Producing collisions for PANAMA, *instantaneously*

Joan Daemen and Gilles Van Assche

STMicroelectronics

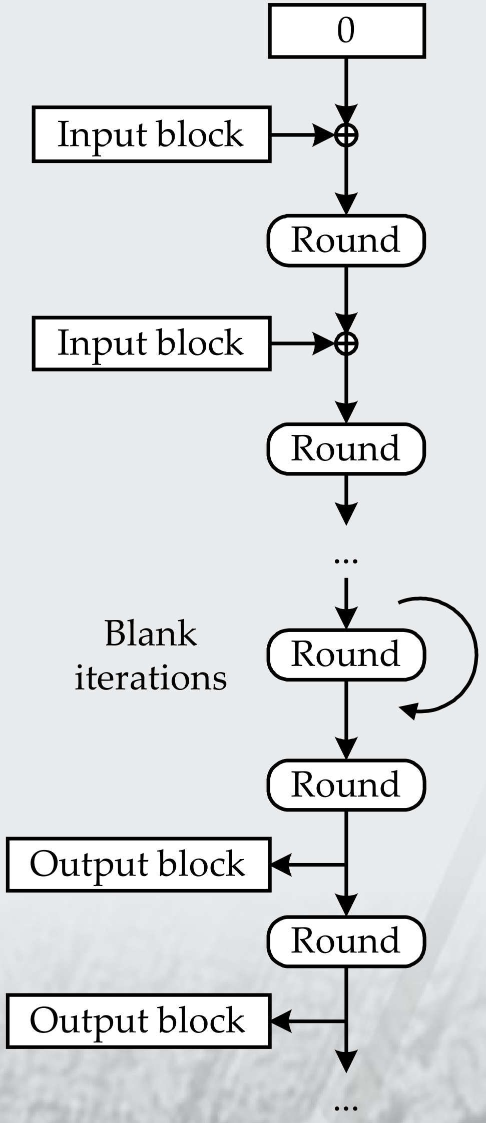
Fast Software Encryption 2007

Outline

- Introduction
- Structure of a collision in PANAMA
- Properties of the non-linear function
- Transferring equations
- Backtracking cost
- Producing the collision
- Conclusion

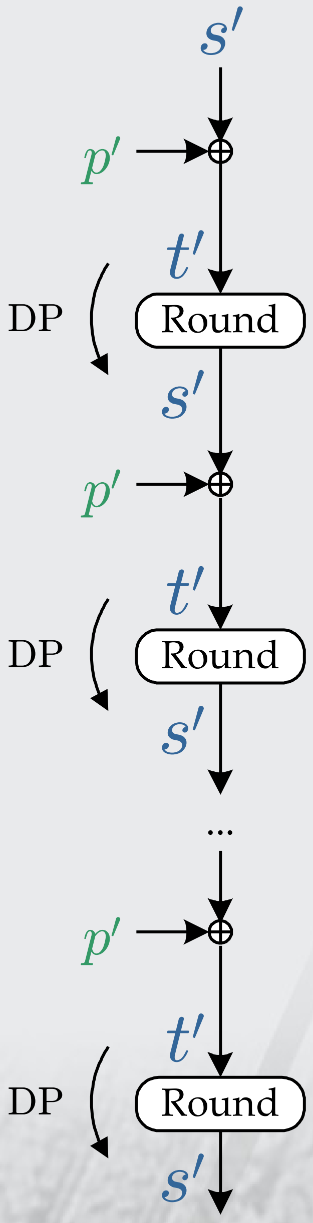
Structure of PANAMA

- Chaining value (CV)
  - Starts from 0
- Iterate with **input** blocks
  - CV size > input block size (l_i)
- Do **blank iterations**
- Iterate with **output** blocks
  - Output mapping
- **Collision in the CV** → collision
  - Blank iterations make it difficult otherwise

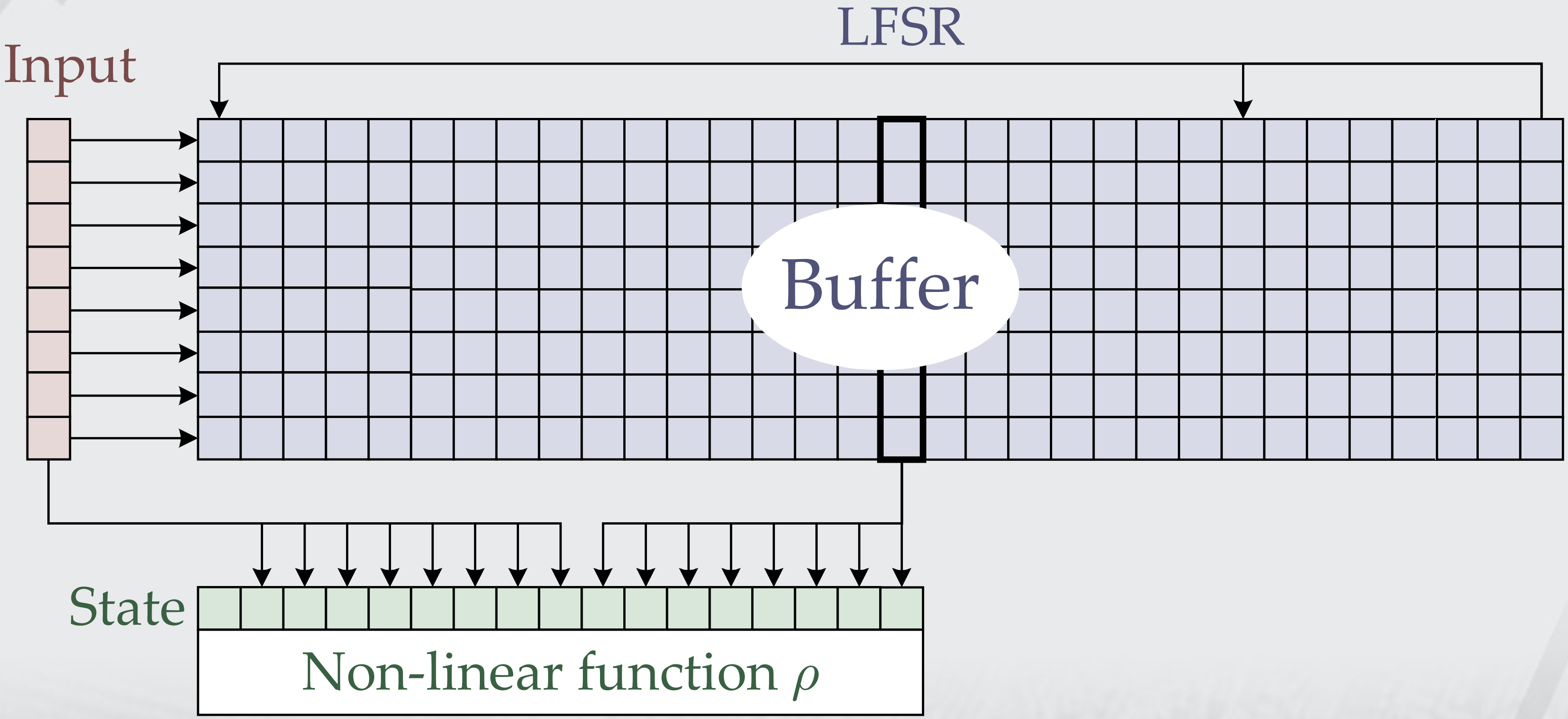


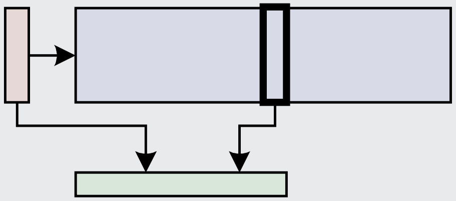
Collision in the chaining value

- Differential trail
  - input differences
  - CV differences
- Collision differential trail
  - Initial CV difference = 0
  - Final CV difference = 0



