Practical pseudo-collisions for hash functions ARIRANG-224/384

Jian Guo[1], Krystian Matusiewicz[2], Lars R. Knudsen[2], San Ling[1], and Huaxiong Wang[1]

[1]School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore

[2]Department of Mathematics, Technical University of Denmark

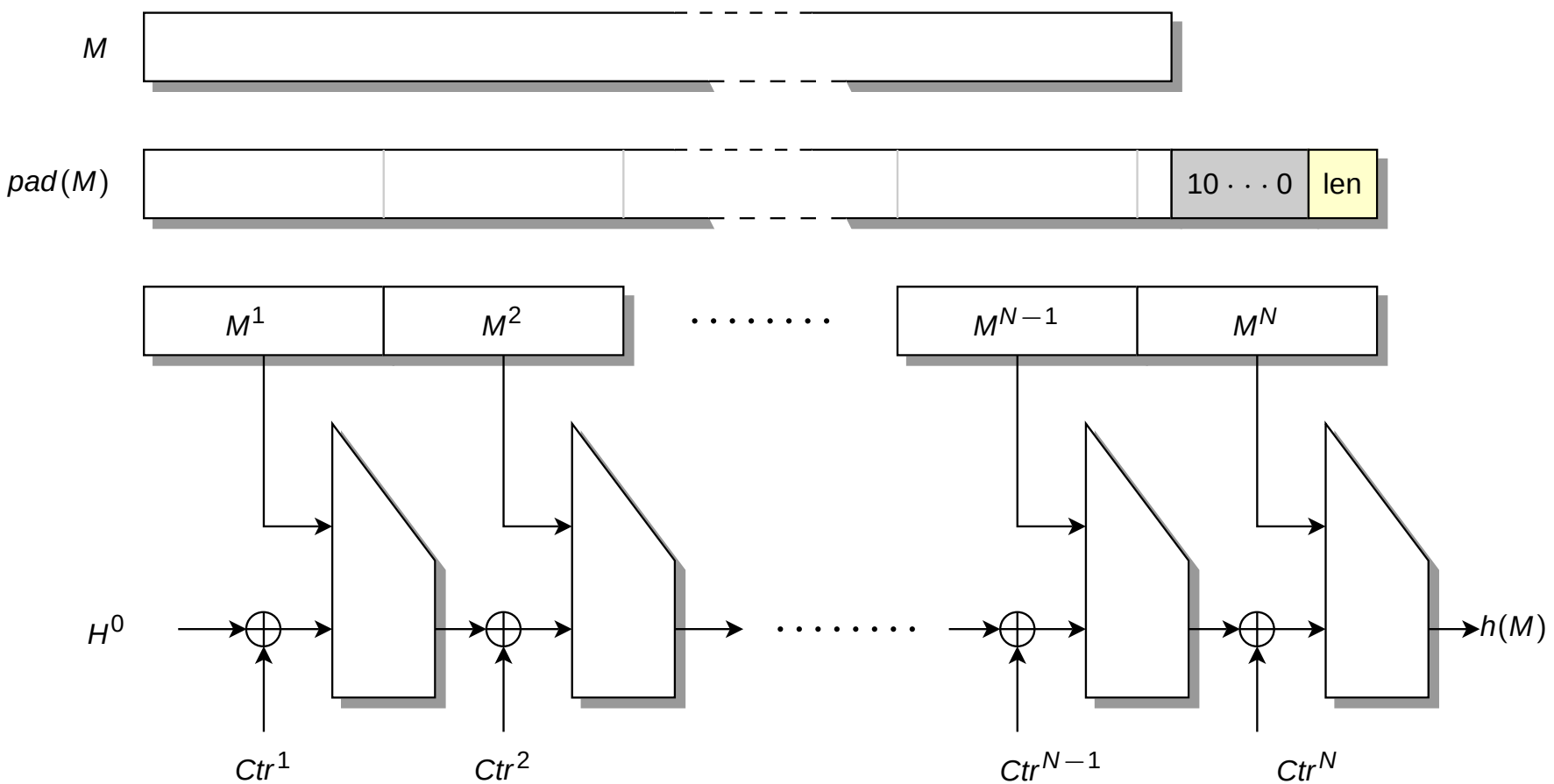
SAC 2009, August 13, 2009

SHA-3 candidate ARIRANG

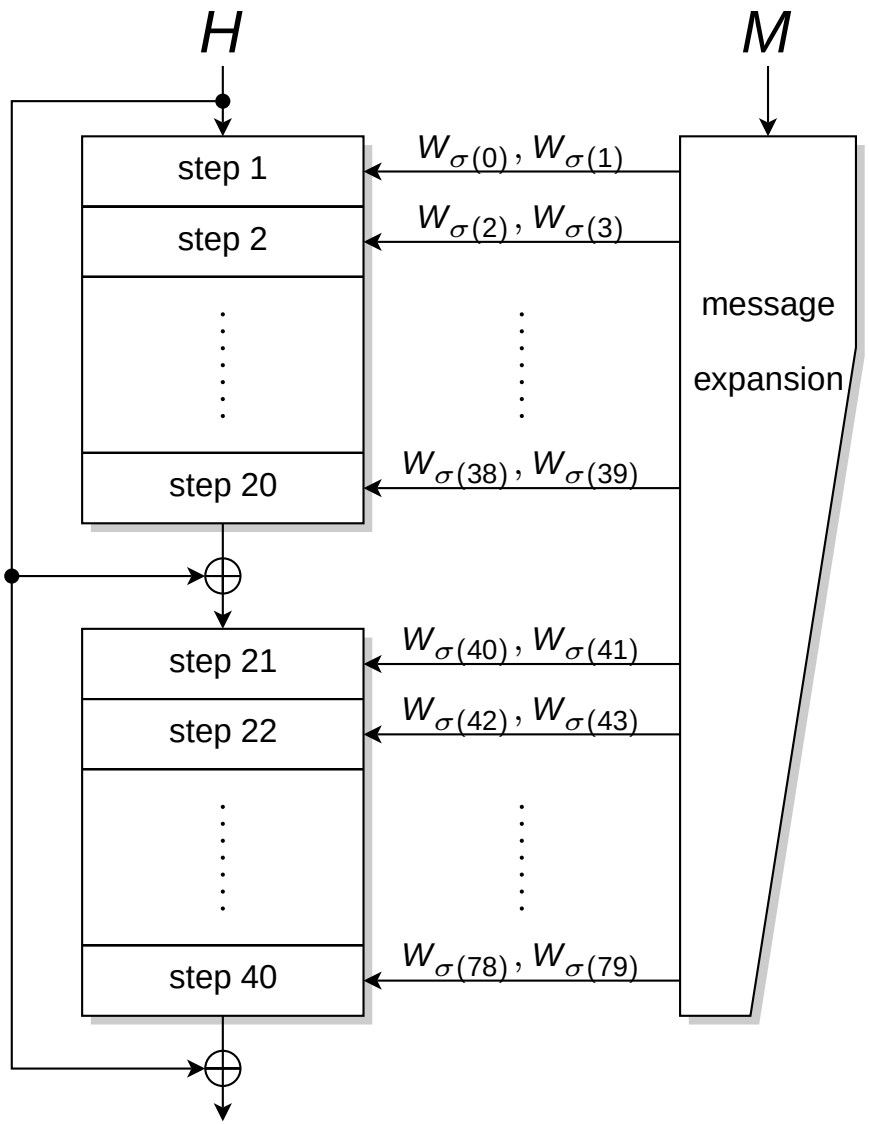
- One of the first round SHA-3 candidates
- Designed by a team from Center for Information Security Technologies (CIST), Korea University:

  Donghoon Chang, Seokhie Hong, Changheon Kang, Jinkeon Kang, Jongsung Kim, Changhoon Lee, Jesang Lee, Jongtae Lee, Sangjin Lee, Yuseop Lee, Jongin Lim, Jaechul Sung
- Design mixing parts from AES-based (S-box, MixColumn) and RAX designs (word rotations)
- Step function similar to an earlier design FORK-256

Hash function



Compression function



Message expansion

1. Generate 16 more words as linear combinations of $M_0, \ldots, M_{15}$
2. Pick (with repetitons) 80 words out of the 32 words obtained in the previous step

