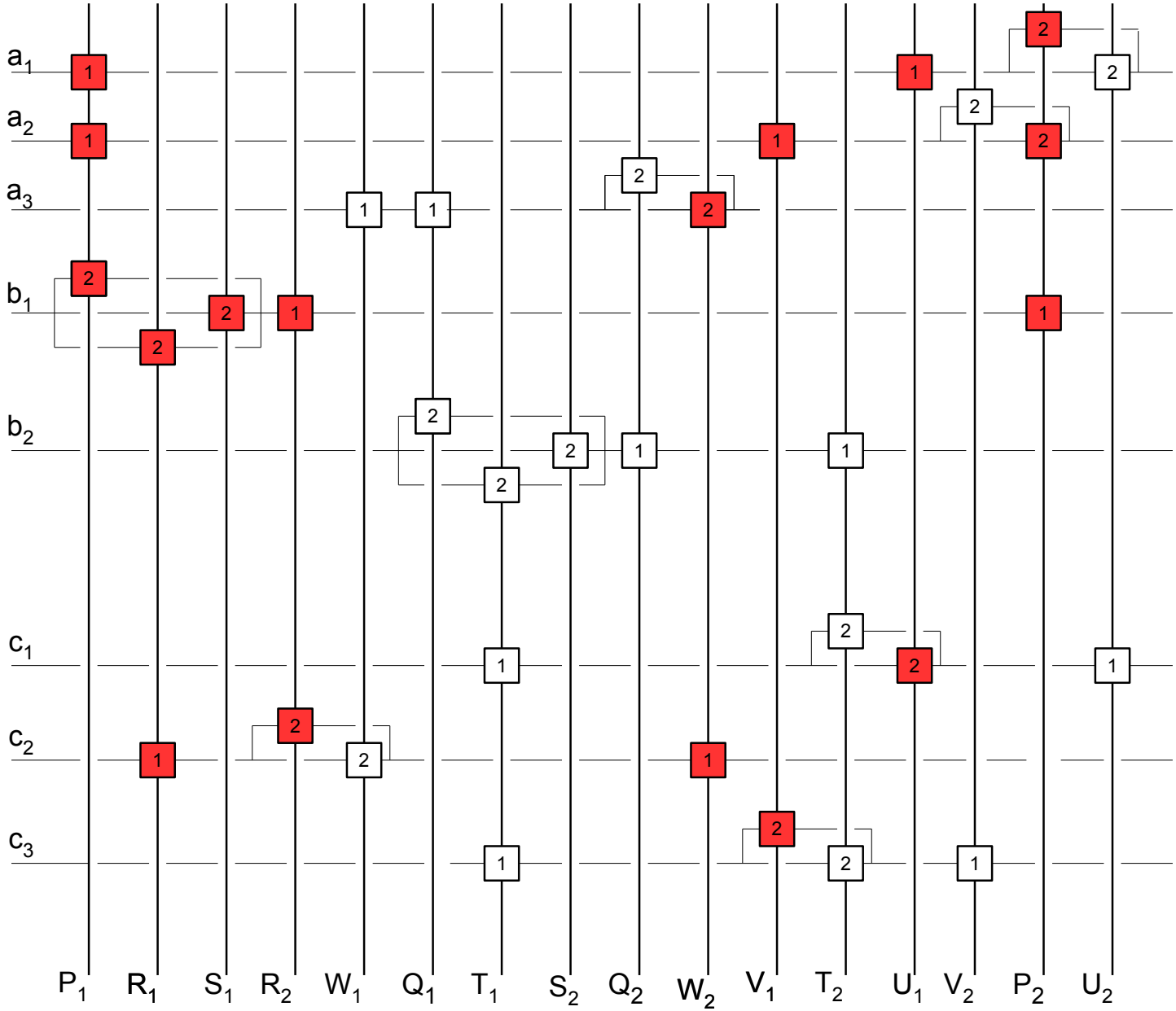
3rd turn



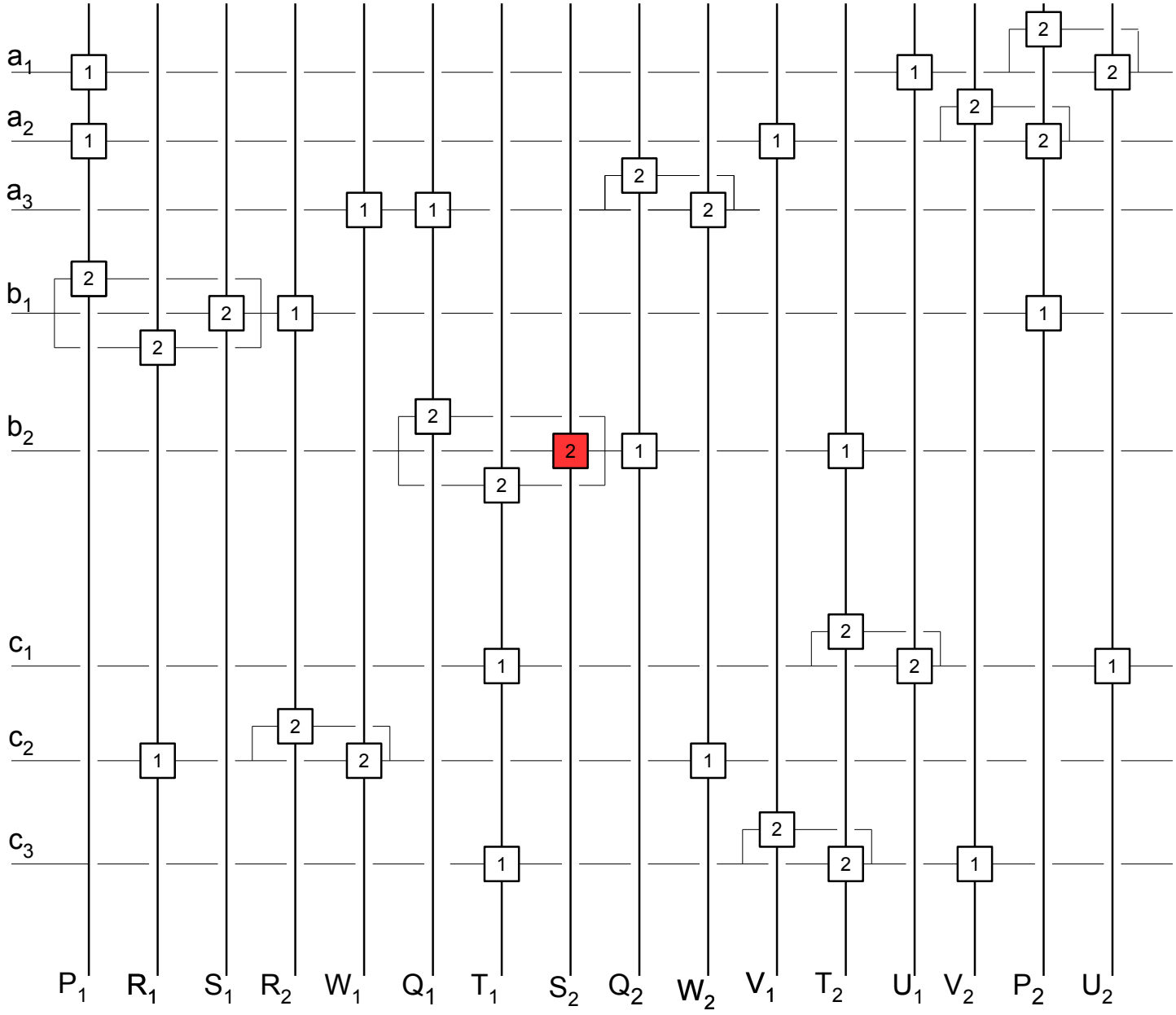
4th turn



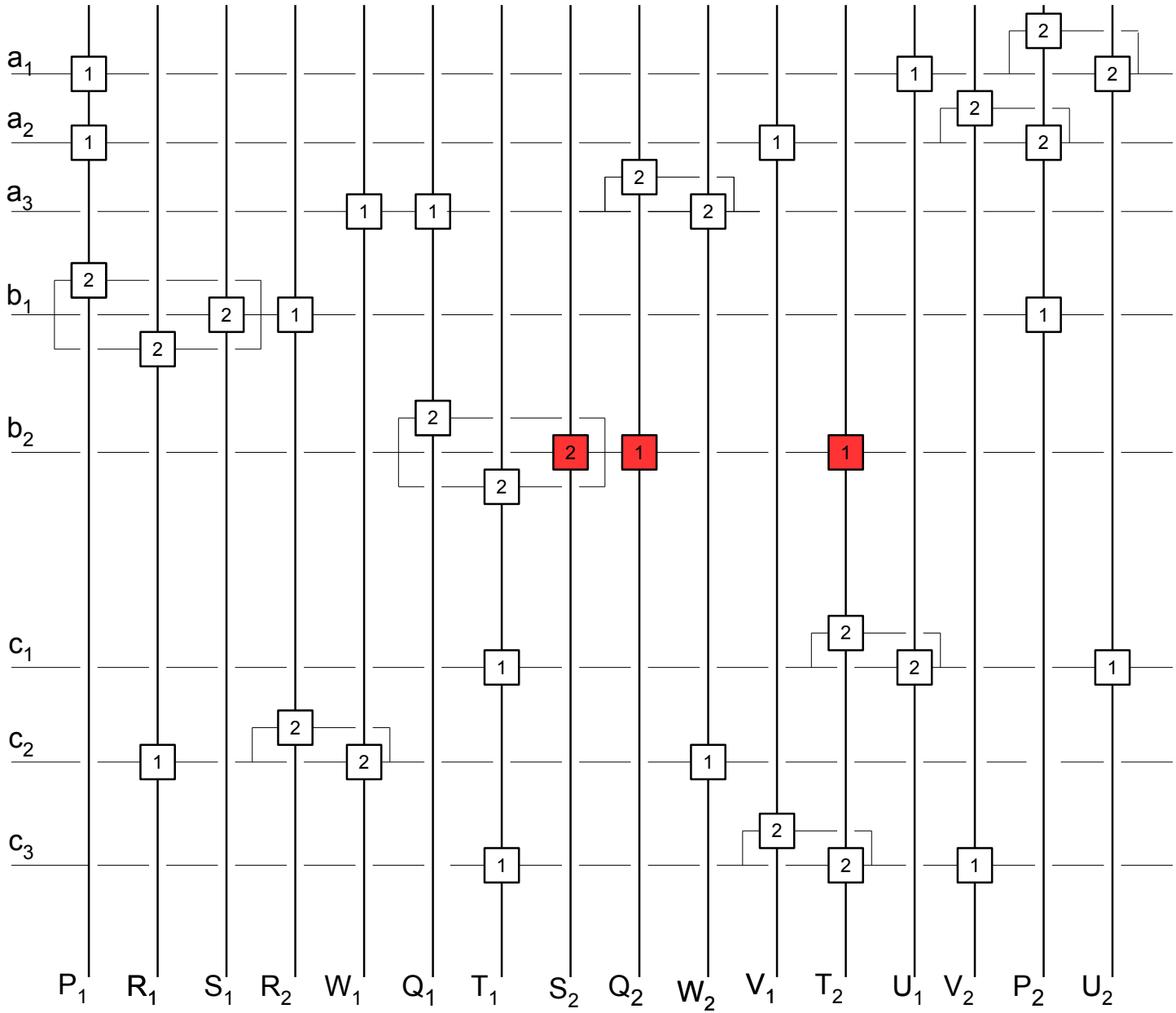
5th turn: Observe Inconsistence



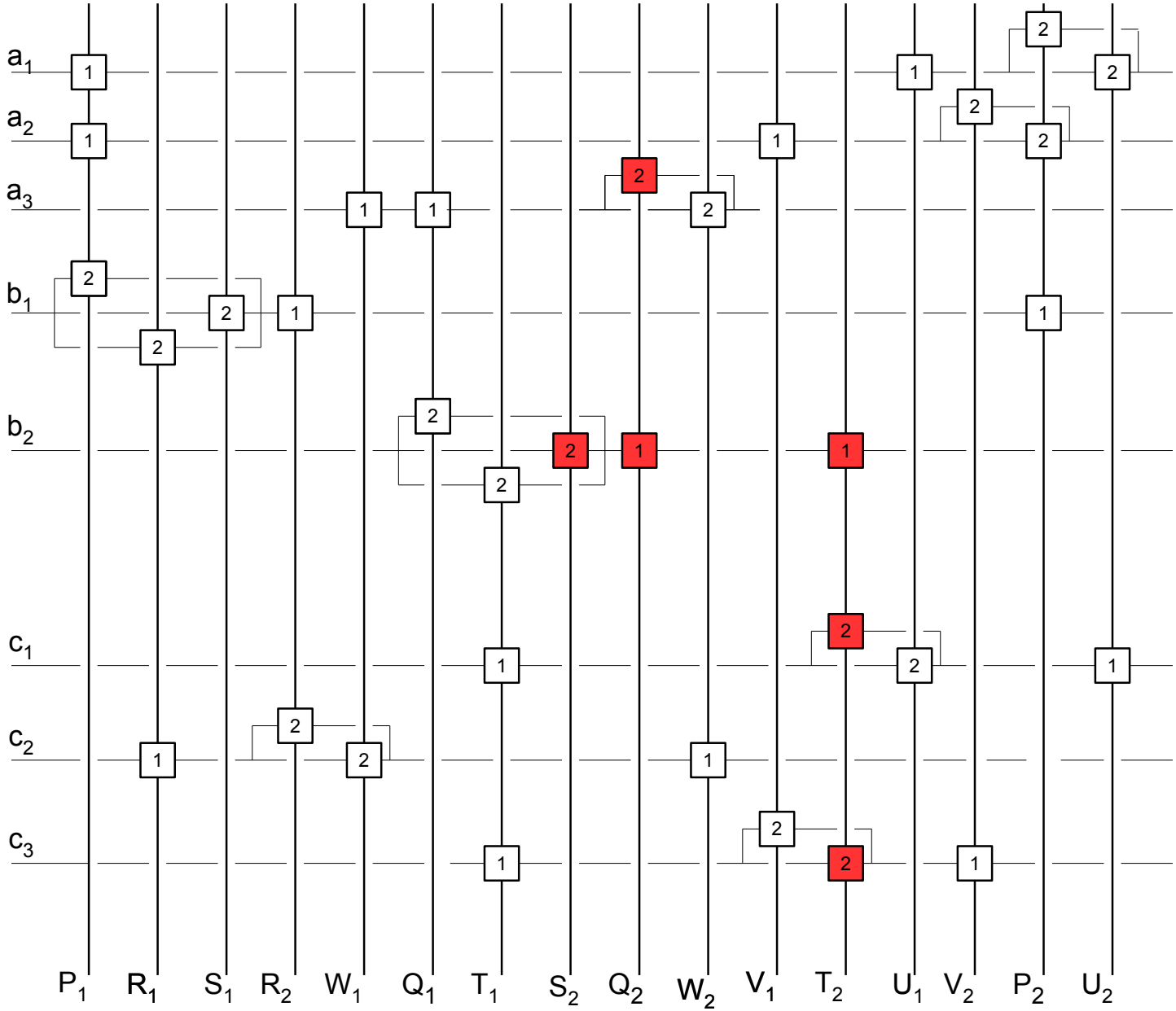
1st turn: introduce guess $x_4 = 1$



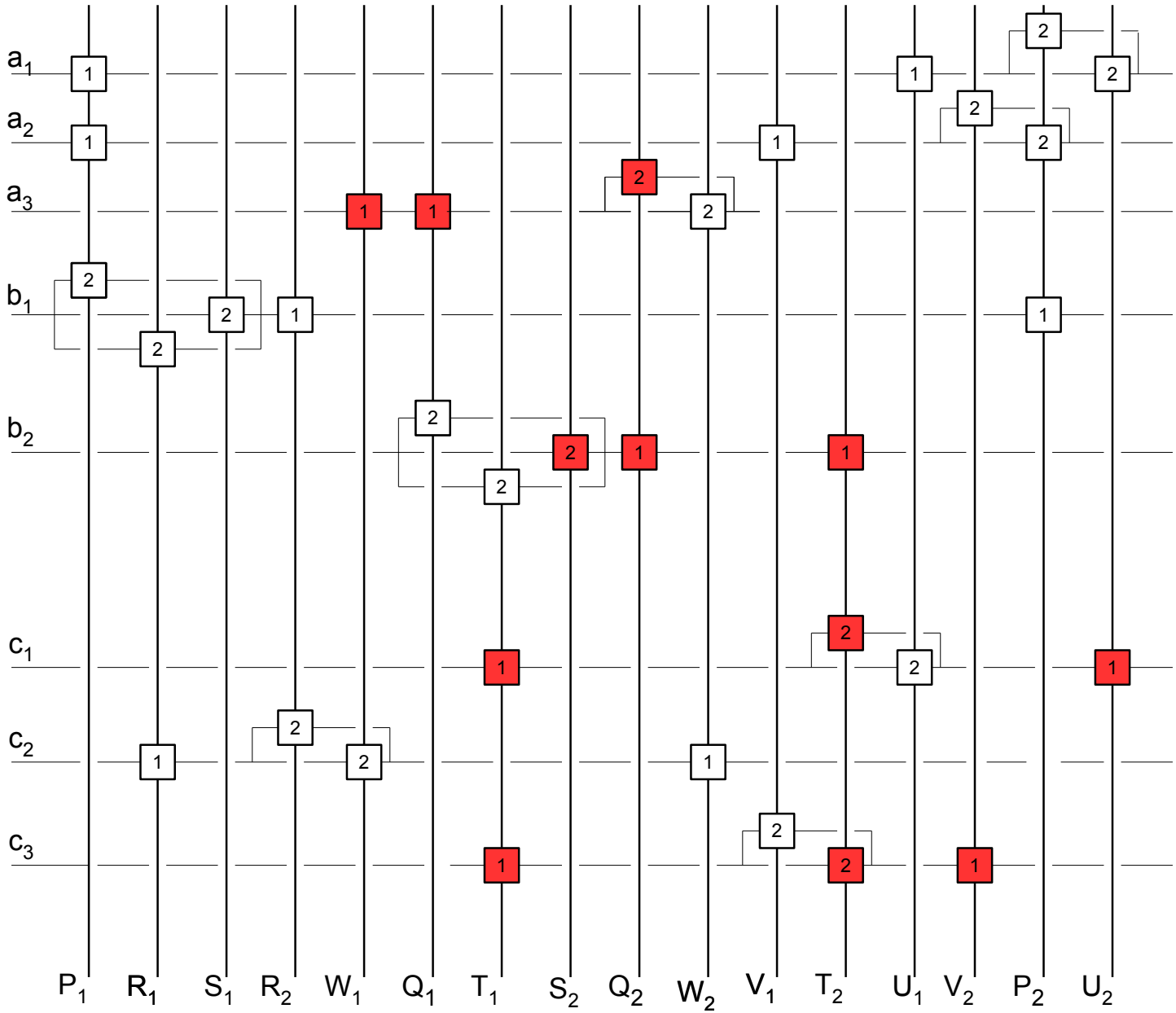
2nd turn