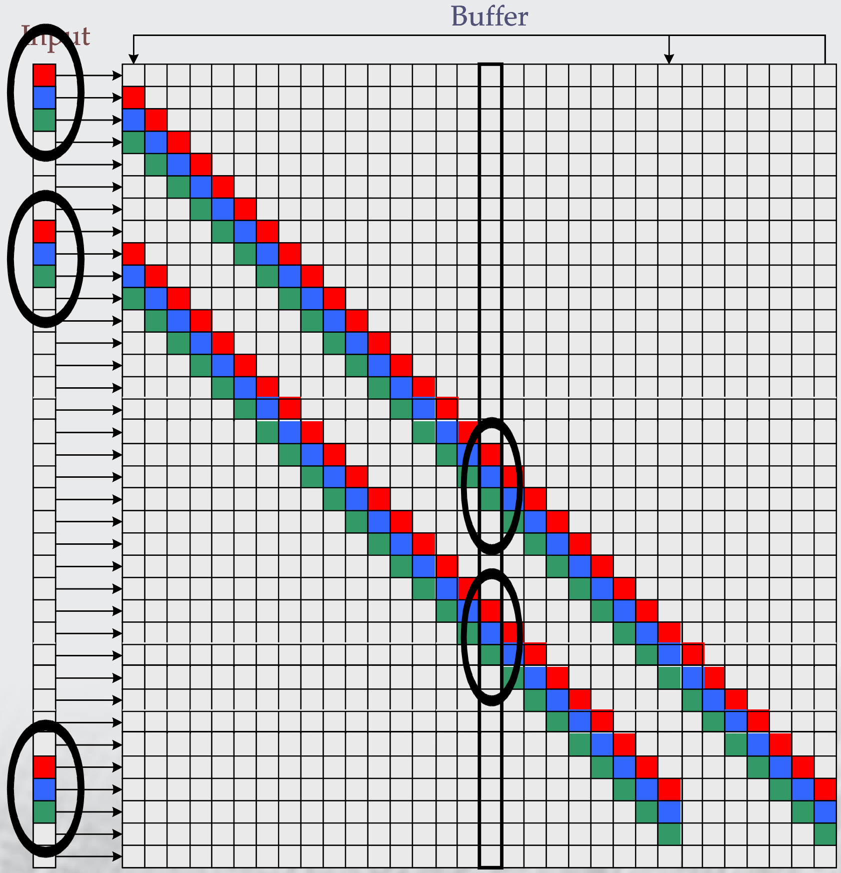
Inside PANAMA = state + buffer





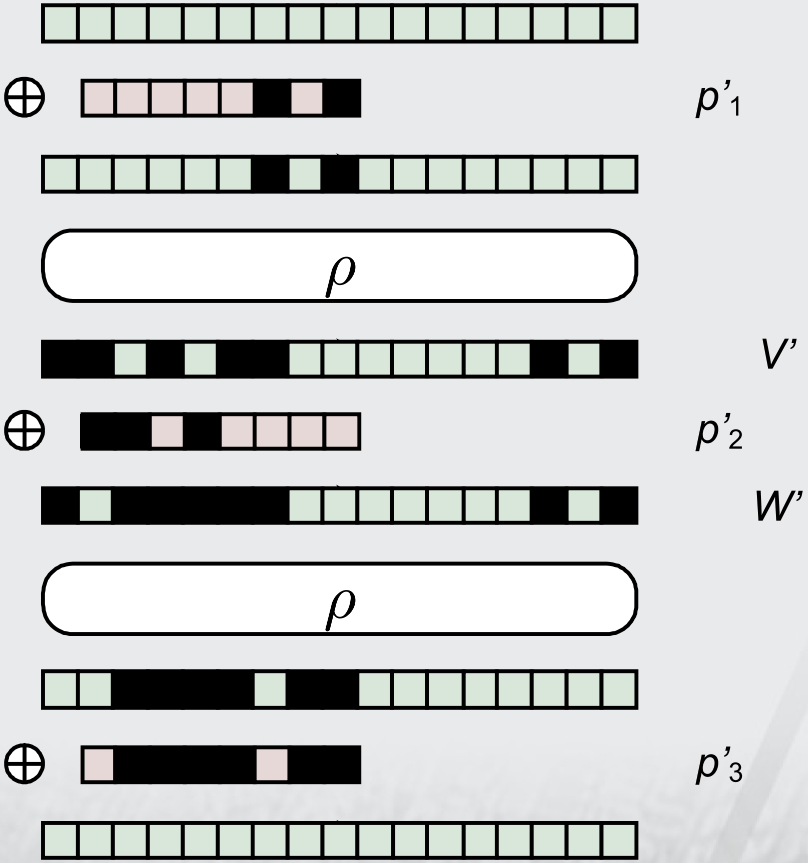
Shape of the differential

- Buffer collisions
  - Atom
  - Rijmen et al.
  - Our attack
- State injection
  - Five instances of …
  - sub-collisions

