Recent Attacks on Hash Functions

Vincent Rijmen
http://www.iaik.tugraz.at/aboutus/people/rijmen/index.php

Hash&Stream, Salzburg, 2007/02/01

Institute for Applied Information Processing and Communications (IAIK) - Krypto Group

Faculty of Computer Science
Graz University of Technology

Overview

- Differential cryptanalysis
- SHA and SHA-1
- Chabaud-Joux attack on SHA [1998]
- Wang attack on SHA-1 [2004]

Differential cryptanalysis

- Developed for block ciphers [BiSh90]
- Don’t look at individual inputs
- Consider *pairs* with a certain *difference*
- Study *propagation* of differences

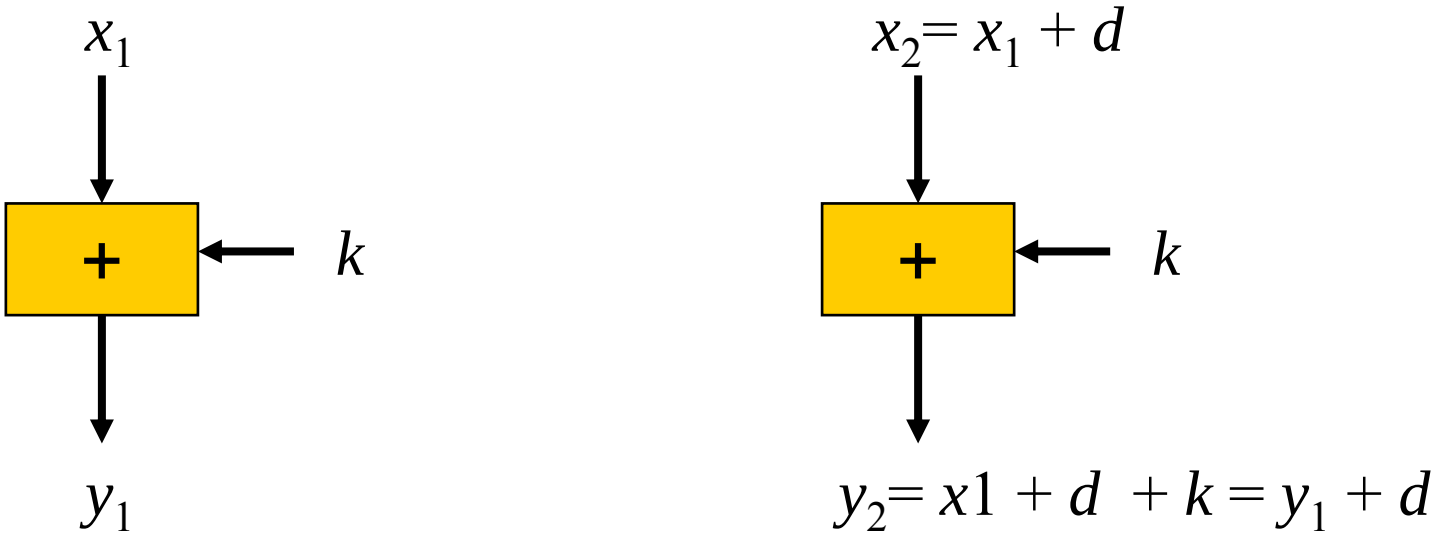
Differential attack

1. Study propagation of differences: choose *good* differences (characteristic)

2. Construct pair of messages
   - Differences as in the characteristic
   - *Right* pair

Propagation of differences

- Addition with a constant



- Difference doesn't change

Propagation of differences

- Linear maps:

$$f(x + d) - f(x) = f(x) + f(d) - f(x) = f(d)$$

- Nonlinear maps:

$$g(x + d) - g(x) = ?$$

- Several output differences possible
  - Each occurring with a frequency (*probability*)

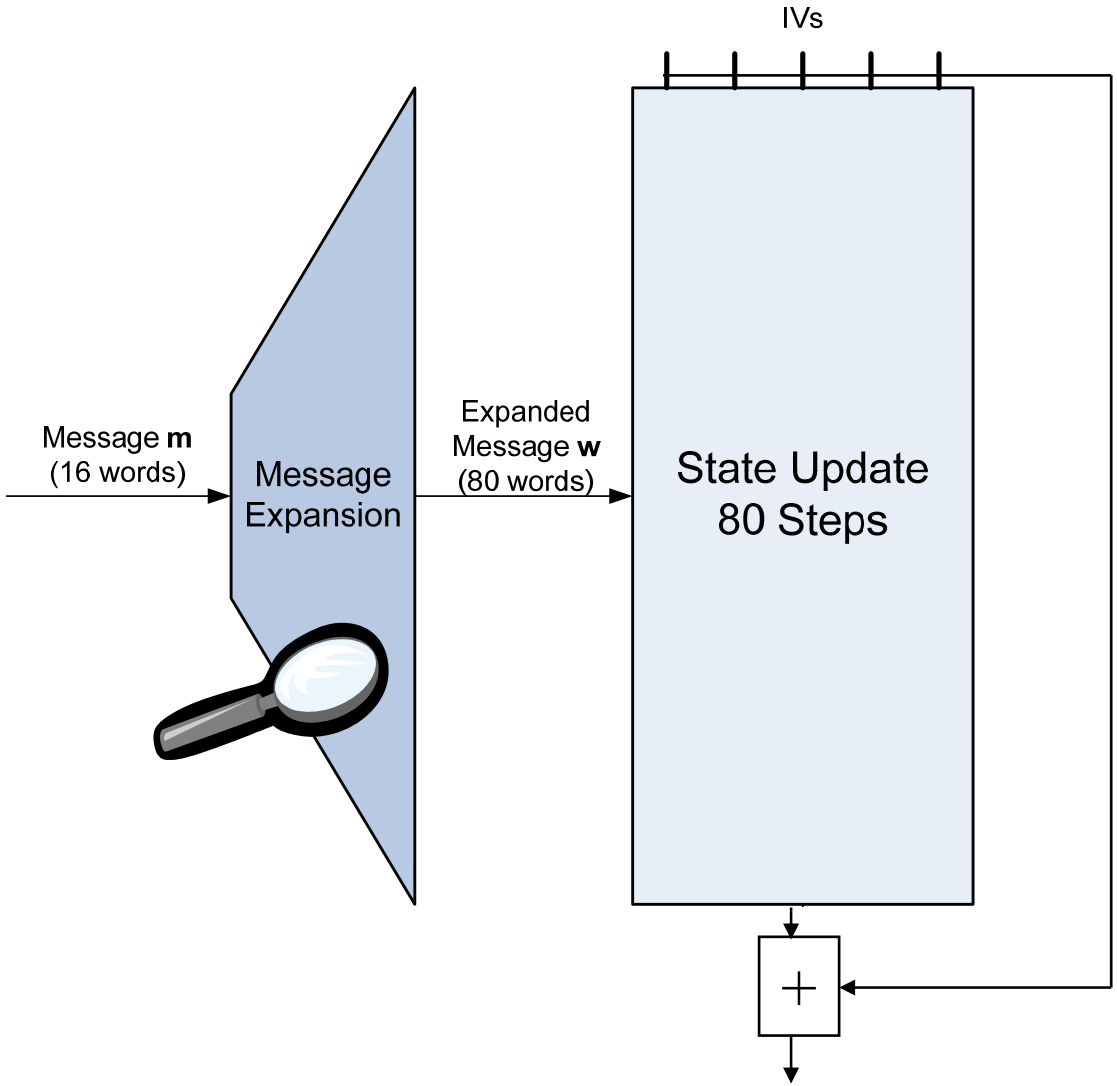
Choice of difference

- Every group operation defines a `difference'
- Most popular choices:
  1. XOR difference
  2. Modular addition