$$M_0, \ldots, M_{15} \longrightarrow$$

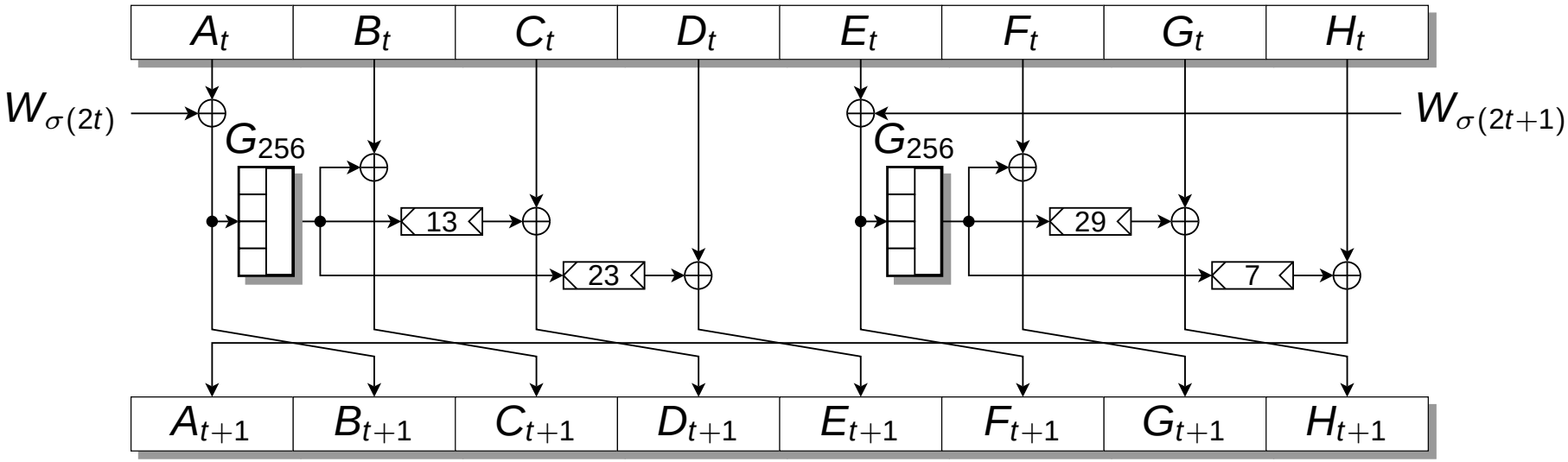
$$\begin{aligned}
W_{16} &\leftarrow (M_9 \oplus M_{11} \oplus M_{13} \oplus M_{15} \oplus K_0) \lll r_0 \\
W_{17} &\leftarrow (M_8 \oplus M_{10} \oplus M_{12} \oplus M_{14} \oplus K_1) \lll r_1 \\
W_{18} &\leftarrow (M_1 \oplus M_3 \oplus M_5 \oplus M_7 \oplus K_2) \lll r_2 \\
W_{19} &\leftarrow (M_0 \oplus M_2 \oplus M_4 \oplus M_6 \oplus K_3) \lll r_3 \\
W_{20} &\leftarrow (M_{14} \oplus M_4 \oplus M_{10} \oplus M_0 \oplus K_4) \lll r_0 \\
W_{21} &\leftarrow (M_{11} \oplus M_1 \oplus M_7 \oplus M_{13} \oplus K_5) \lll r_1 \\
W_{22} &\leftarrow (M_6 \oplus M_{12} \oplus M_2 \oplus M_8 \oplus K_6) \lll r_2 \\
W_{23} &\leftarrow (M_3 \oplus M_9 \oplus M_{15} \oplus M_5 \oplus K_7) \lll r_3 \\
W_{24} &\leftarrow (M_{13} \oplus M_{15} \oplus M_1 \oplus M_3 \oplus K_8) \lll r_0 \\
W_{25} &\leftarrow (M_4 \oplus M_6 \oplus M_8 \oplus M_{10} \oplus K_9) \lll r_1 \\
W_{26} &\leftarrow (M_5 \oplus M_7 \oplus M_9 \oplus M_{11} \oplus K_{10}) \lll r_2 \\
W_{27} &\leftarrow (M_{12} \oplus M_{14} \oplus M_0 \oplus M_2 \oplus K_{11}) \lll r_3 \\
W_{28} &\leftarrow (M_{10} \oplus M_0 \oplus M_6 \oplus M_{12} \oplus K_{12}) \lll r_0 \\
W_{29} &\leftarrow (M_{15} \oplus M_5 \oplus M_{11} \oplus M_1 \oplus K_{13}) \lll r_1 \\
W_{30} &\leftarrow (M_2 \oplus M_8 \oplus M_{14} \oplus M_4 \oplus K_{14}) \lll r_2 \\
W_{31} &\leftarrow (M_7 \oplus M_{13} \oplus M_3 \oplus M_9 \oplus K_{15}) \lll r_3
\end{aligned}$$

$\longrightarrow$

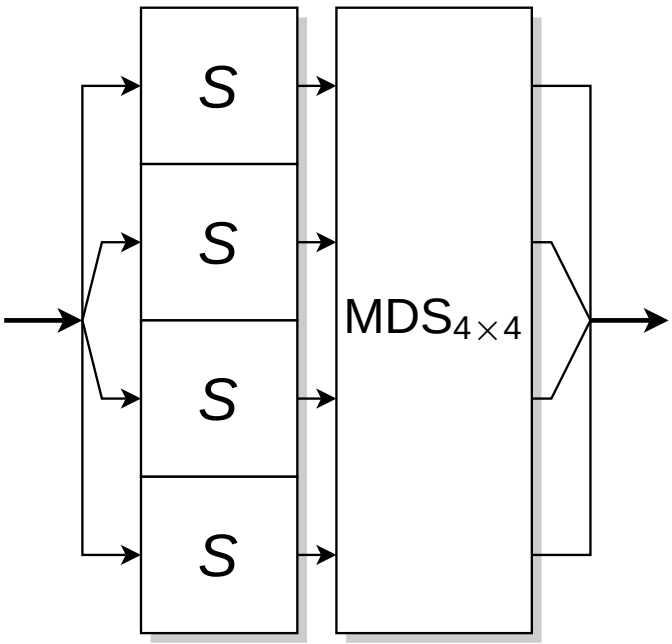
$\sigma(i)$	$\sigma(i)$
16, 17	24, 25
0, 1	12, 5
2, 3	14, 7
4, 5	0, 9
6, 7	2, 11
18, 19	26, 27
8, 9	4, 13
10, 11	6, 15
12, 13	8, 1
14, 15	10, 3
20,21	28, 29
3, 6	7, 2
9,12	13, 8
15, 2	3, 14
5, 8	9, 4
22,23	30, 31
11,14	15, 10
1, 4	5, 0
7,10	11, 6
13, 0	1, 12

Step transformation

- transforms 8 32-bit words of the state and 8 words of the expanded message to new state
- uses 32-bit rotations, XORs and a 32×32 bit function G_{256}
- only non-linear (over $\mathbb{F}_2$) part is G_{256}



Function G_{256}



- 32×32 composite “megabox”:
  - 4 bytewise AES S-boxes
  - Followed by $MDS_{4\times4}$ transformation (AES MixColumn)

ARIRANG-512 uses a similar function G_{512} defined on 8 32-bit words and using $MDS_{8\times8}$.

Basic observations

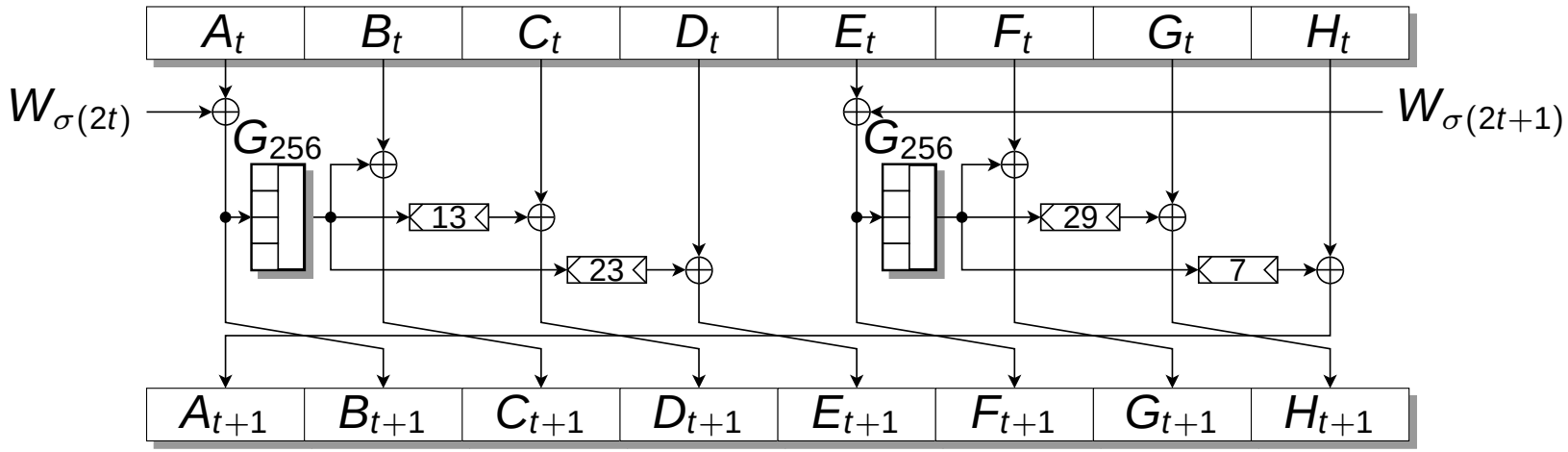
- $MDS_{4\times4}$ has fixed points of the form (a, a, a, a)

$$MDS_{4\times4} = \begin{bmatrix} z & z+1 & 1 & 1 \\ 1 & z & z+1 & 1 \\ 1 & 1 & z & z+1 \\ z+1 & 1 & 1 & z \end{bmatrix}$$

- S-box differential $\texttt{0xff} \rightarrow \texttt{0xff}$ is possible with prob. 2^{-7}.
- Differential $\texttt{0xffffffff} \rightarrow \texttt{0xffffffff}$ for G_{256} has probability 2^{-28}
- 512-bit variant: no fixed points for MDS, but still can get all-ones to all-ones differences