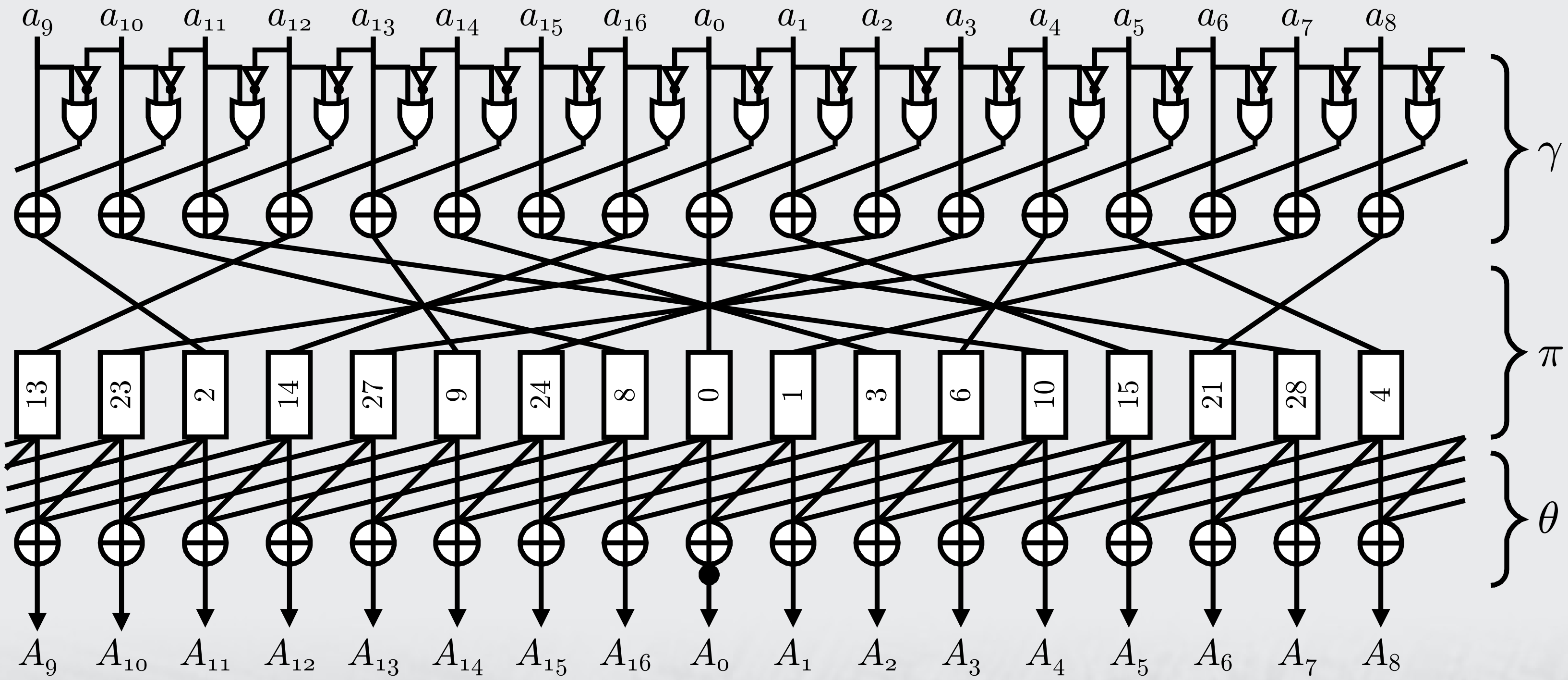


Sub-collision in state

- Two-round differential trail
- completely determined by
  - 3-block input difference sequence
  - State difference
- Two differentials over ρ