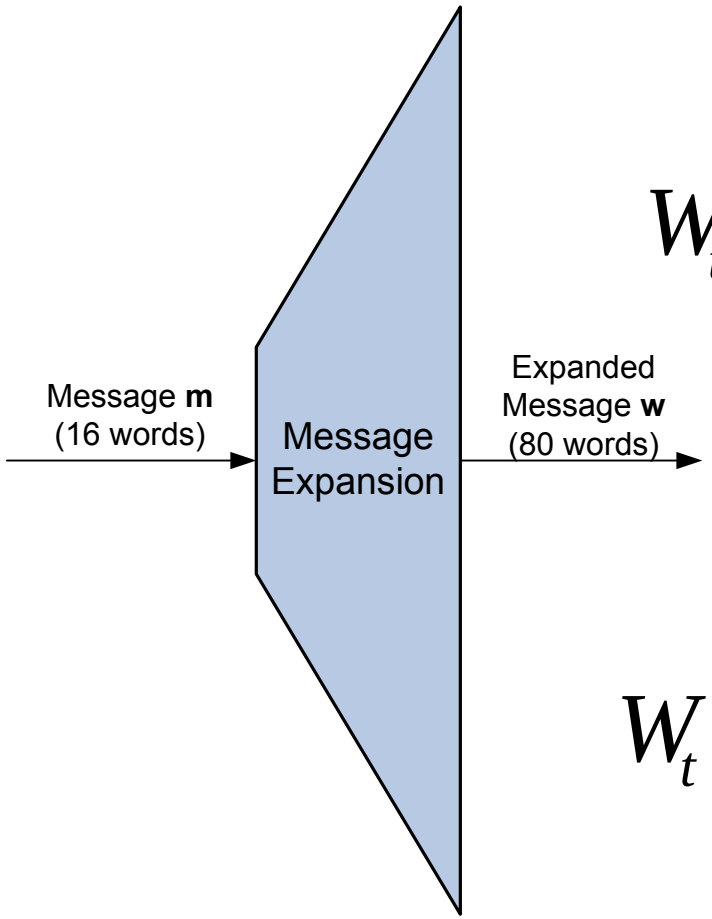
Differential attack

1. Find sequence of differences (*characteristic*)
   - With high probability
   - = occurring for large fraction of the pairs
   - For collision: last difference = 0
2. Construct *right* pair of messages
   - = with these differences

Outline of SHA compression function



Outline of SHA – Message Expansion



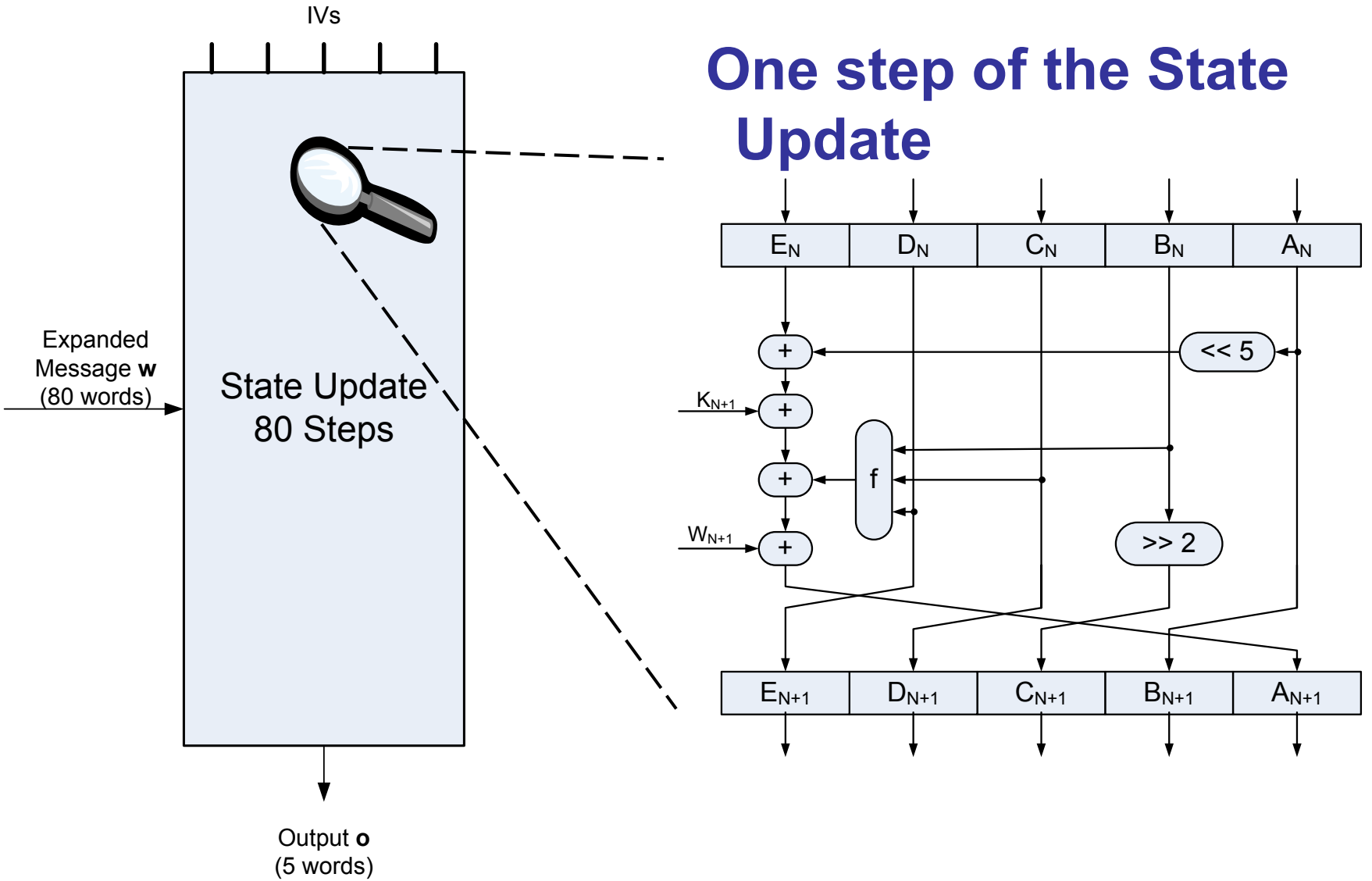
SHA

$$W_t = \begin{cases} M_t & for (0 \le t \le 15) \\ W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16} & for (16 \le t \le 79) \end{cases}$$

SHA-1

$$W_t = \begin{cases} M_t & for (0 \le t \le 15) \\ (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) <\!\!< 1 & for (16 \le t \le 79) \end{cases}$$