All-one differences

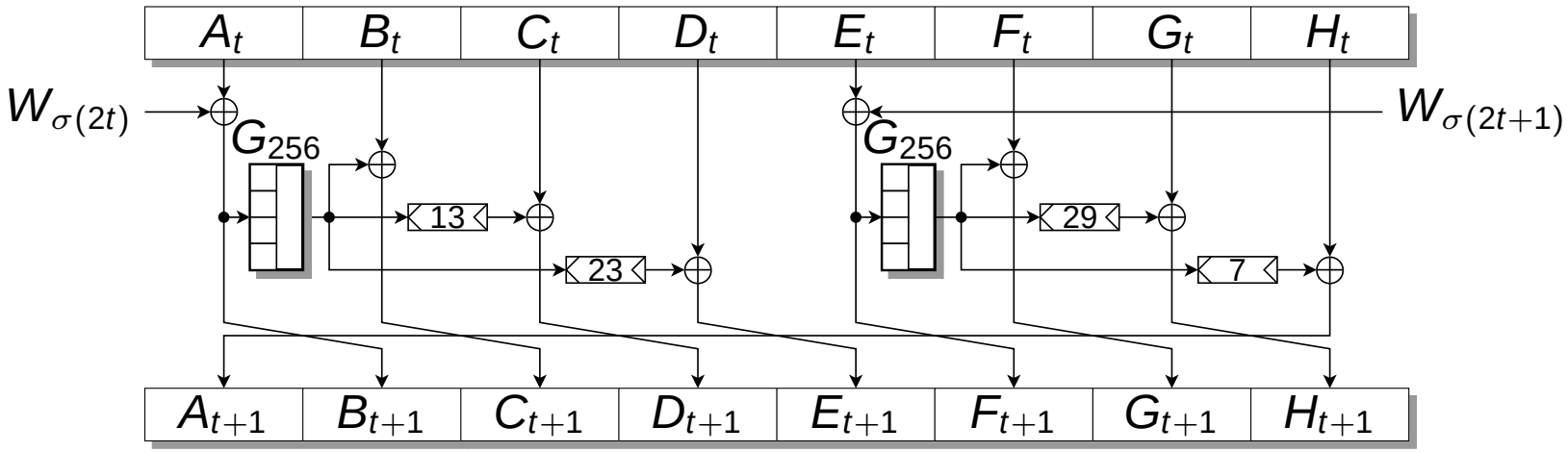
If we consider only all-one differences:



All-one differences

If we consider only all-one differences:

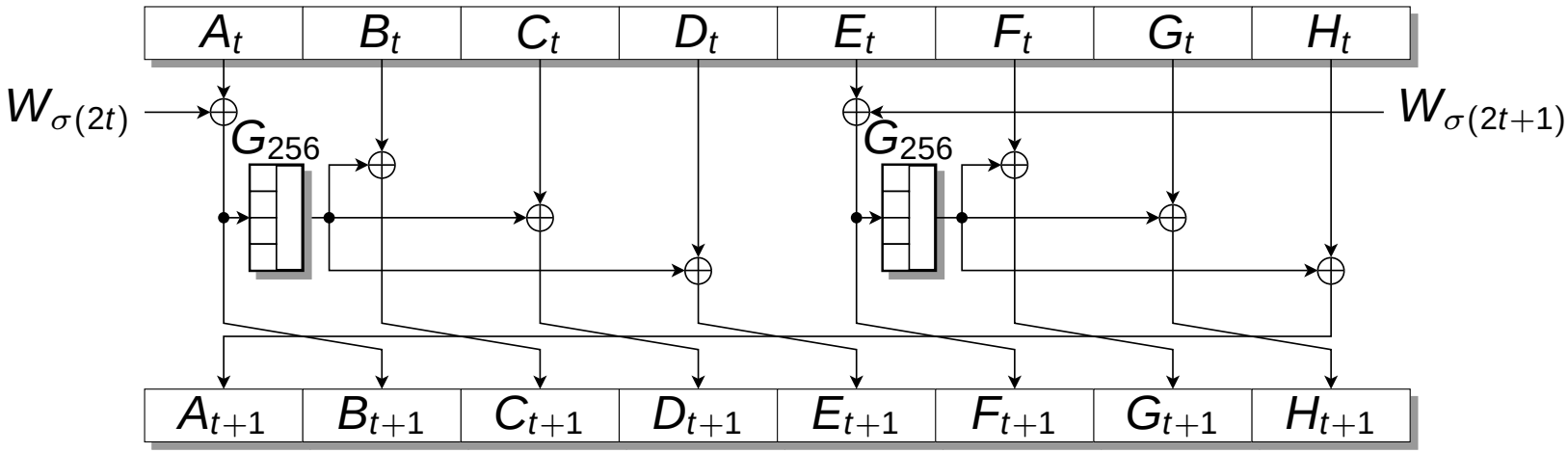
- rotations in step function do not play any role



All-one differences

If we consider only all-one differences:

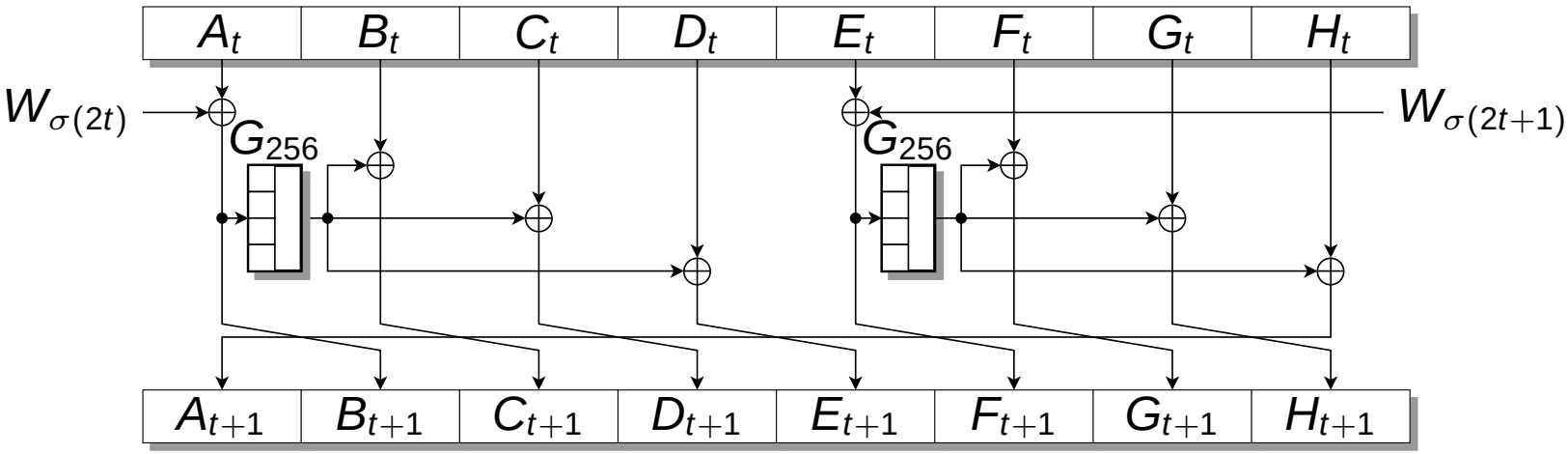
- rotations in step function do not play any role



All-one differences

If we consider only all-one differences:

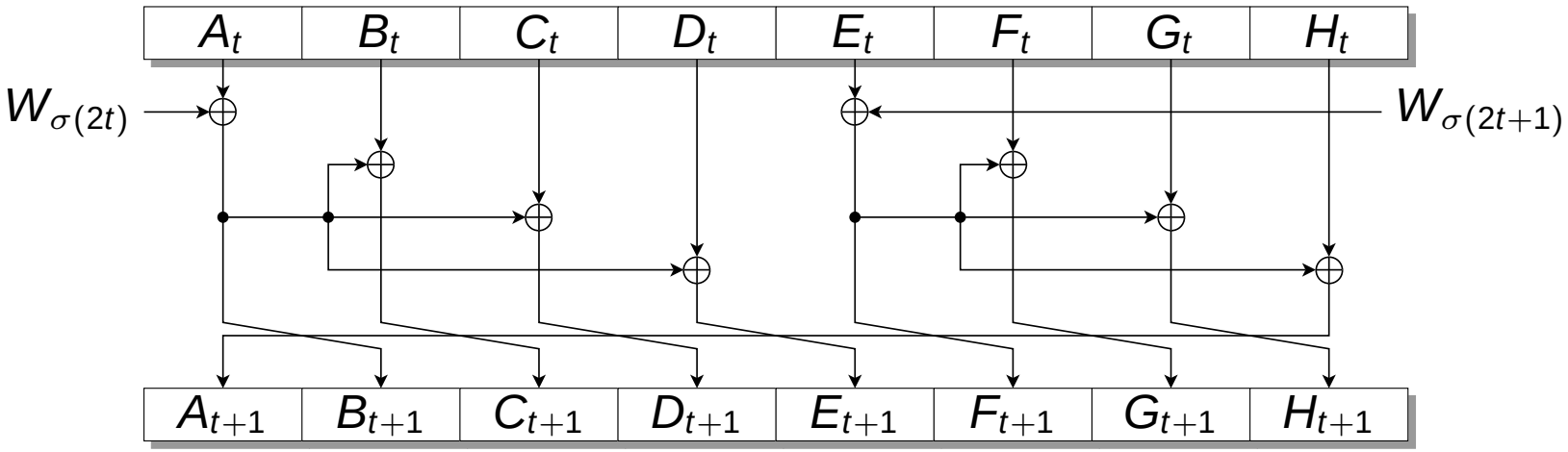
- rotations in step function do not play any role
- we can replace G_{256} with identity (with prob. 2^{-28})



All-one differences

If we consider only all-one differences:

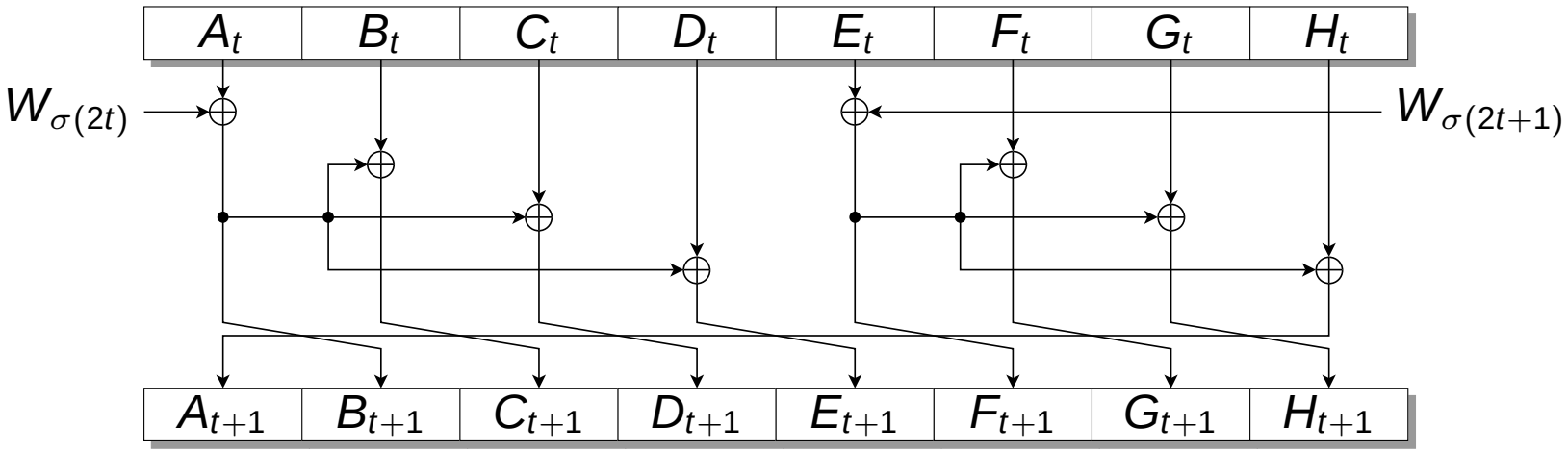
- rotations in step function do not play any role
- we can replace G_{256} with identity (with prob. 2^{-28})



All-one differences

If we consider only all-one differences:

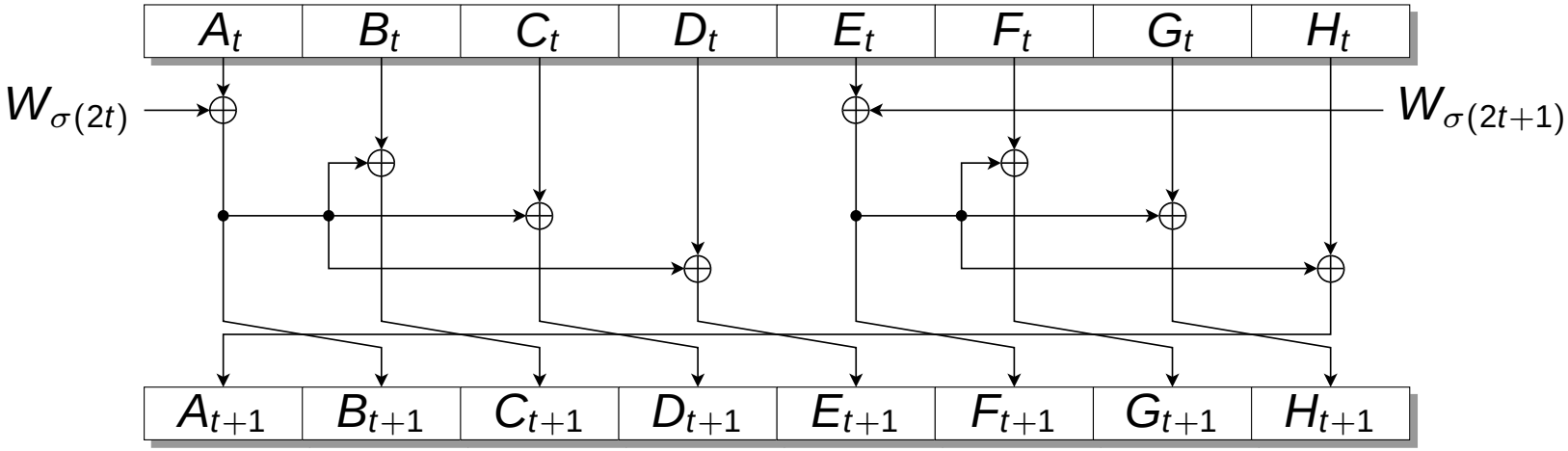
- rotations in step function do not play any role
- we can replace G_{256} with identity (with prob. 2^{-28})
- One register can be represented as a single bit



All-one differences

If we consider only all-one differences:

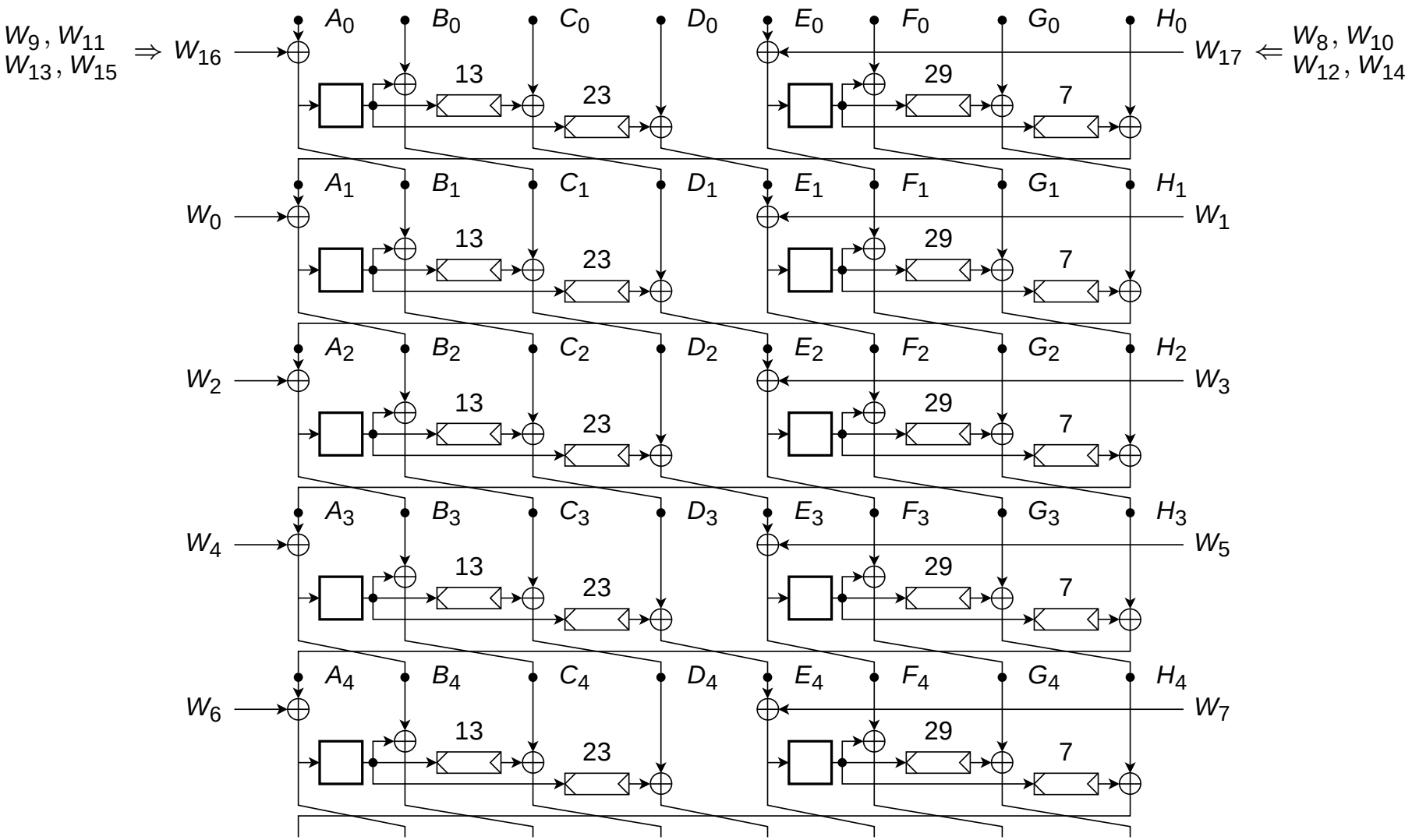
- rotations in step function do not play any role
- we can replace G_{256} with identity (with prob. 2^{-28})
- One register can be represented as a single bit
- Linearized model has $8 + 16$ variables: we have 2^{24} paths