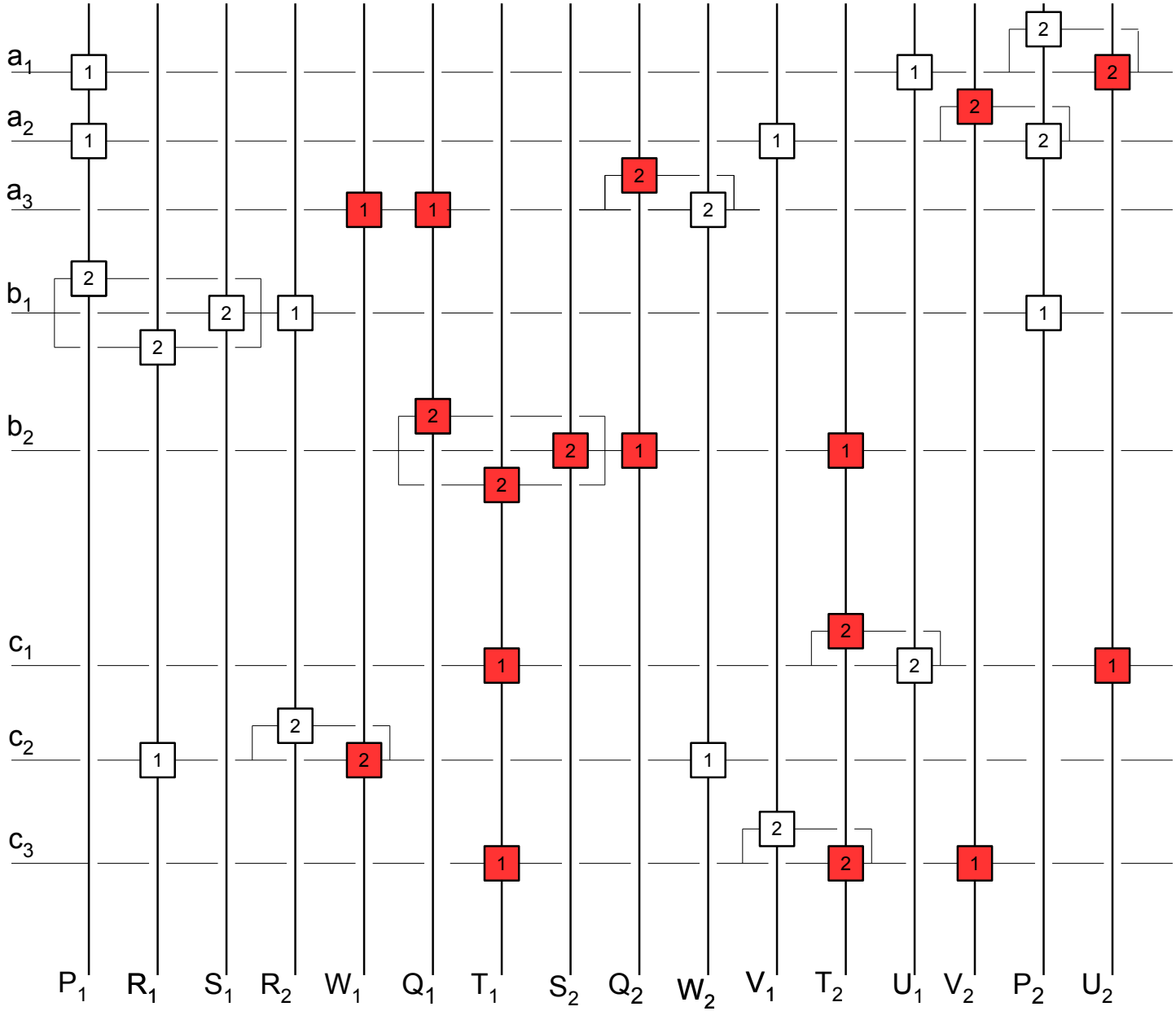
3rd turn



4th turn



5th turn: Observe Inconsistence



DES and TripleDES equations

- Variables:
- 64-bit plain-text, cipher-text,
- internal state blocks and 56(112)-bit key
- Equations from S-boxes as

$$Y_4 \oplus Z_4 = S(X_6 \oplus K_6)$$

- Each equation: 20 variables and 2^{16} RHS
- Study the system parameters

TripleDES system parameters

- 1712 variables
- 384 equations
- 16831 edges
- 3929 maximal edges
- 71320 tuples
- 1.1×10^9 switches

Implement on Modern Semiconductor Crystals?

- Transistor works as a switch
- 1.7×10^9 transistors on Dual-Core Itanium2 processor
- Circuit Lattice speed $\leq 2\times$(number of rounds) transistor turns
- 96 turns for TripleDES
- One transistor turn, say 100GHz(1000GHz reported)
- 1GHz key-rejecting rate when using for brute force
- Reported(2006) 0.13GHz with implementing encryption

Research Directions

- Reduce the number of switches:
- Not all tuples are relevant
- Use switches that control many circuits.
- Estimate the number of turns necessary for solving general systems

Conclusions

- Any Boolean problem with bounded number of variables is representable by MS linear equations
- New Gluing Algorithm is experimentally shown faster
- Using only maximal edges significantly economizes parameters
- Equation solving is shown as voltage expansion through a lattice of switches
- Our approach seems more flexible than implementing encryption as enables handling any equation system representing cipher
- Applications to DES, TripleDES are discussed