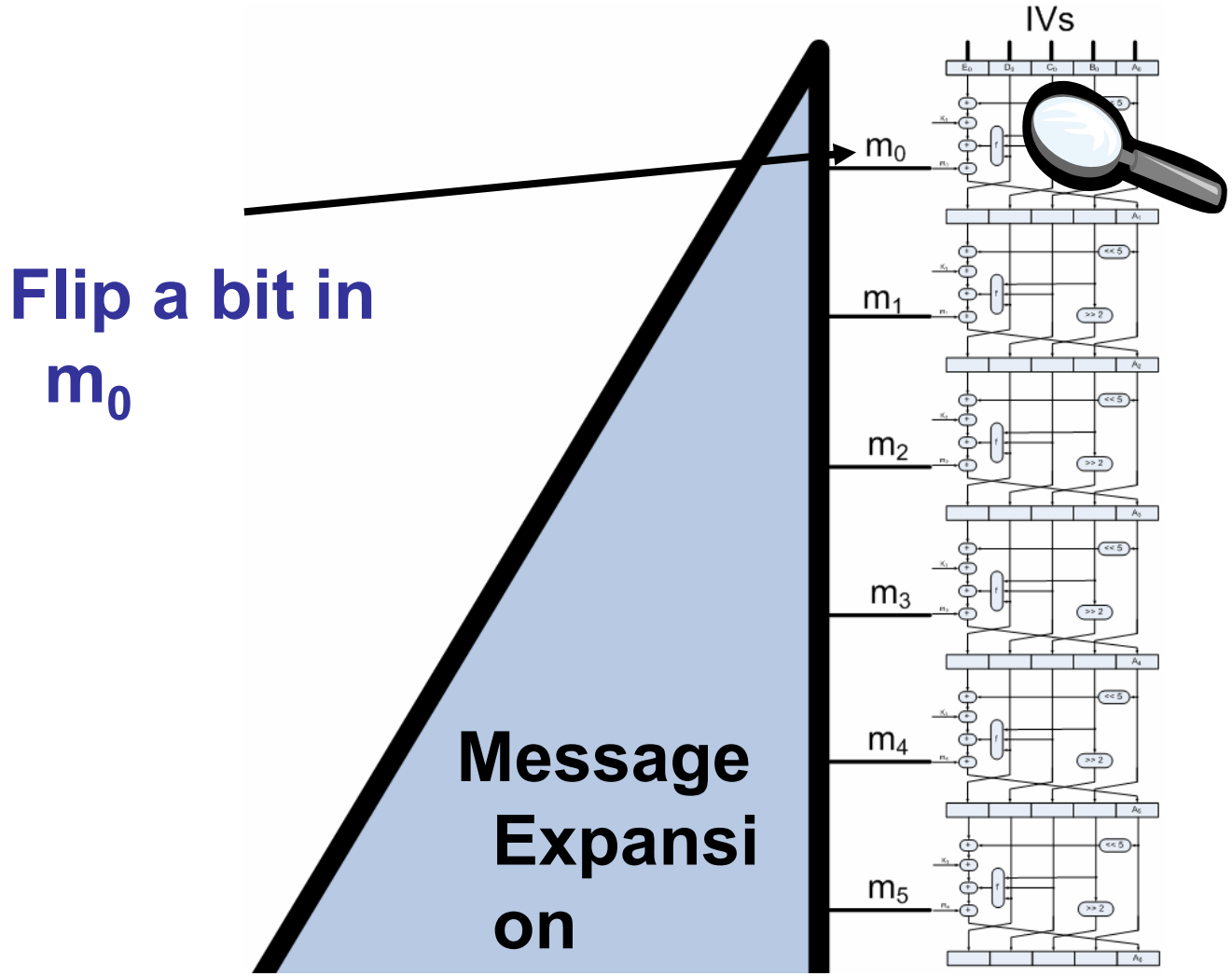
Outline of SHA – State Update



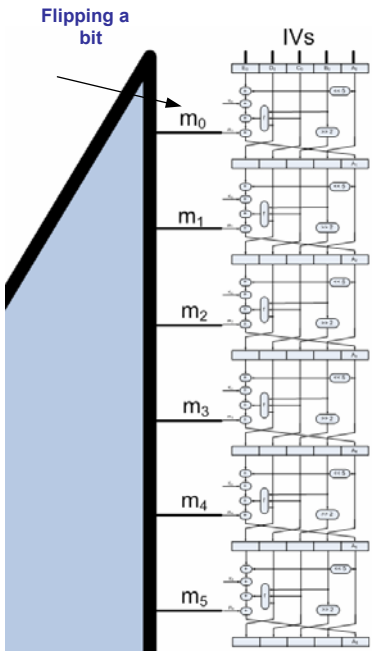
State update

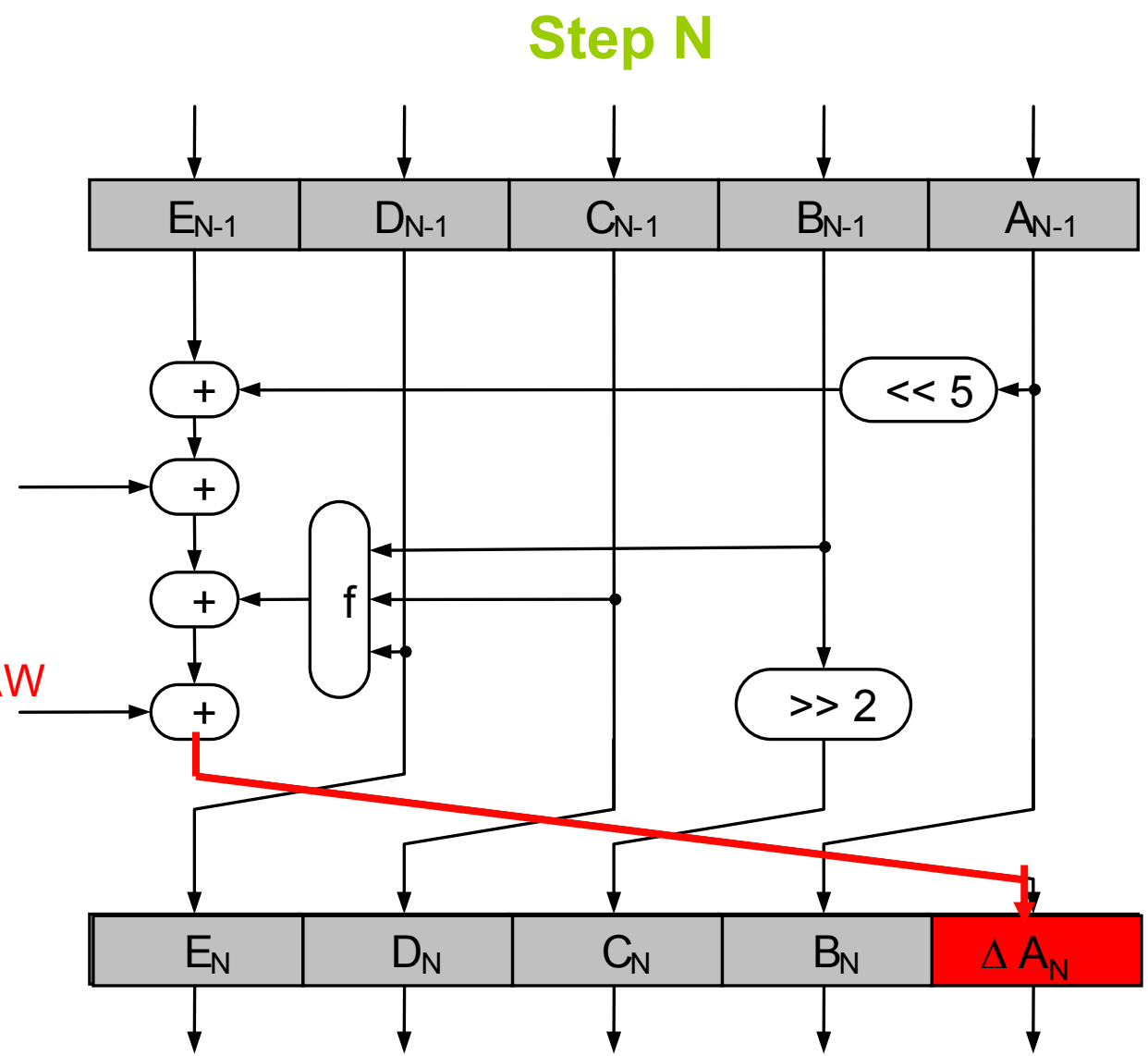
- Modular additions
- Rotations
- Boolean function f
  - Steps 1-20: “IF”
  - Steps 21-40: XOR
  - Steps 41-60: “MAJ”
  - Steps 61-80: XOR