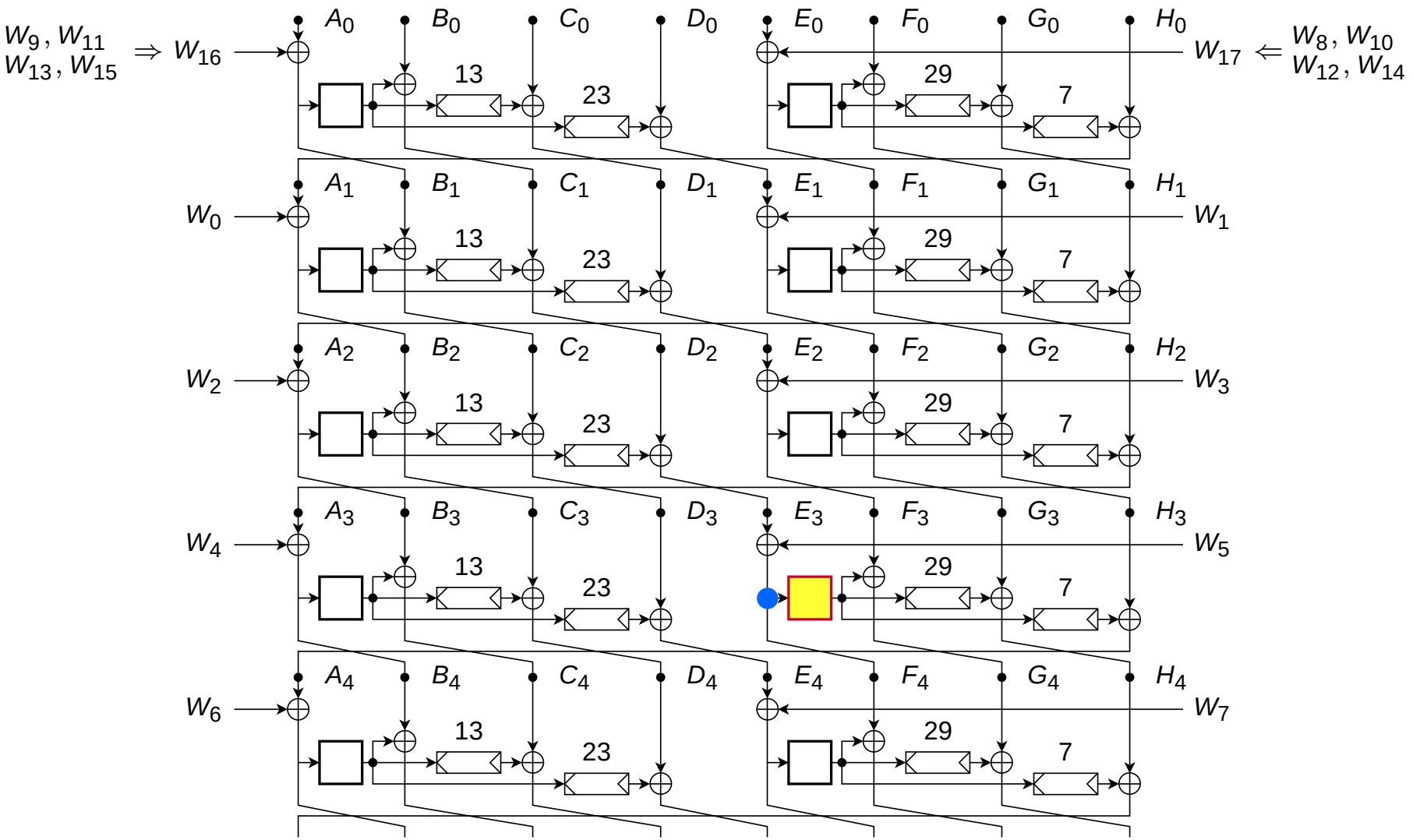


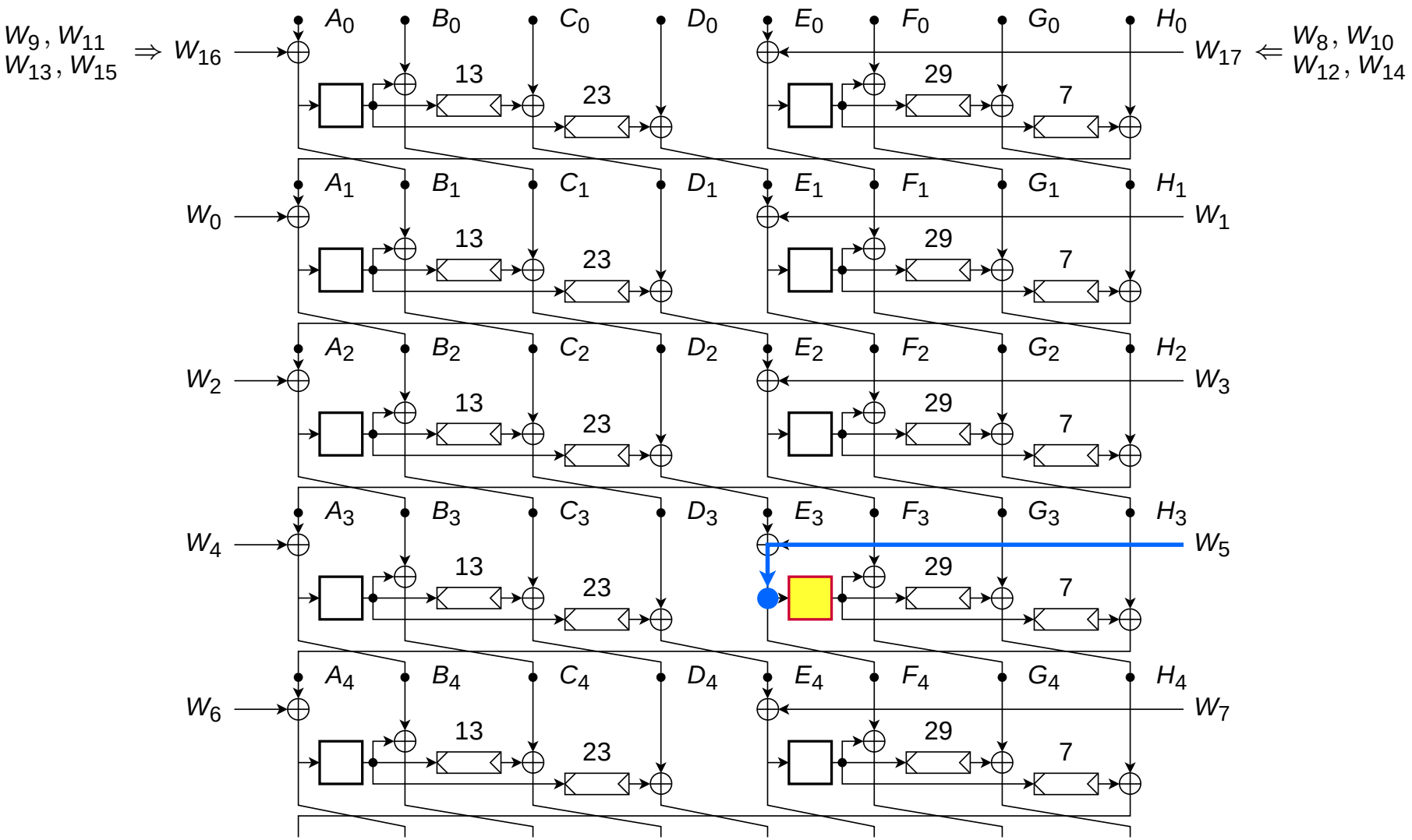
Satisfying conditions

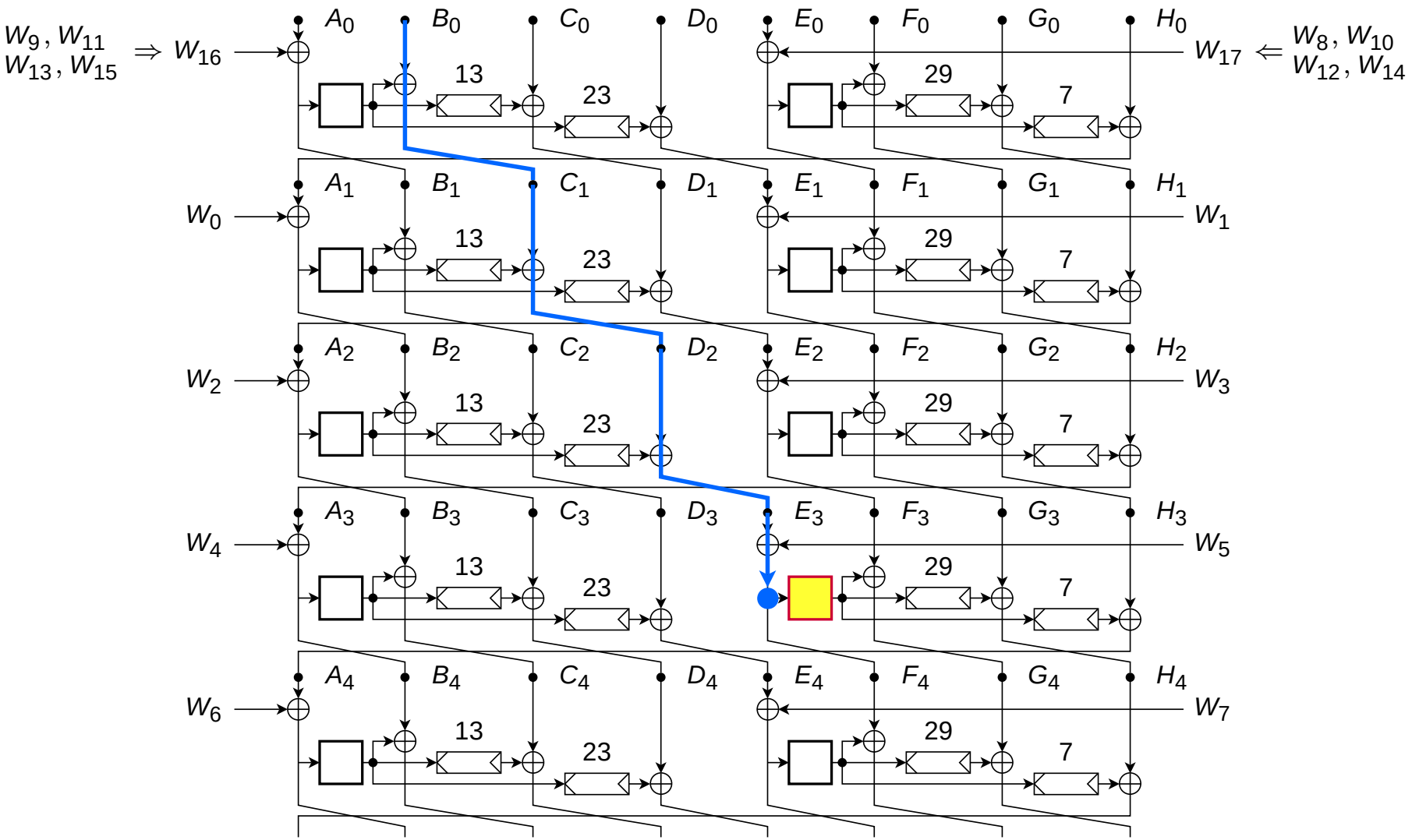
To eliminate probabilistic behaviour, we want to set inputs of active G_{256} to "good" values.