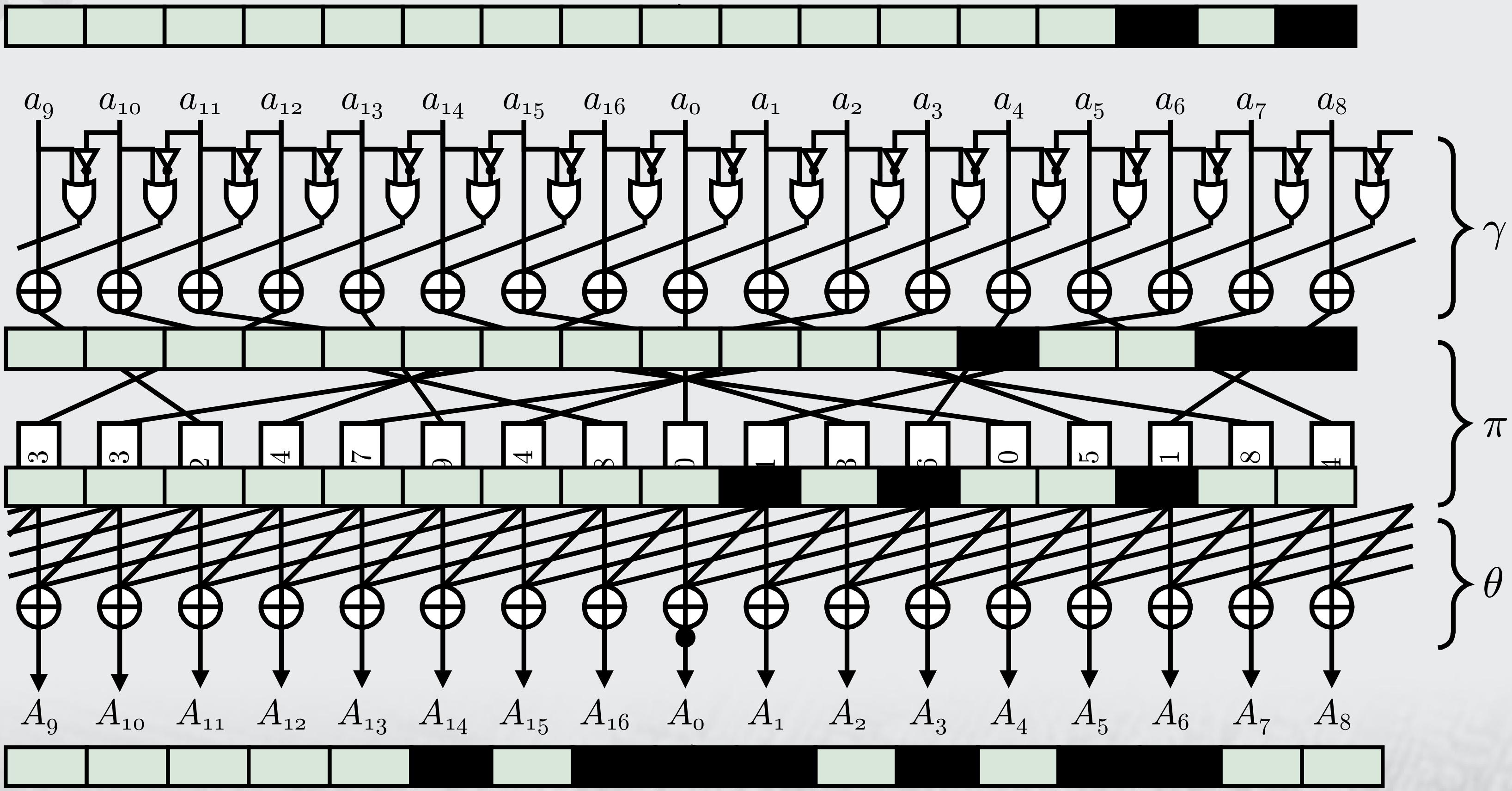


PANAMA's state updating function ρ

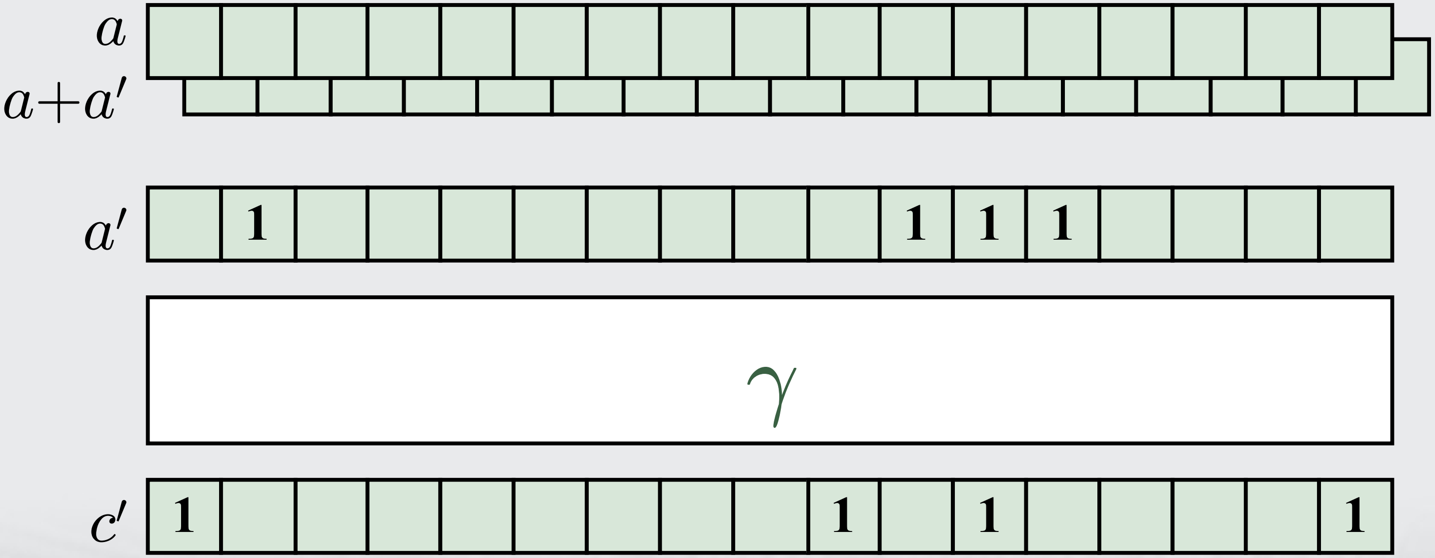


a_9 a_{10} a_{11} a_{12} a_{13} a_{14} a_{15} a_{16} a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8
γ
π
13 23 2 14 27 9 24 8 0 1 3 6 10 15 21 28 4
θ
A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8

PANAMA's state updating function *ρ*


a_9 a_{10} a_{11} a_{12} a_{13} a_{14} a_{15} a_{16} a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8
γ
π
3 3 2 4 7 9 4 8 0 1 3 6 0 5 1 8 4
θ
A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8

Differential over γ



a																	
$a+a'$																	
a'		1									1	1	1				
c'	1									1		1					1