Propagation of a small difference



Propagation of a small difference

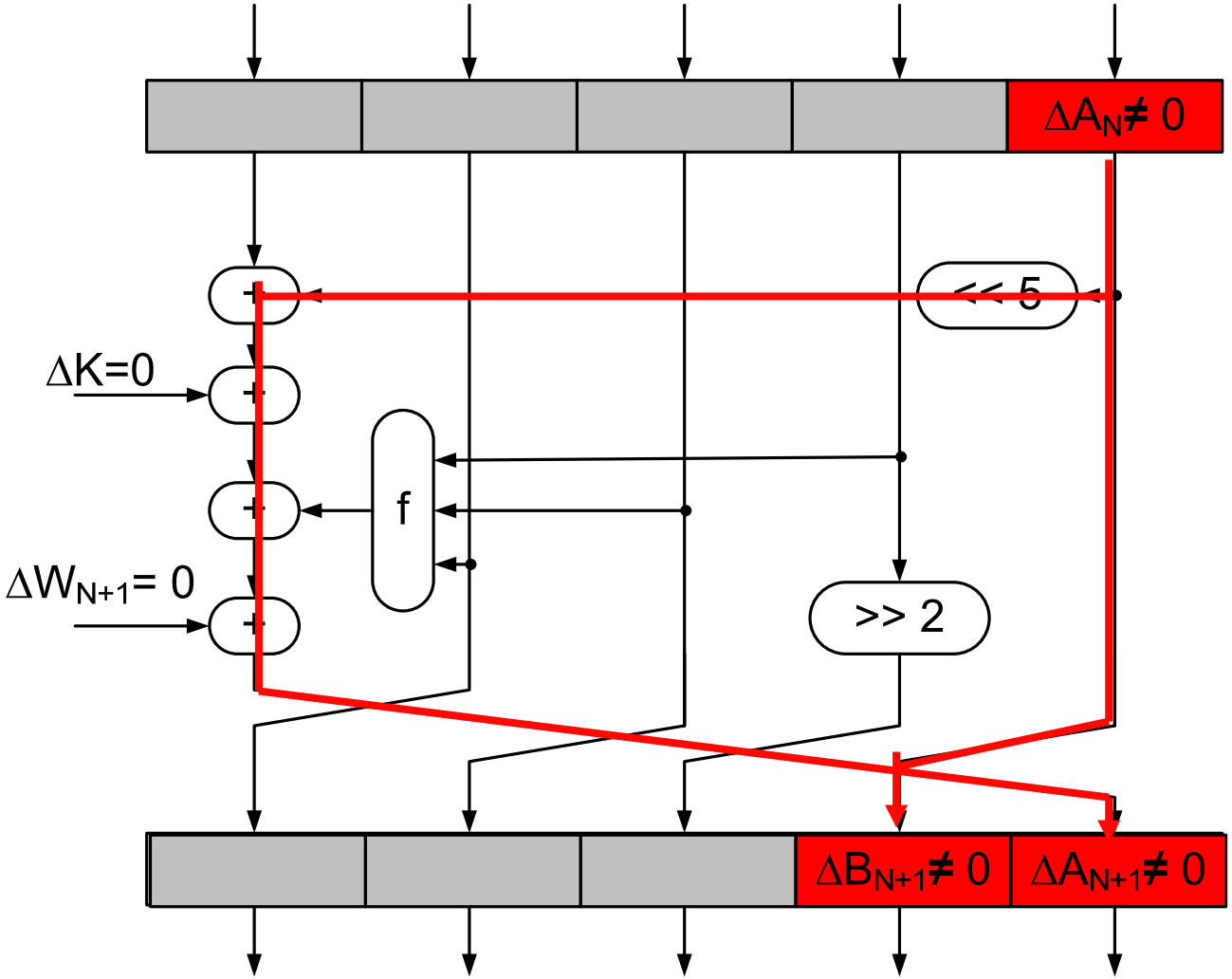




Flip a bit

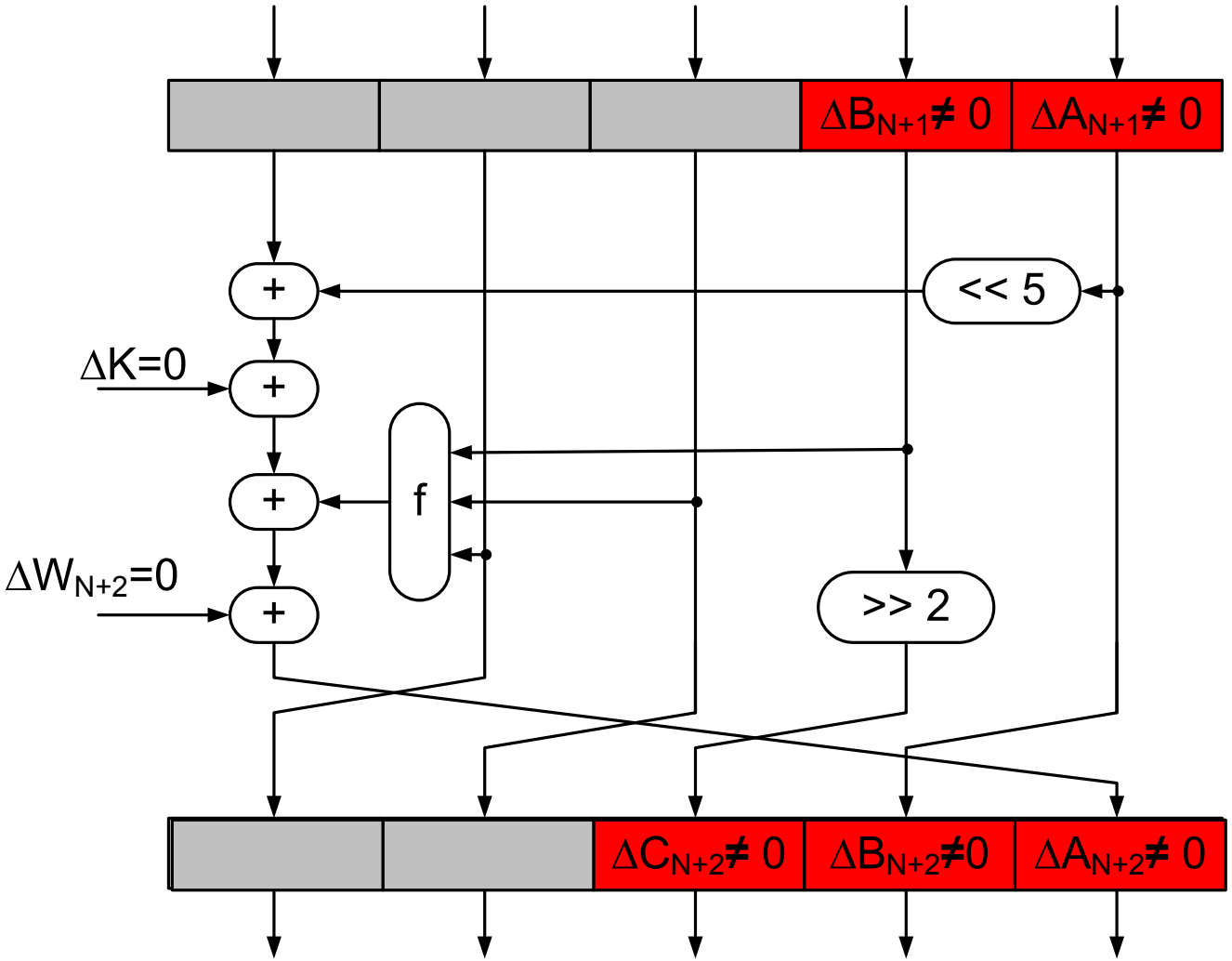
Propagation of a small difference

Step N+1



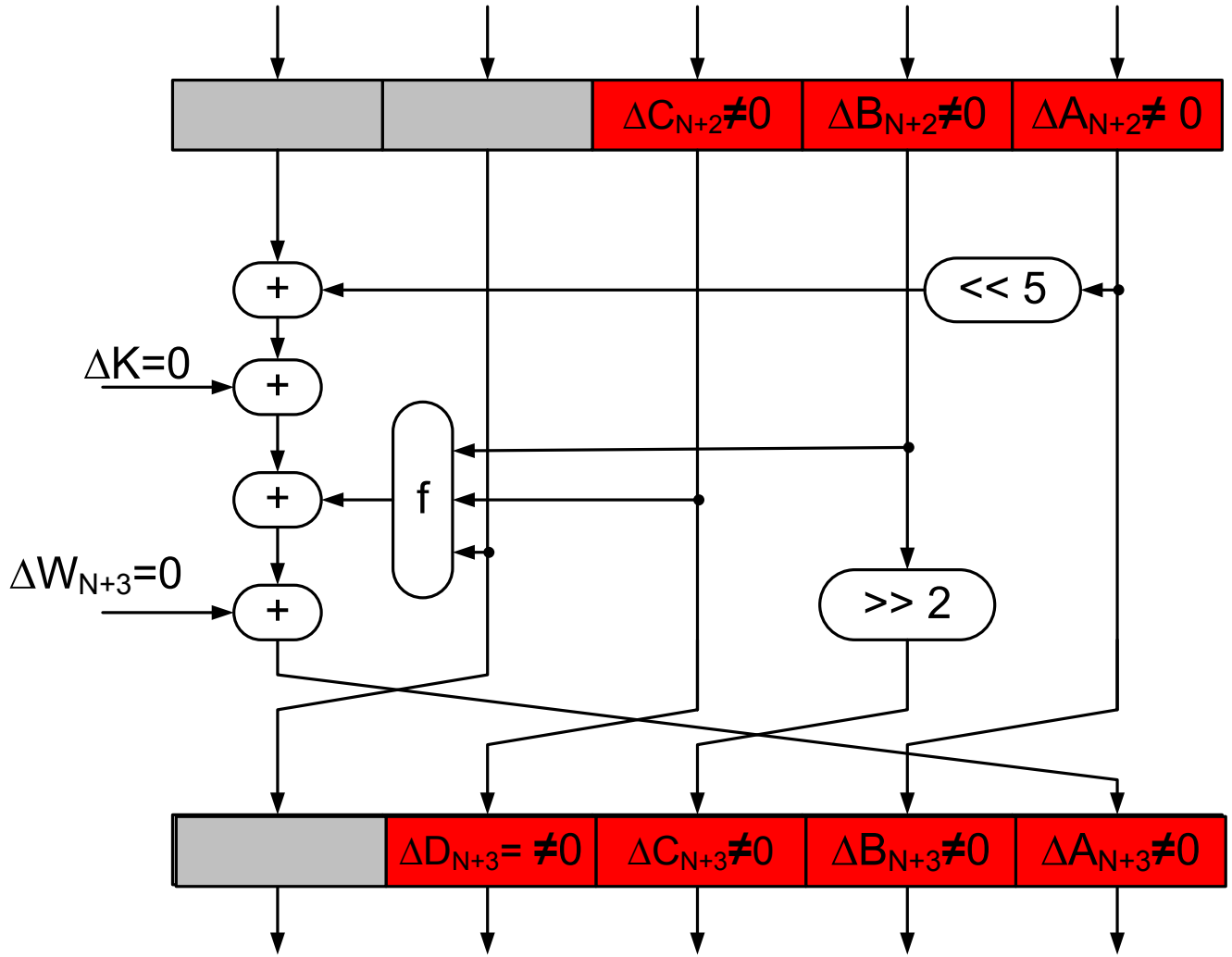
Propagation of a small difference

Step N+2



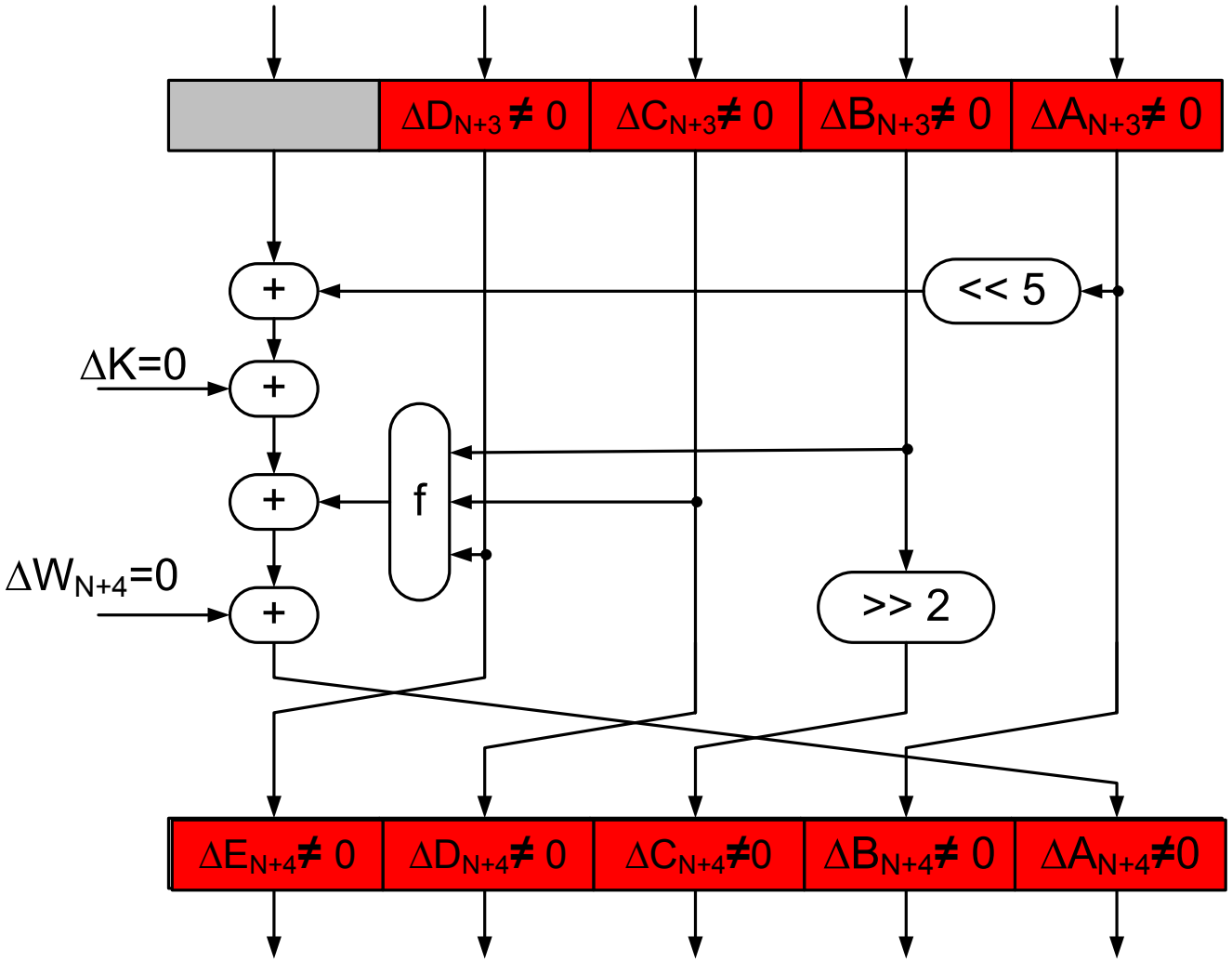
Propagation of a small difference