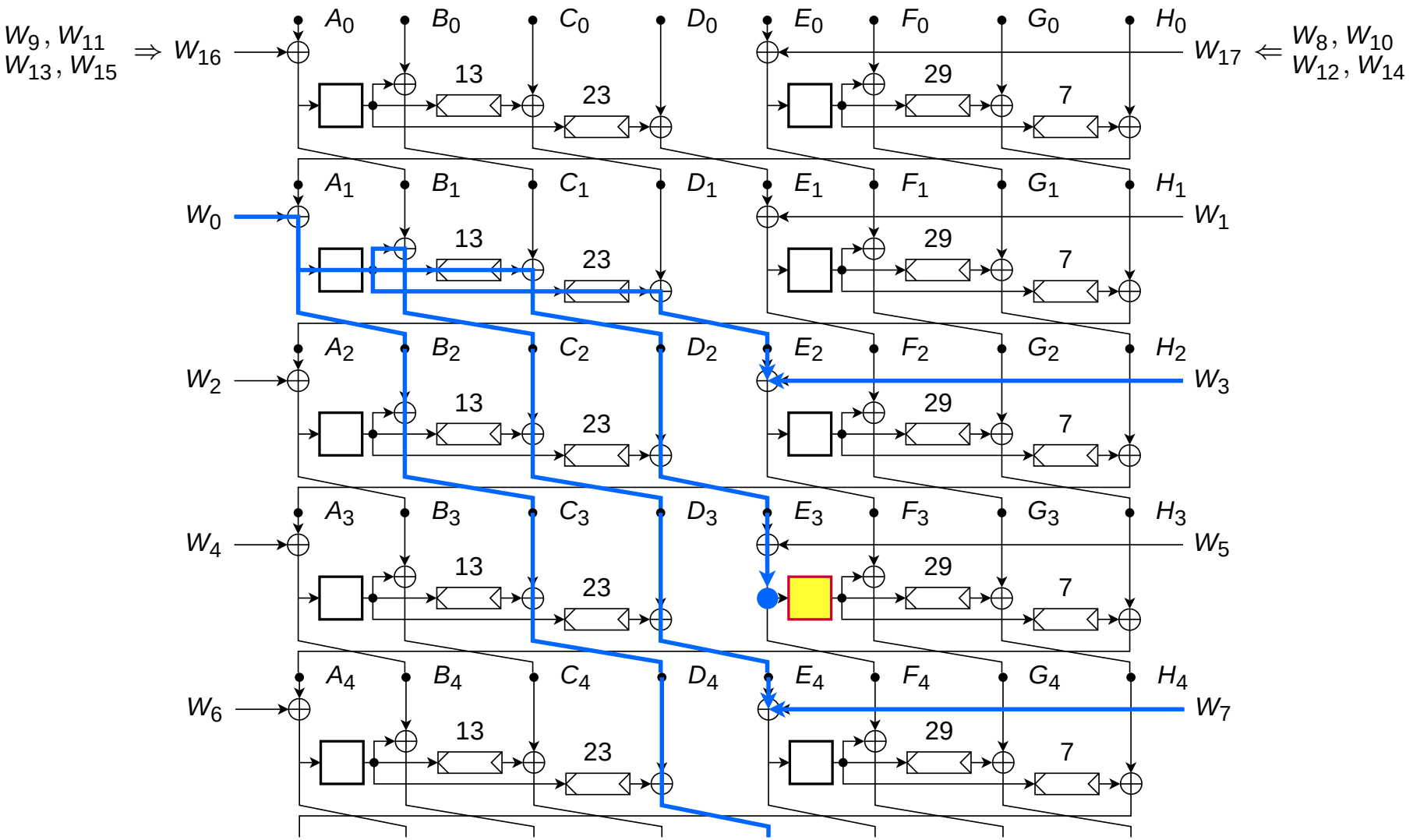
- We have full control over words $W_0, \ldots, W_{15}$
- Through linear combinations, we have some control over words $W_{16}, \ldots, W_{31}$
- For semi-free-start collisions and pseudo-collisions, we additionally have control over initial values $IV_0, \ldots, IV_7$


W_9, W_{11}
W_{13}, W_{15}
$\Rightarrow W_{16}$
W_8, W_{10}
W_{12}, W_{14}
$\Leftarrow W_{17}$
A_0 B_0 C_0 D_0 E_0 F_0 G_0 H_0
A_1 B_1 C_1 D_1 E_1 F_1 G_1 H_1
A_2 B_2 C_2 D_2 E_2 F_2 G_2 H_2
A_3 B_3 C_3 D_3 E_3 F_3 G_3 H_3
A_4 B_4 C_4 D_4 E_4 F_4 G_4 H_4
13
23
29
7
W_0
W_1
W_2
W_3
W_4
W_5
W_6
W_7


W_9, W_{11}
W_{13}, W_{15}
$\Rightarrow W_{16}$
A_0 B_0 C_0 D_0 E_0 F_0 G_0 H_0
$W_{17} \Leftarrow W_8, W_{10}$
W_{12}, W_{14}
13
23
29
7
W_0
A_1 B_1 C_1 D_1 E_1 F_1 G_1 H_1
W_1
W_2
A_2 B_2 C_2 D_2 E_2 F_2 G_2 H_2
W_3
W_4
A_3 B_3 C_3 D_3 E_3 F_3 G_3 H_3
W_5
W_6
A_4 B_4 C_4 D_4 E_4 F_4 G_4 H_4
W_7


W_9, W_{11}
W_{13}, W_{15}
$\Rightarrow W_{16}$
W_8, W_{10}
W_{12}, W_{14}
$\Leftarrow W_{17}$
W_0
W_1
W_2
W_3
W_4
W_5
W_6
W_7
A_0 B_0 C_0 D_0 E_0 F_0 G_0 H_0
A_1 B_1 C_1 D_1 E_1 F_1 G_1 H_1
A_2 B_2 C_2 D_2 E_2 F_2 G_2 H_2
A_3 B_3 C_3 D_3 E_3 F_3 G_3 H_3
A_4 B_4 C_4 D_4 E_4 F_4 G_4 H_4
13
23
29
7