Differential over γ

$a_0=0$

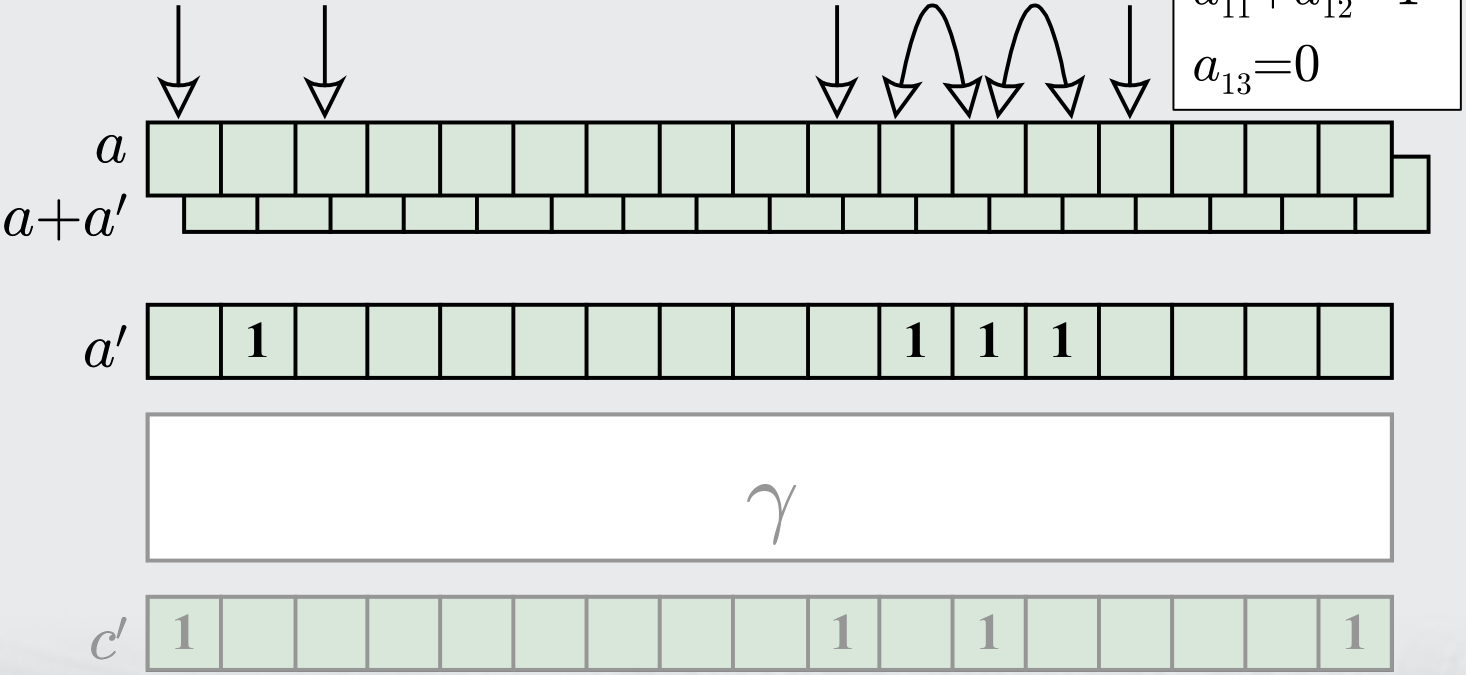
$a_2=1$

$a_9=1$

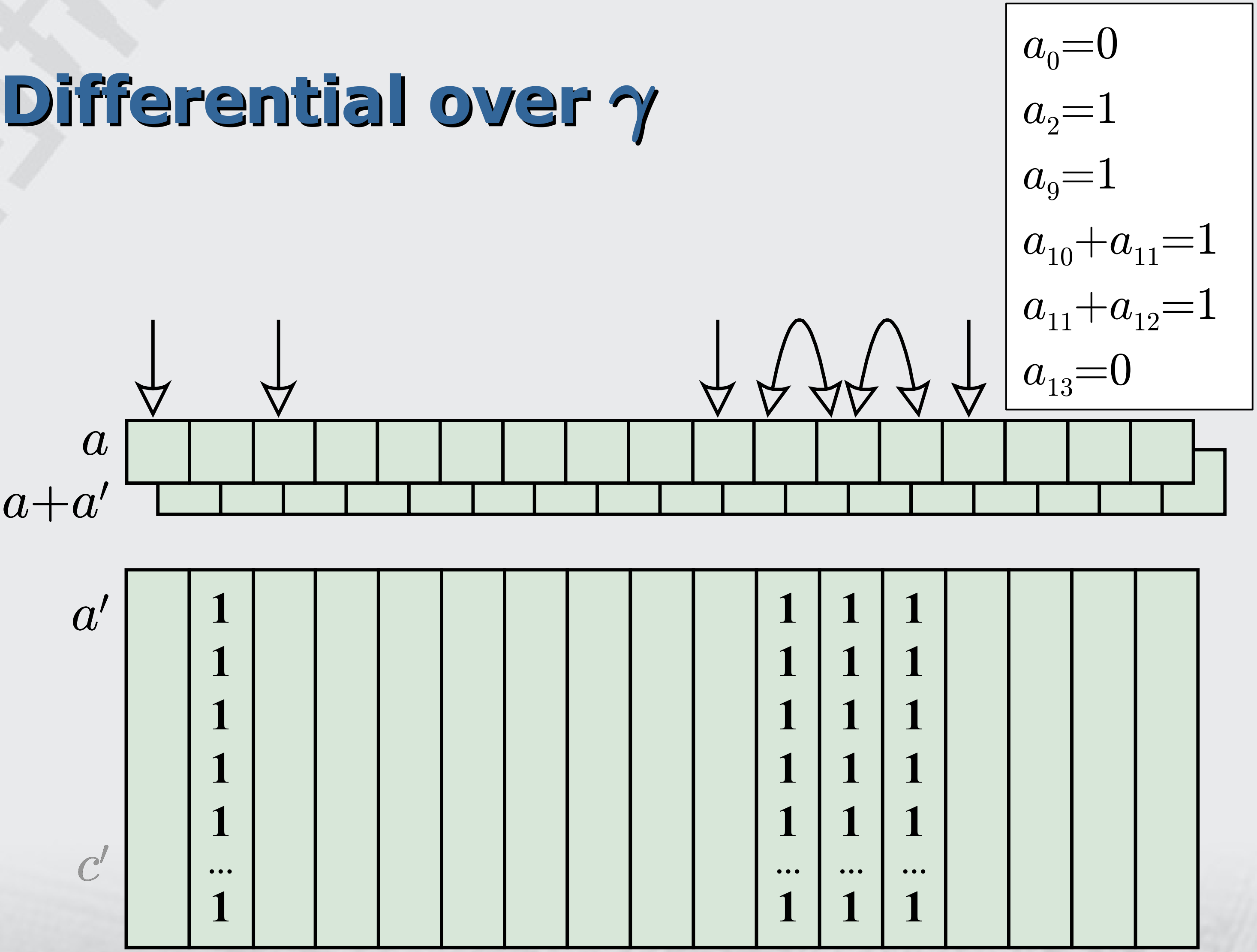
$a_{10}+a_{11}=1$

$a_{11}+a_{12}=1$

$a_{13}=0$



Differential over γ



$a_0 = 0$

$a_2 = 1$

$a_9 = 1$

$a_{10} + a_{11} = 1$

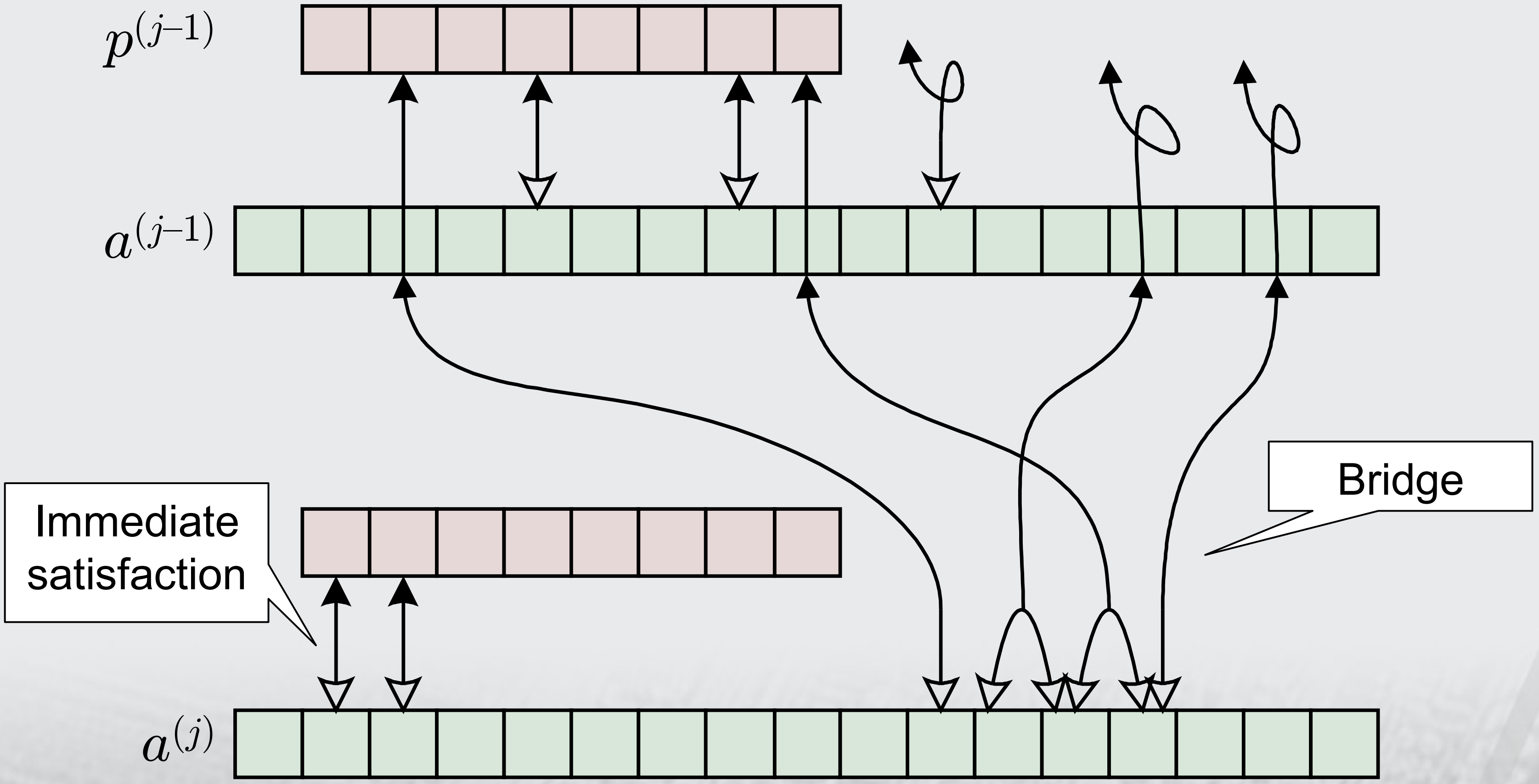
$a_{11} + a_{12} = 1$