Step N+3



Propagation of a small difference

Step N+4

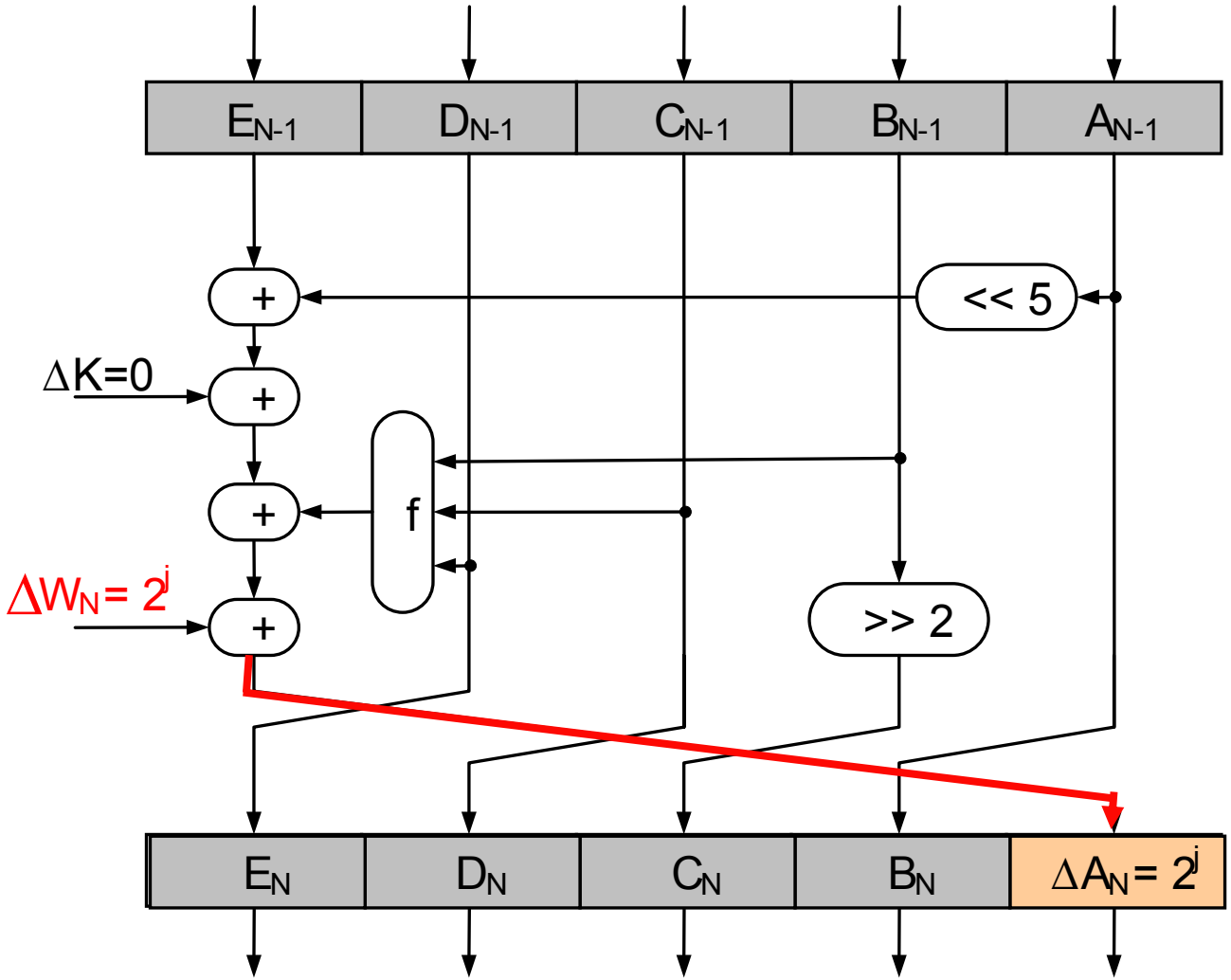


Differential cryptanalysis of SHA

- Small differences quickly expand
- Approach by Chabaud & Joux [1998]
  - Perturbations and corrections
  - Theoretical attack on SHA
  - Later on improved to practical attack [2004]
- Difference = XOR operation