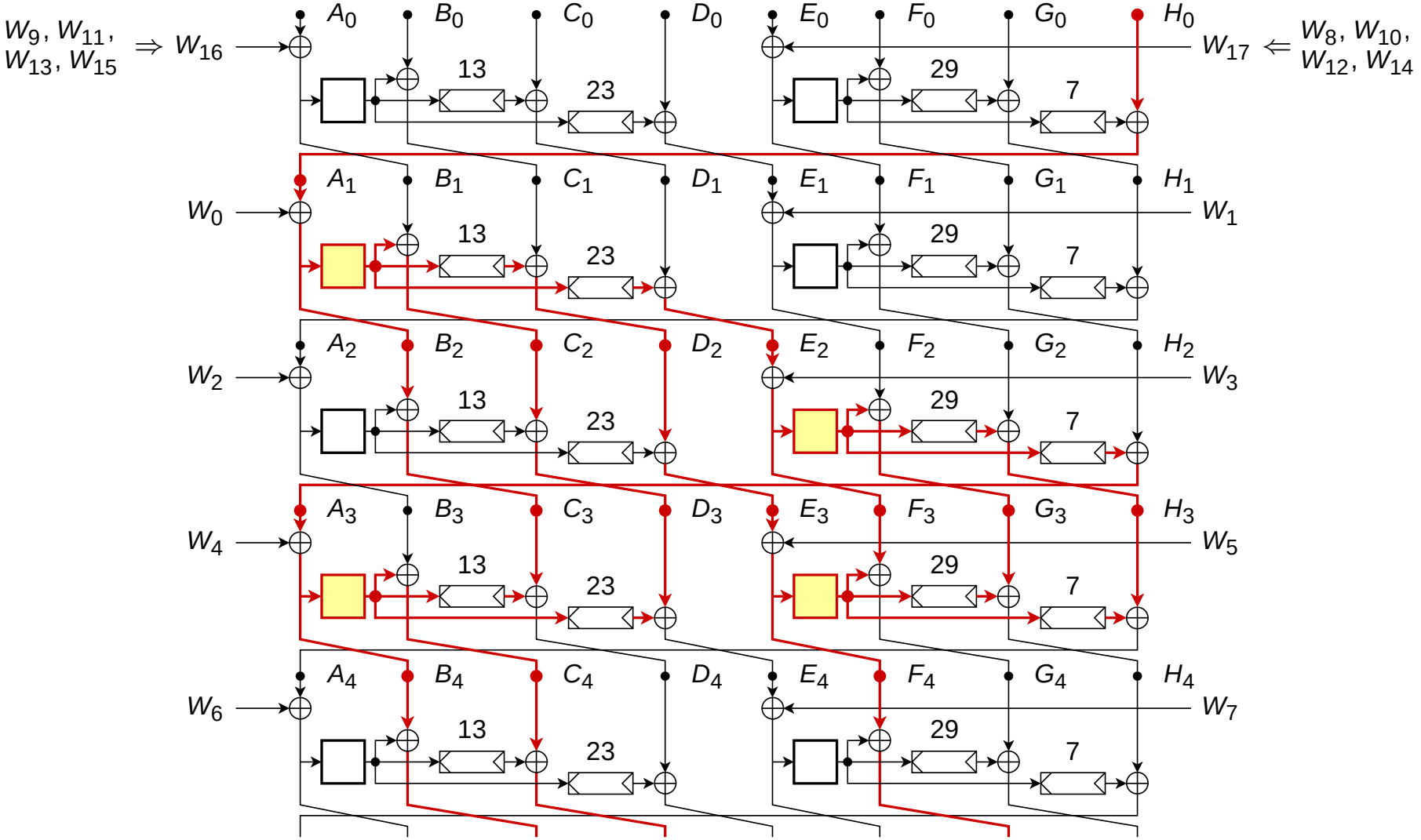

W_9, W_{11}
W_{13}, W_{15}
$\Rightarrow W_{16}$
A_0 B_0 C_0 D_0 E_0 F_0 G_0 H_0
$W_{17} \Leftarrow W_8, W_{10}$
W_{12}, W_{14}
13
23
29
7
W_0
A_1 B_1 C_1 D_1 E_1 F_1 G_1 H_1
W_1
W_2
A_2 B_2 C_2 D_2 E_2 F_2 G_2 H_2
W_3
W_4
A_3 B_3 C_3 D_3 E_3 F_3 G_3 H_3
W_5
W_6
A_4 B_4 C_4 D_4 E_4 F_4 G_4 H_4
W_7


W_9, W_{11}
W_{13}, W_{15}
$\Rightarrow W_{16}$
W_8, W_{10}
W_{12}, W_{14}
$\Leftarrow W_{17}$
A_0 B_0 C_0 D_0 E_0 F_0 G_0 H_0
A_1 B_1 C_1 D_1 E_1 F_1 G_1 H_1
A_2 B_2 C_2 D_2 E_2 F_2 G_2 H_2
A_3 B_3 C_3 D_3 E_3 F_3 G_3 H_3
A_4 B_4 C_4 D_4 E_4 F_4 G_4 H_4
13
23
29
7
W_0
W_1
W_2
W_3
W_4
W_5
W_6
W_7

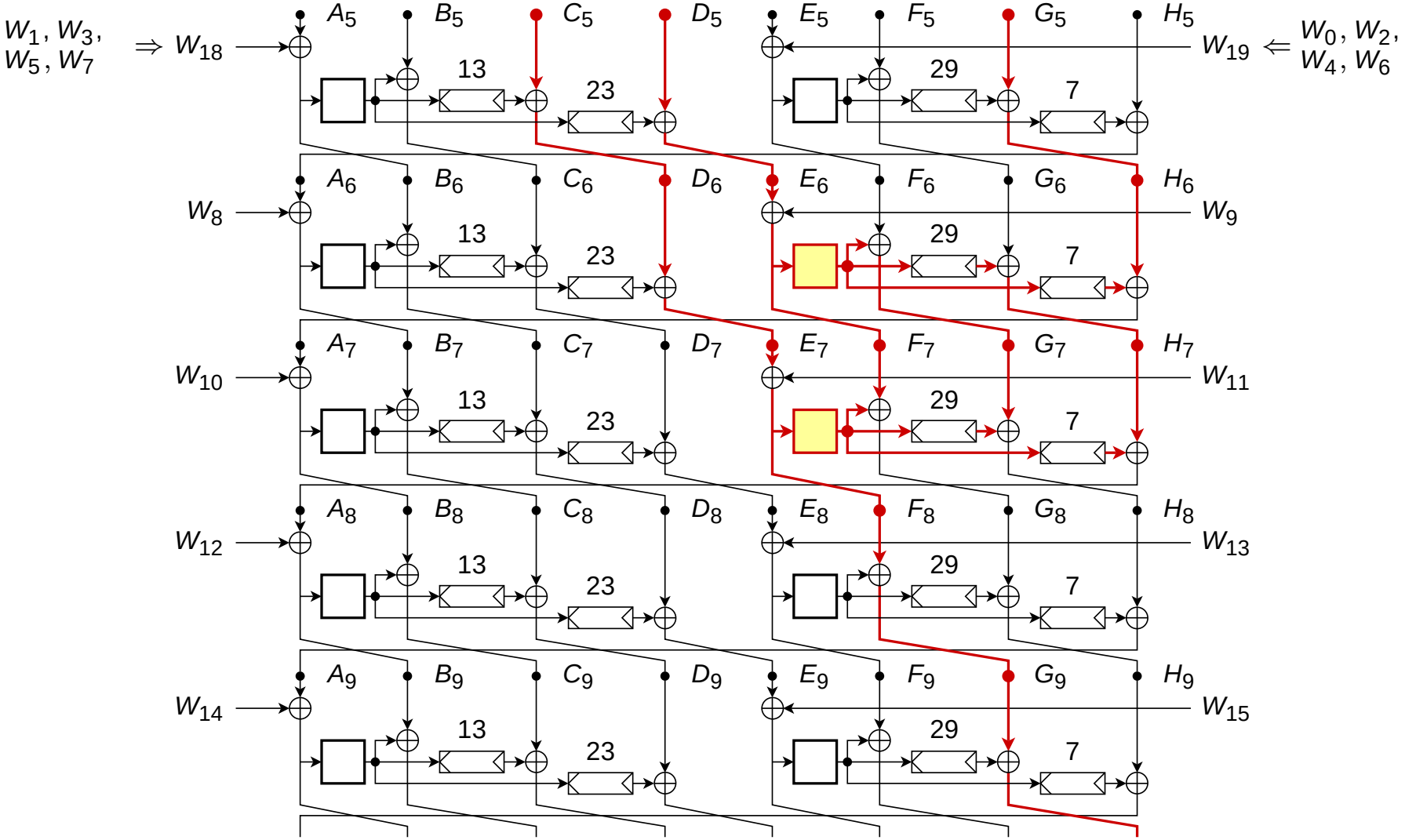
Satisfying conditions

- If we can use initial values, conditions in steps 1–4 are always possible
- Depending on the number of active G, usually we can correct around 16–18 steps
- Might be possible to correct 20 steps in some cases