$a_{13} = 0$

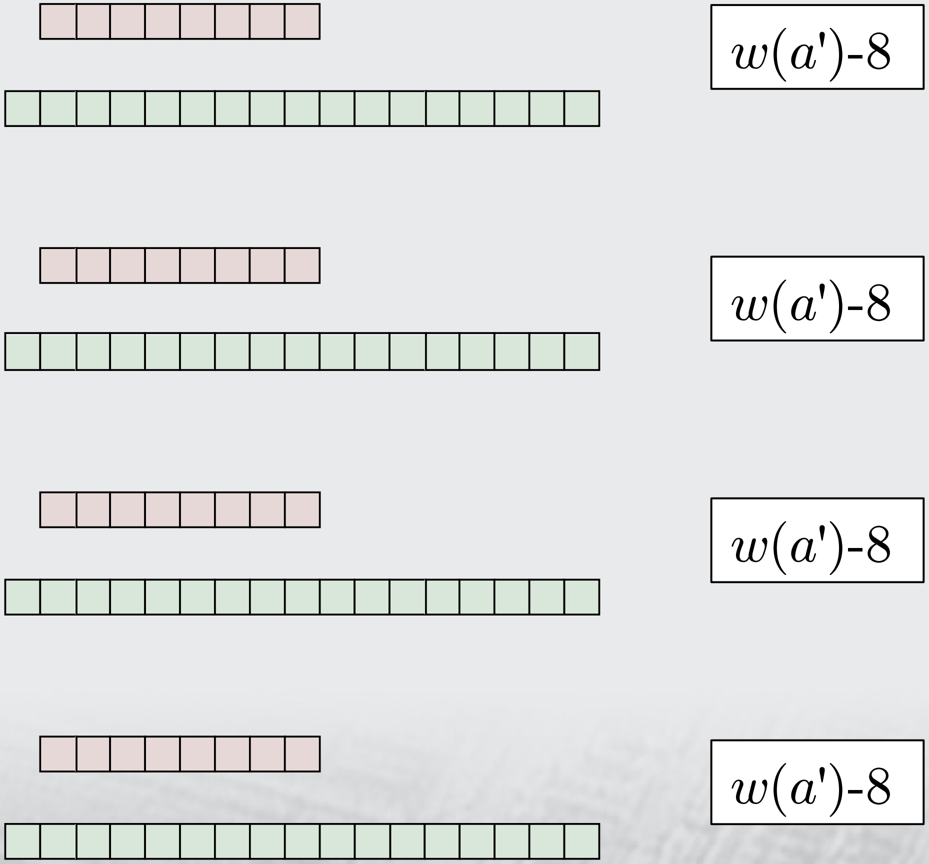
Differential over γ

- Given differential (a', c')
  - Linear conditions on the absolute value a
    - Simple condition (1 bit) or parity conditions (2 bits)
  - Location of conditions only determined by a'
  - Number of conditions is $w(a')$, weight of a'