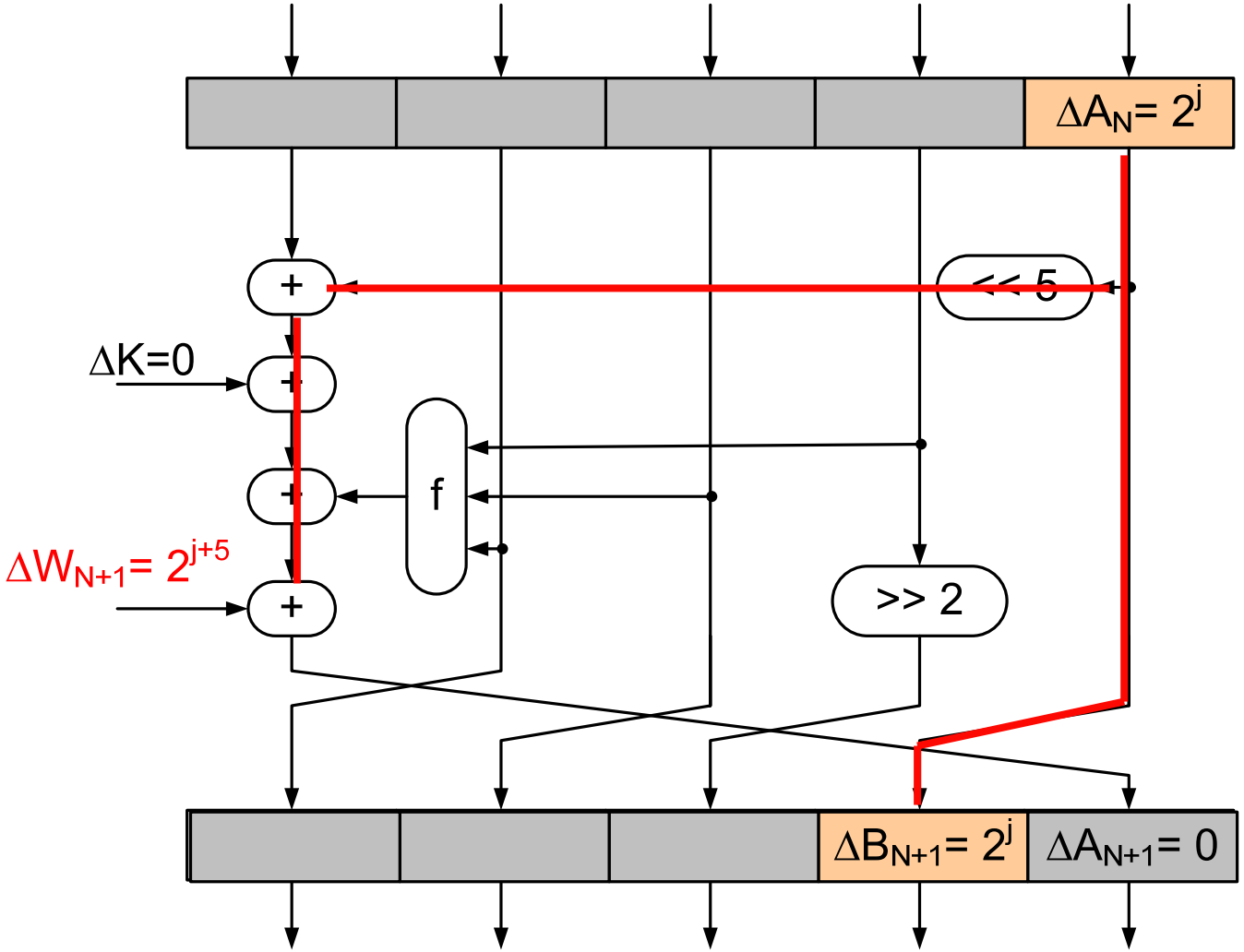
Perturbation

Step N



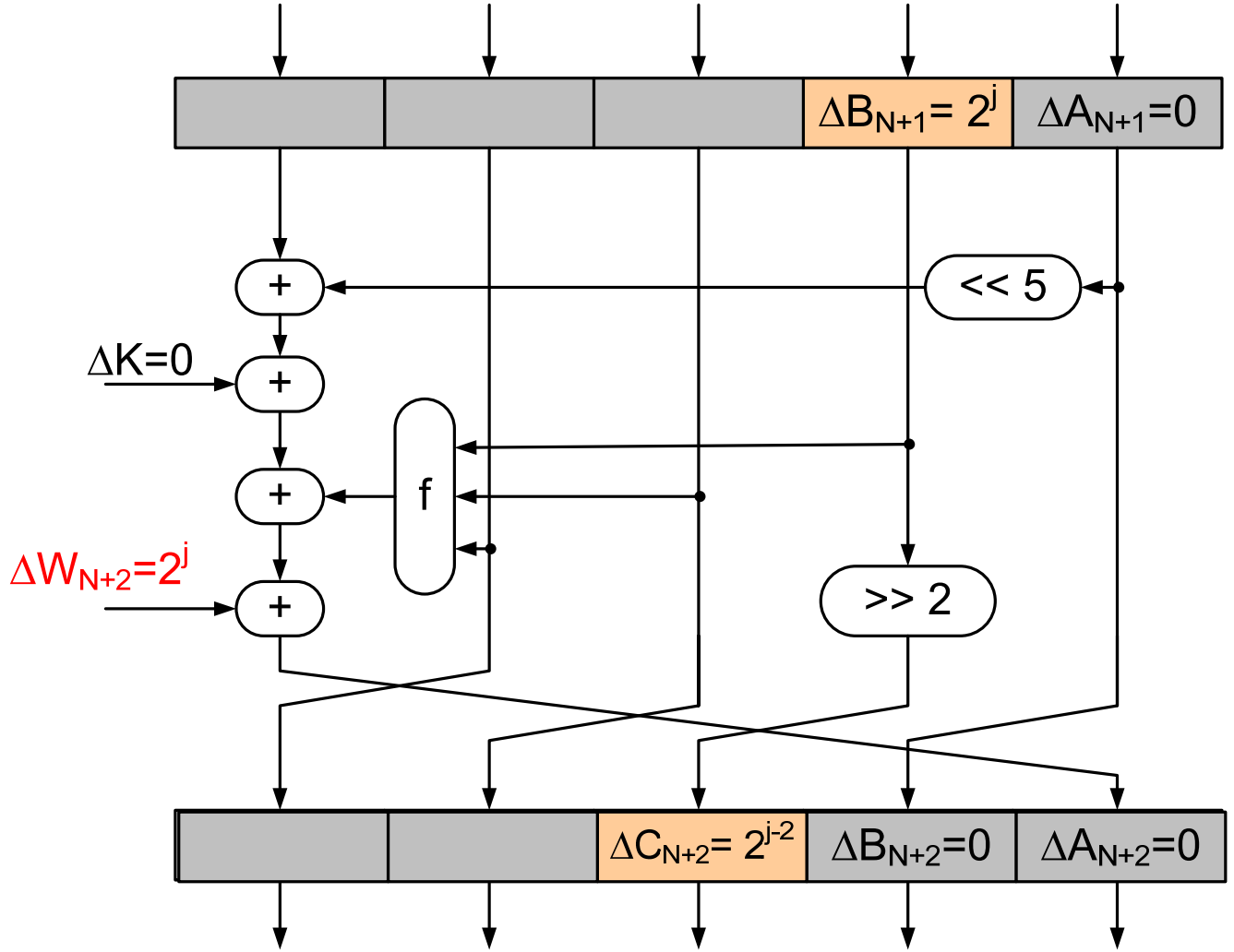
Correction 1

Step N+1



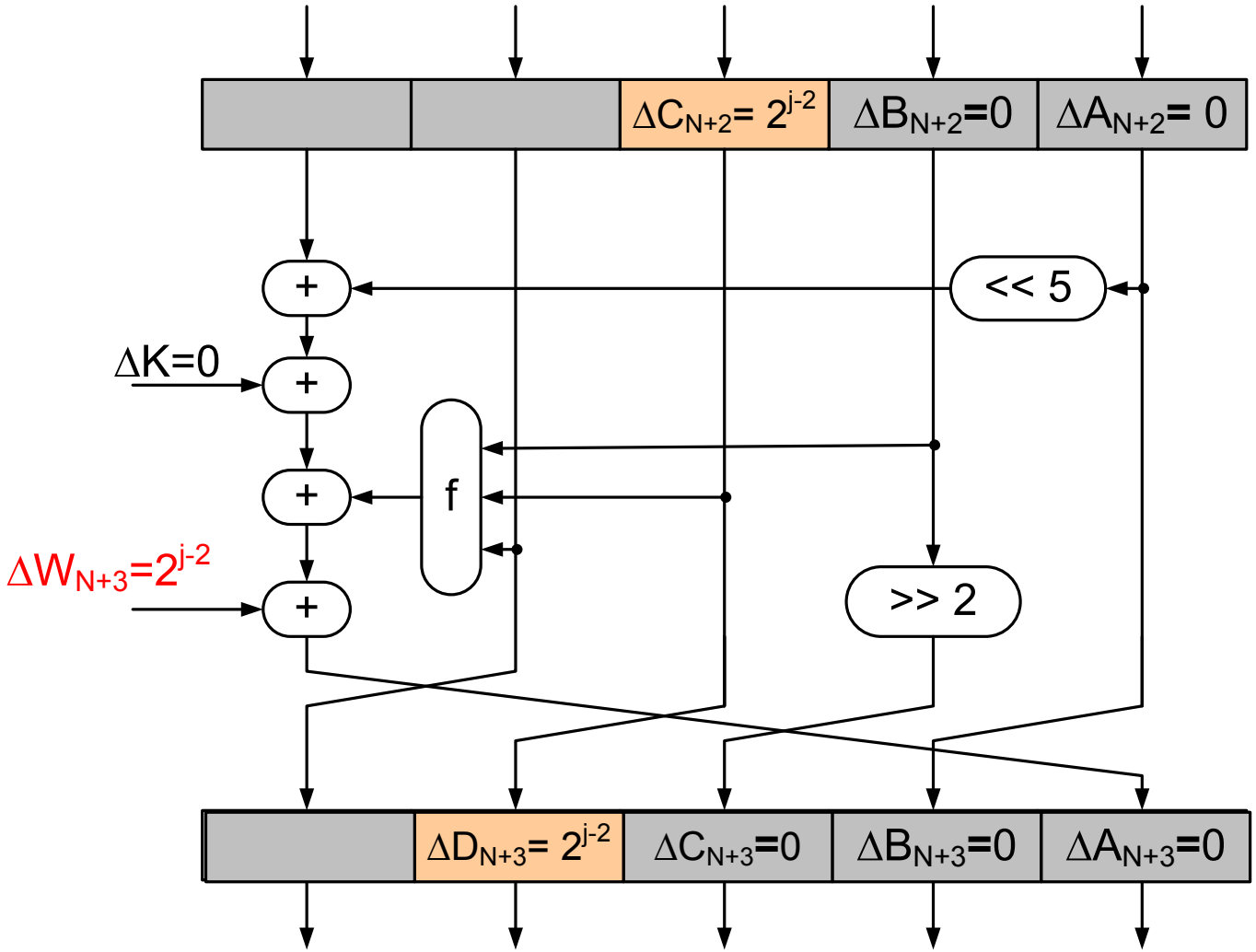
Correction 2

Step N+2



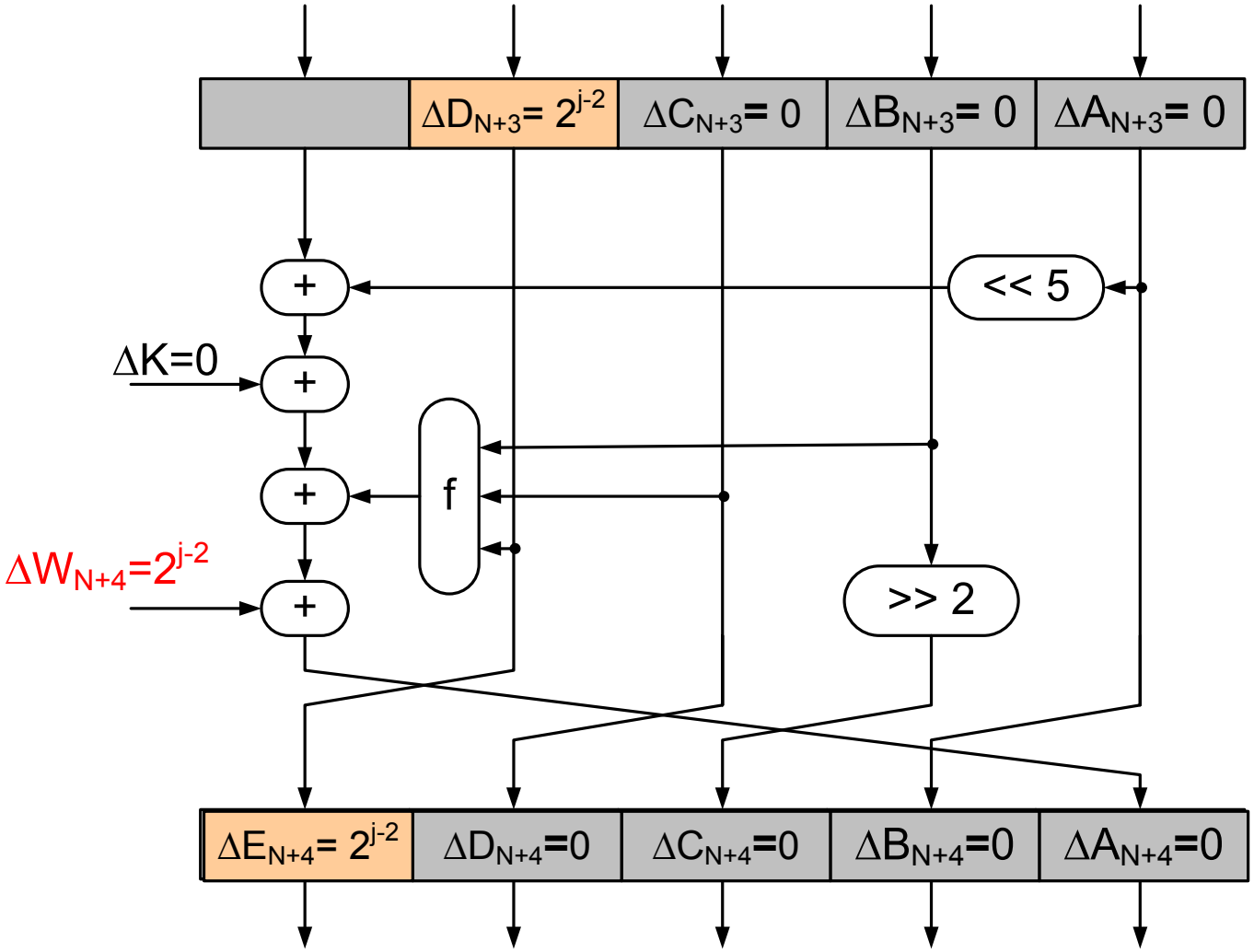
Correction 3

Step N+3



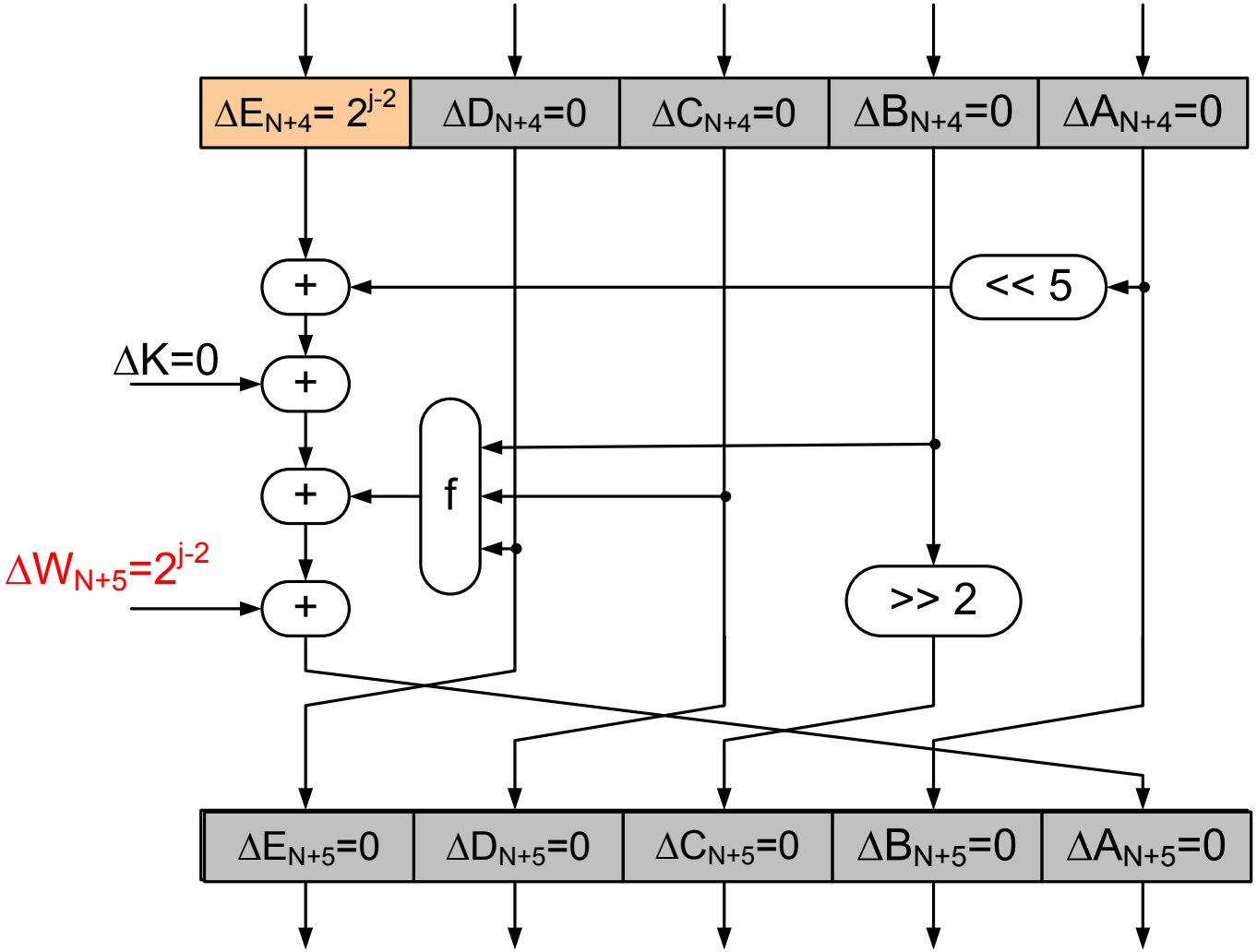
Correction 4

Step N+4



Correction 5

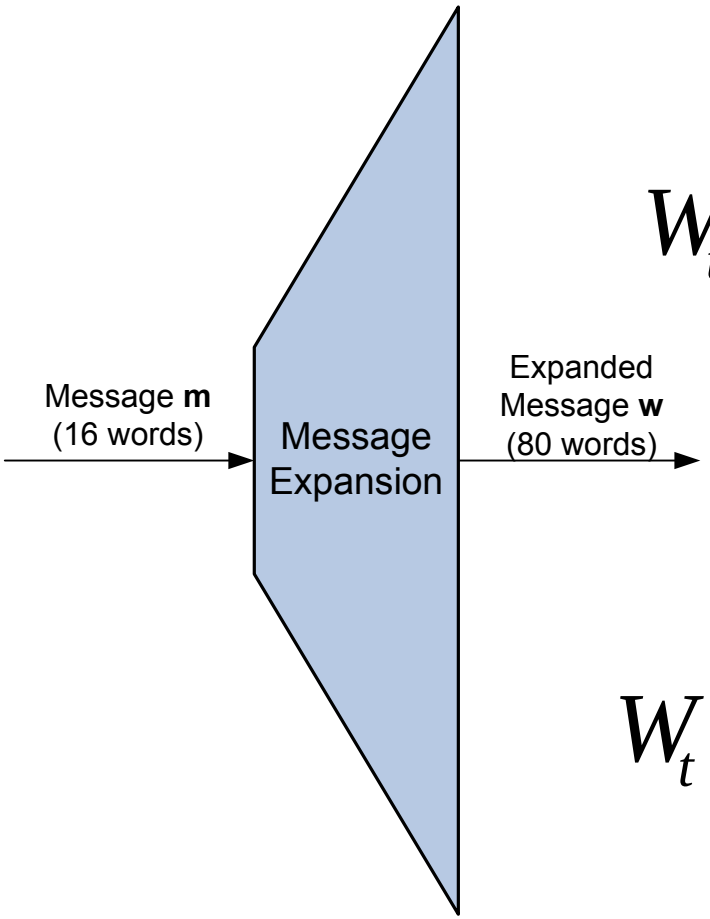
Step N+5



Local collision

- Resynchronisation of internal state
- One perturbation and 5 corrections
- Creating local collisions is not so difficult
- Problem: message expansion

Outline of SHA – Message Expansion



SHA

$$W_t = \begin{cases} M_t & for (0 \le t \le 15) \\ W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16} & for (16 \le t \le 79) \end{cases}$$

SHA-1