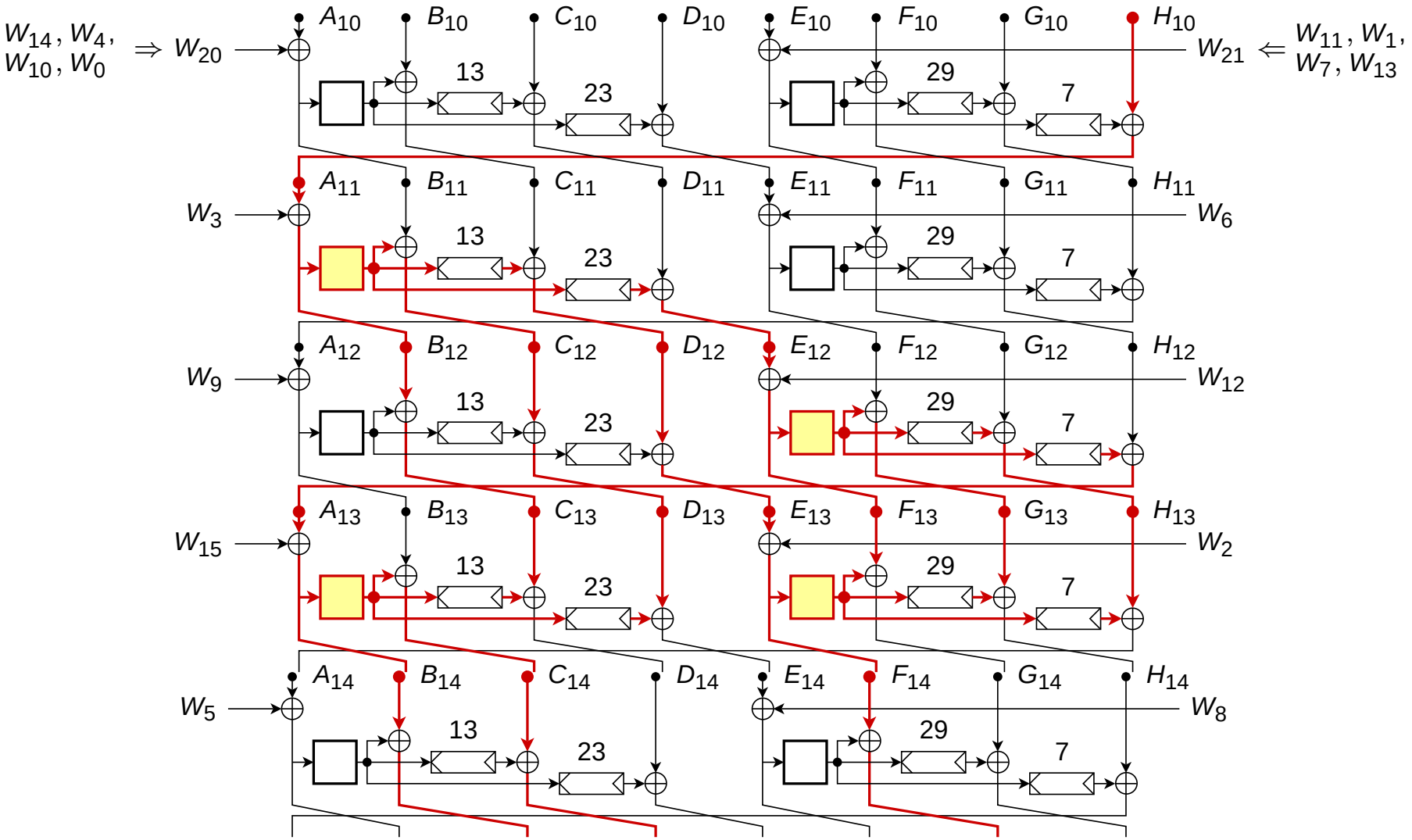
Pseudo-collision path: steps 1 – 5



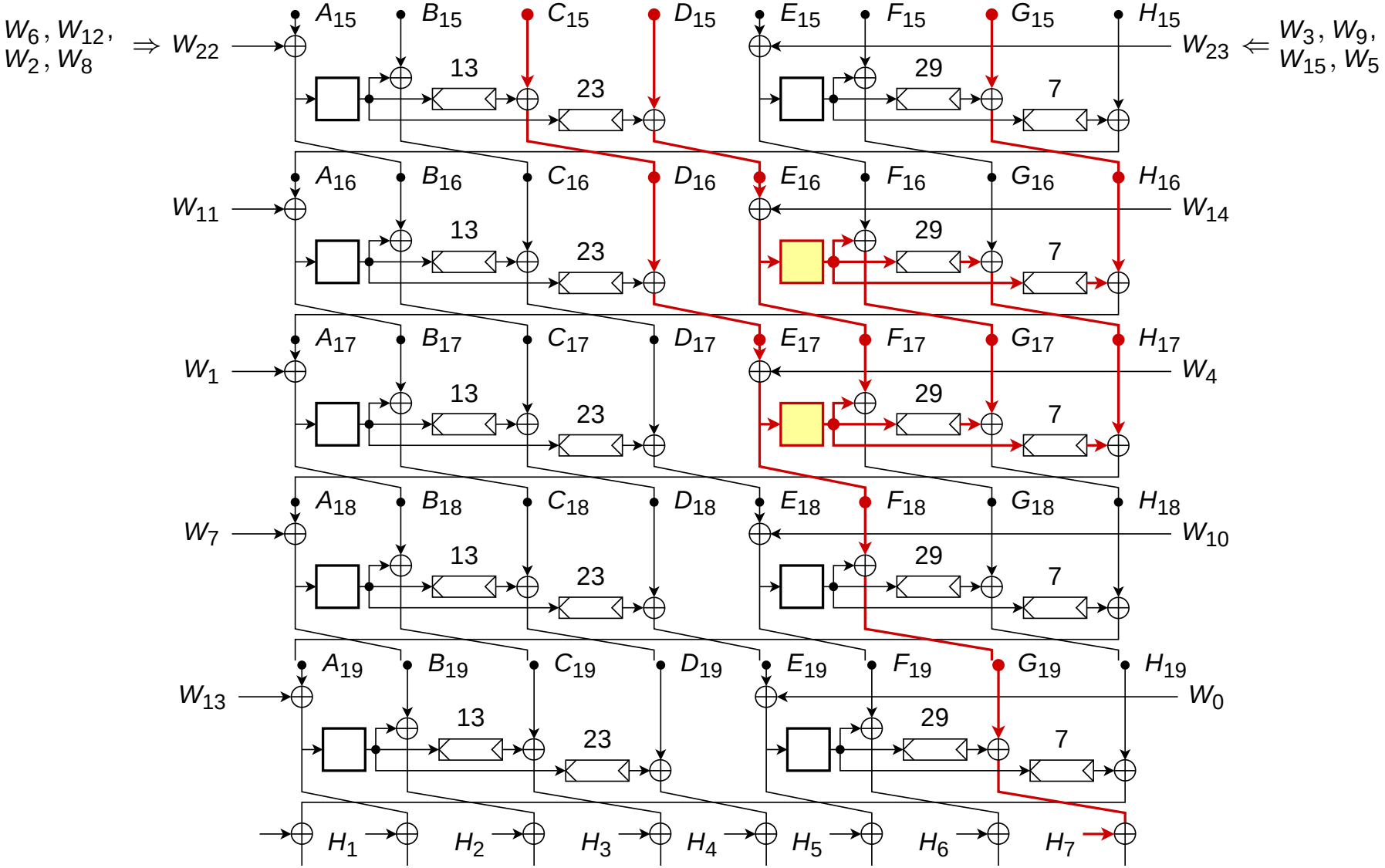
Pseudo-collision path: steps 6 – 10



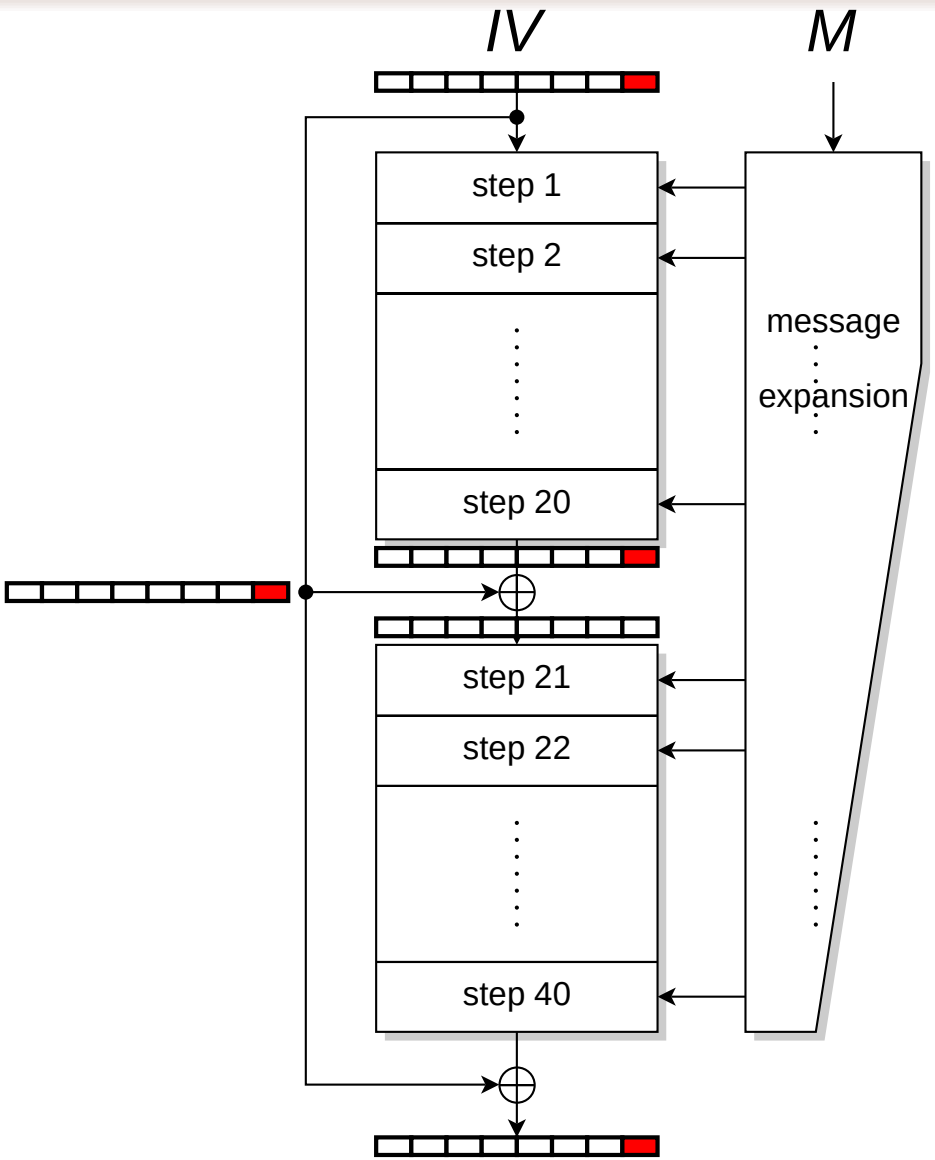
Pseudo-collision path: steps 11 – 15



Pseudo-collision path: steps 16 – 20



Pseudo-collisions for ARIRANG-224/384



- single message block
- can use 14 message words, last two for padding
- message corrections: 12 active G_{256} in steps 2–18, complexity $\approx 2^{23}$
- register H discarded for ARIRANG-224/384
- pseudo-collision for the complete hash function

Summary of results

Compression function		
Result	Complexity	Example
32-bit near-collision for full ARIRANG-256 compress	1	Y
64-bit near-collision for full ARIRANG-512 compress	1	Y
26-step (out of 40) collision for ARIRANG-256/512	1	Y
Hash function		
Result	Complexity	Example
pseudo-collision for full ARIRANG-224/384 hash	2^{23} / 1	Y

Lessons learnt

- Double feed-forward may not be a good idea
- Very simple message expansion can be a weakness
- All-one difference analysis may be useful for designs mixing AES parts with rotations, XORs

Thank you!