Transferring conditions



Counting conditions and degrees of freedom



$w(a')$-8

$w(a')$-8

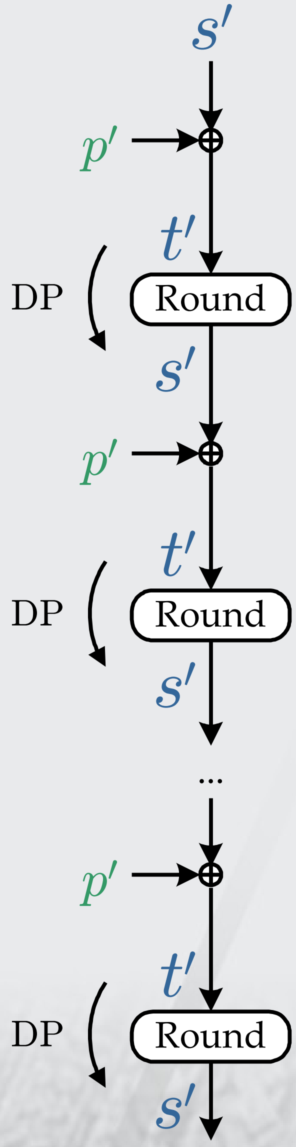
$w(a')$-8

$w(a')$-8

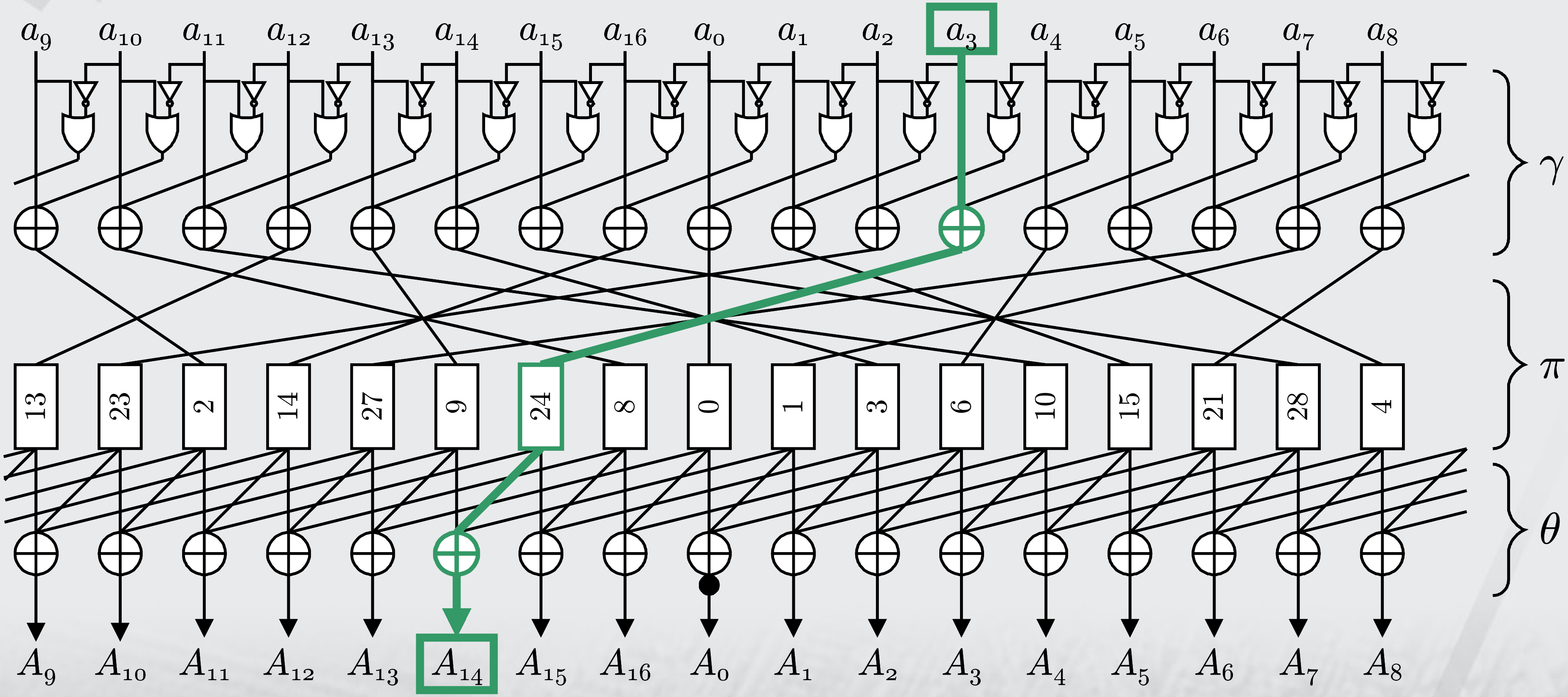
The backtracking cost

$w(a')$	$w(a')$-8
0	-8
0	-8
0	-8
12	4
9	1
14	6
6	-2
2	-6
11	3
9	1
0	-8

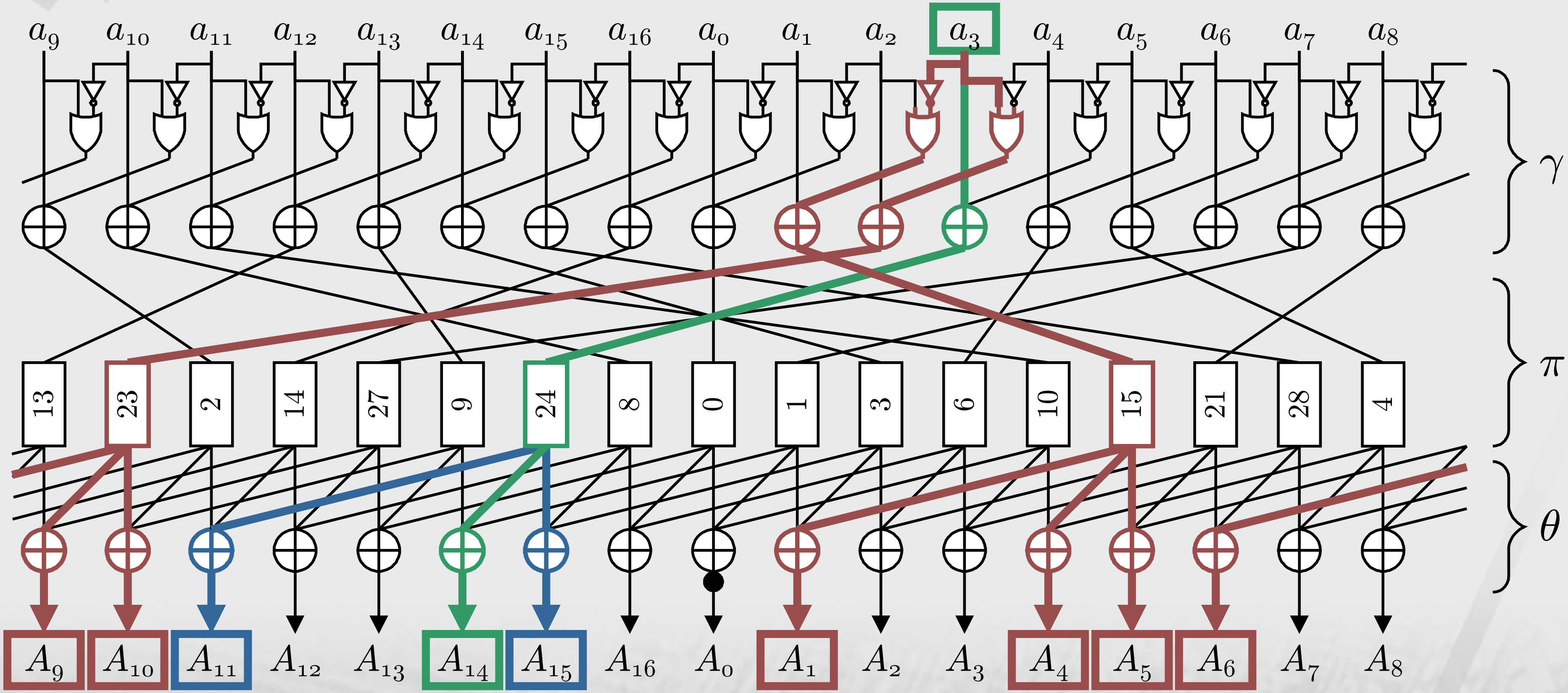
$$\boxed{\max \sum w(a')\text{-}8}$$



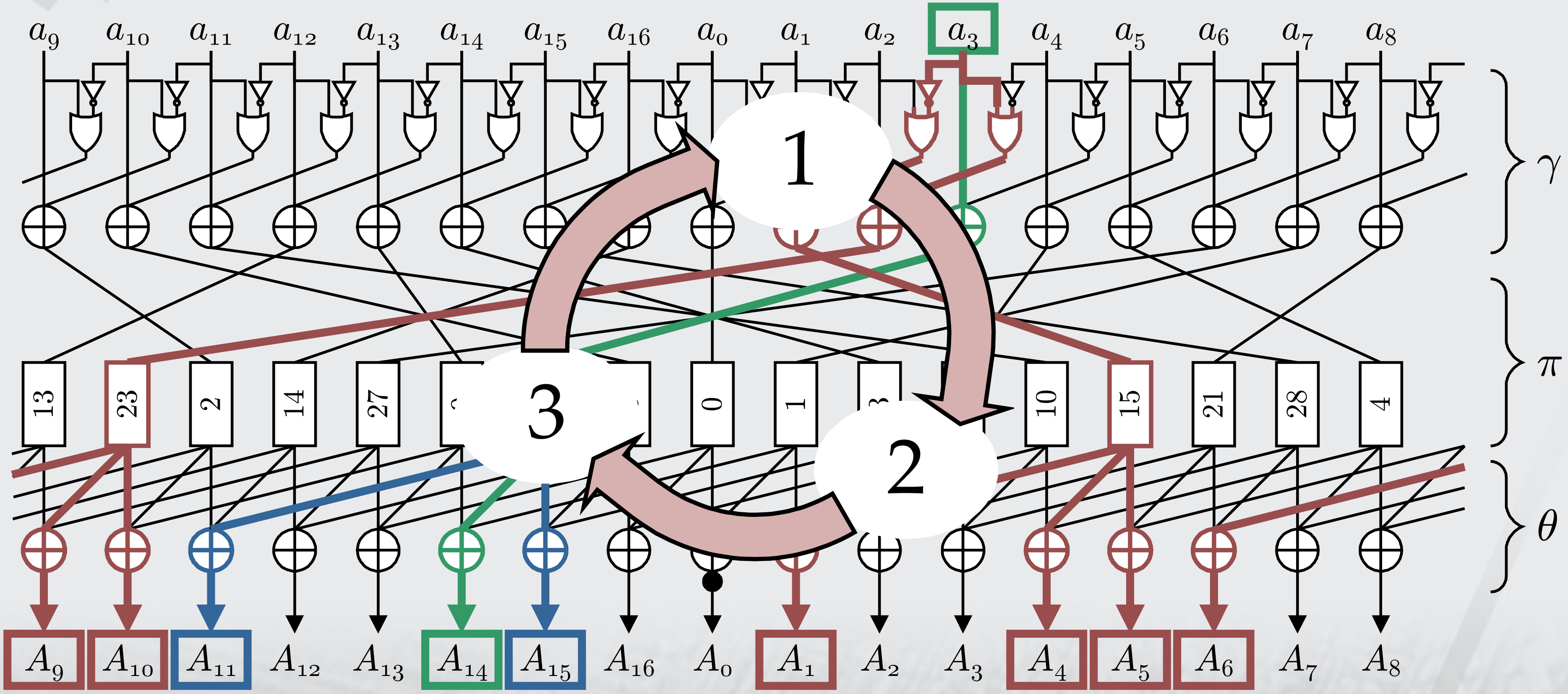
Bridging


a_9
a_{10}
a_{11}
a_{12}
a_{13}
a_{14}
a_{15}
a_{16}
a_0
a_1
a_2
a_3
a_4
a_5
a_6
a_7
a_8
γ
π
13
23
2
14
27
9
24
8
0
1
3
6
10
15
21
28
4
θ
A_9
A_{10}
A_{11}
A_{12}
A_{13}
A_{14}
A_{15}
A_{16}
A_0
A_1
A_2
A_3
A_4
A_5
A_6
A_7
A_8