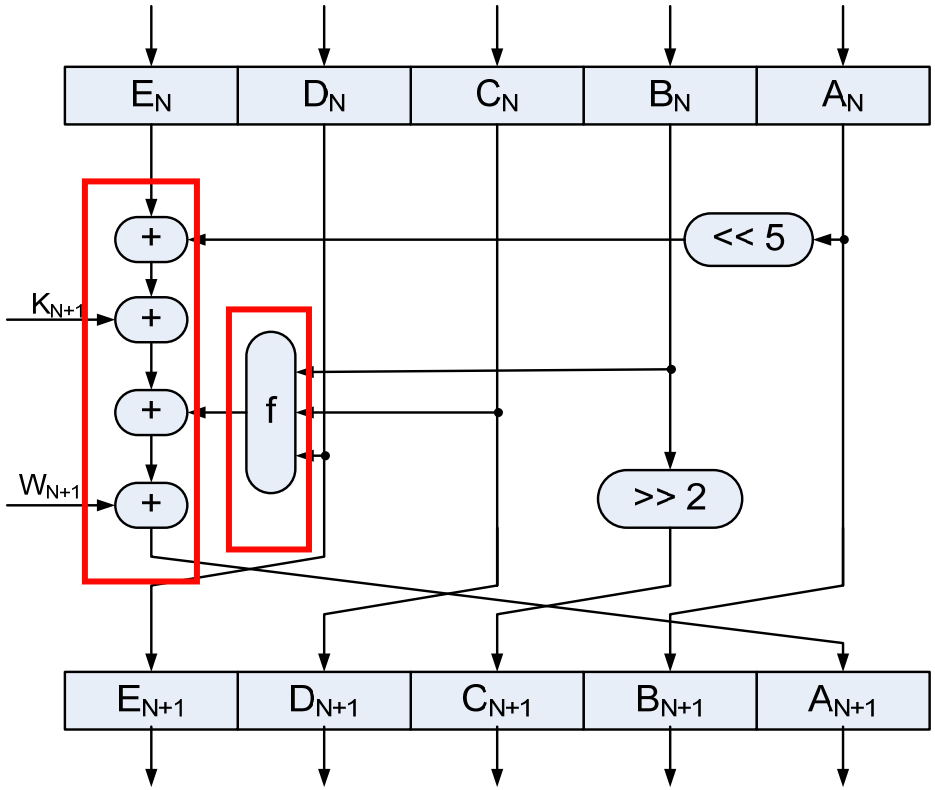
$$W_t = \begin{cases} M_t & for (0 \le t \le 15) \\ (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) <\!\!< 1 & for (16 \le t \le 79) \end{cases}$$

Message expansion

- Every bit changed influences other bits
- Impossible to find m_1, m_2 such that $E(m_1)$, $E(m_2)$ differ in 6 bits only

- Constructing a global collision = finding good characteristic

Conditions imposed by nonlinear elements



- Boolean function f
- Modular addition

Modular addition

- Linear except for carry effects
  - Carry = 0 with probability 1/2
  - Carry moves upwards only
  - No carry from MSB
- Requirement: difference propagation as with XOR

Boolean functions

- Linear in 40 out of 80 steps
- Bitwise parallel: every input bit affects 1 output bit