Dependency removal

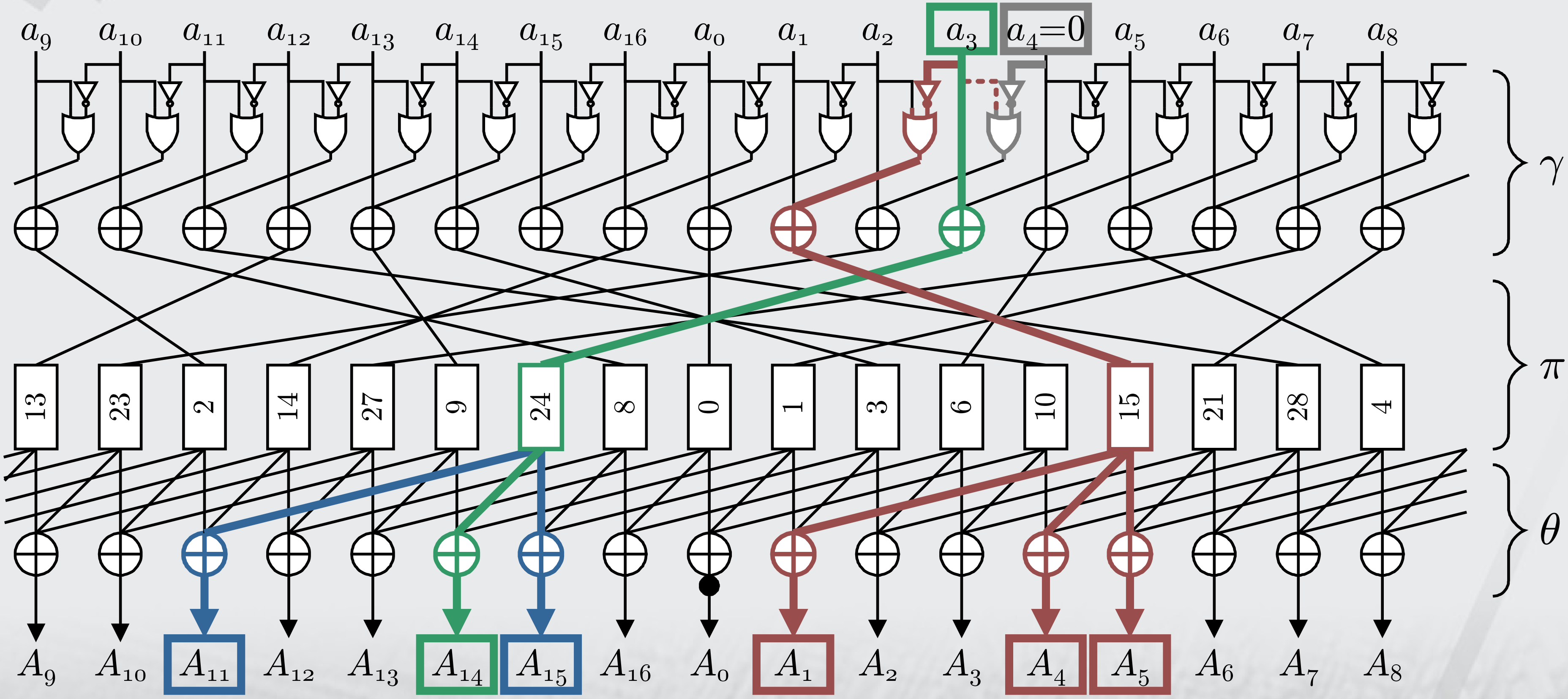

a_9 a_{10} a_{11} a_{12} a_{13} a_{14} a_{15} a_{16} a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8
γ
π
13 23 2 14 27 9 24 8 0 1 3 6 10 15 21 28 4
θ
A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8

Dependency removal

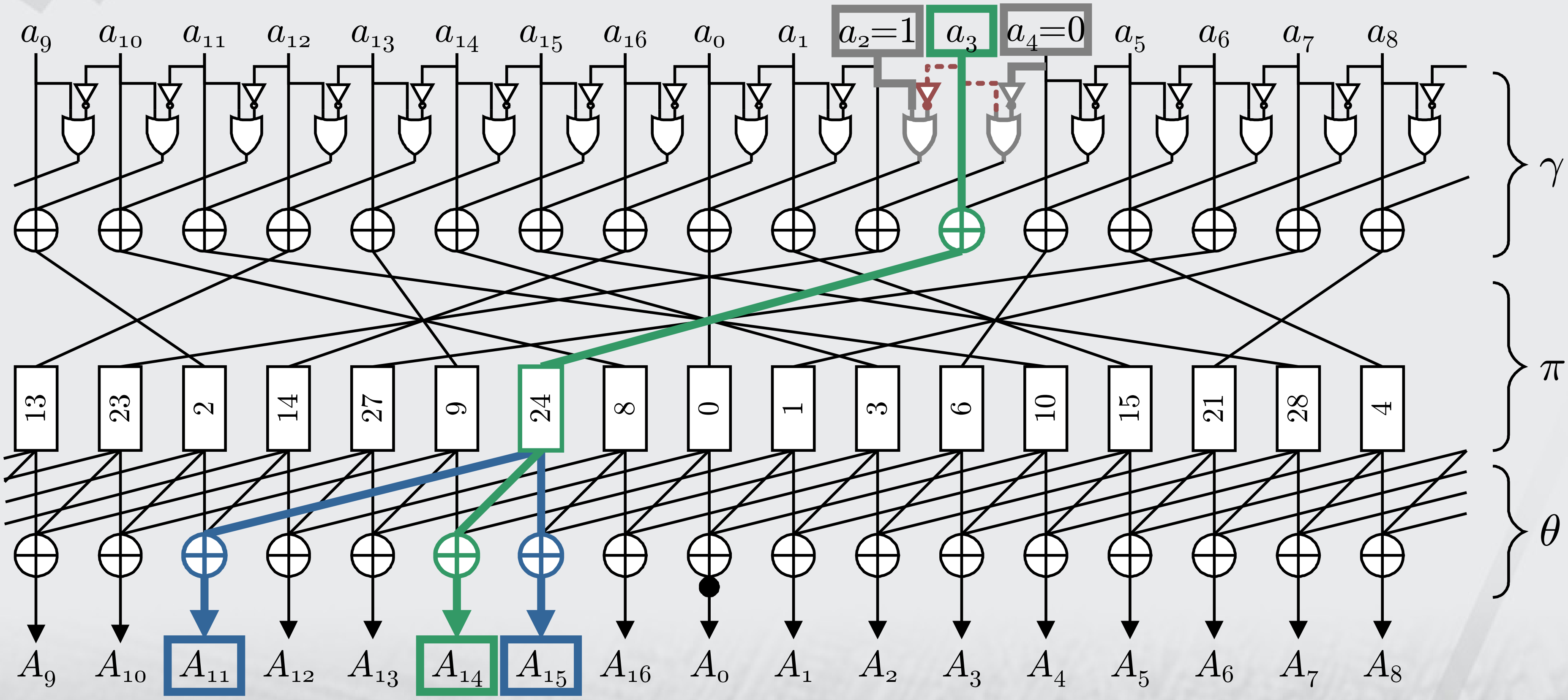


a_9 a_{10} a_{11} a_{12} a_{13} a_{14} a_{15} a_{16} a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8
γ
π
θ
13
23
2
14
27
0
1
10
15
21
28
4
1
2
3
A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8

Dependency removal


a_9 a_{10} a_{11} a_{12} a_{13} a_{14} a_{15} a_{16} a_0 a_1 a_2 a_3 $a_4=0$ a_5 a_6 a_7 a_8
γ
π
θ
13 23 2 14 27 9 24 8 0 1 3 6 10 15 21 28 4
A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8

Dependency removal



Producing the collision

- Choose a differential
  - Least number of conditions to be bridged
- Work out the equations
  - Immediate satisfaction
  - Bridges
  - Dependencies
- Finally, it takes
  - 35 input blocks
  - 30 bridges
  - So a total of 65 evaluations of the round function

Conclusion

- PANAMA hash function is broken
  - Source file to generate collisions available
- The way forward: RADIOGATÚN
  - Feedback from state to buffer
  - Lower number of input words per round
  - Backtracking cost
  - Ongoing

http://radiogatun.noekeon.org/panama