- We set as requirement: difference propagations as with XOR
  - High probability
  - Easy to find good characteristics

Finding good characteristic

- Linear approximation
  - Boolean functions → XOR
  - Additions → XOR
- Determine message differences that produce collision for the linear approximation
  - Set of linear equations
- Find difference resulting in low weight sequence

Linear code approach

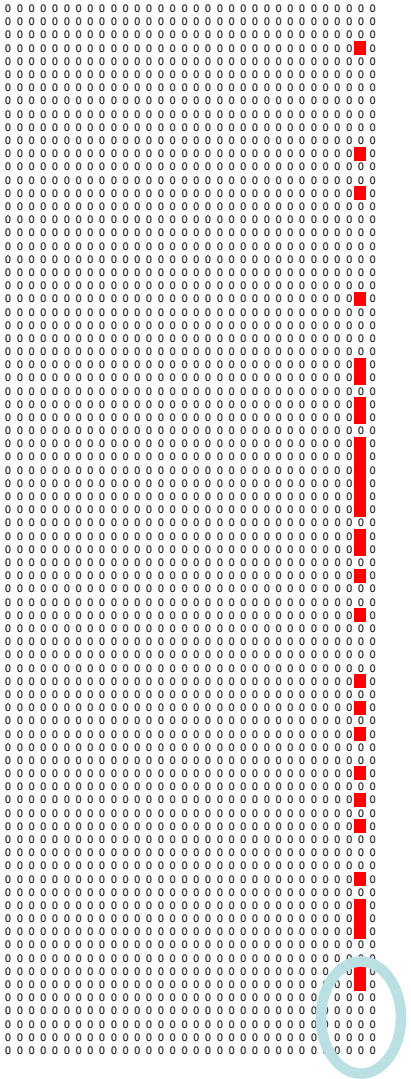
- L-SHA:
  - E: Expansion matrix
  - A, B: State update matrices

$$h = \mathrm{A}\,\mathrm{E}\,m \oplus \mathrm{B}\,iv$$

$$h_2 \oplus h_1 = \mathrm{A}\,\mathrm{E}\,(m_2 \oplus m_1) = 0$$

- A E: check matrix of linear code
- $(m_2 \oplus m_1)$: low-weight code word
  - Heuristic algorithms

Building a collision for SHA

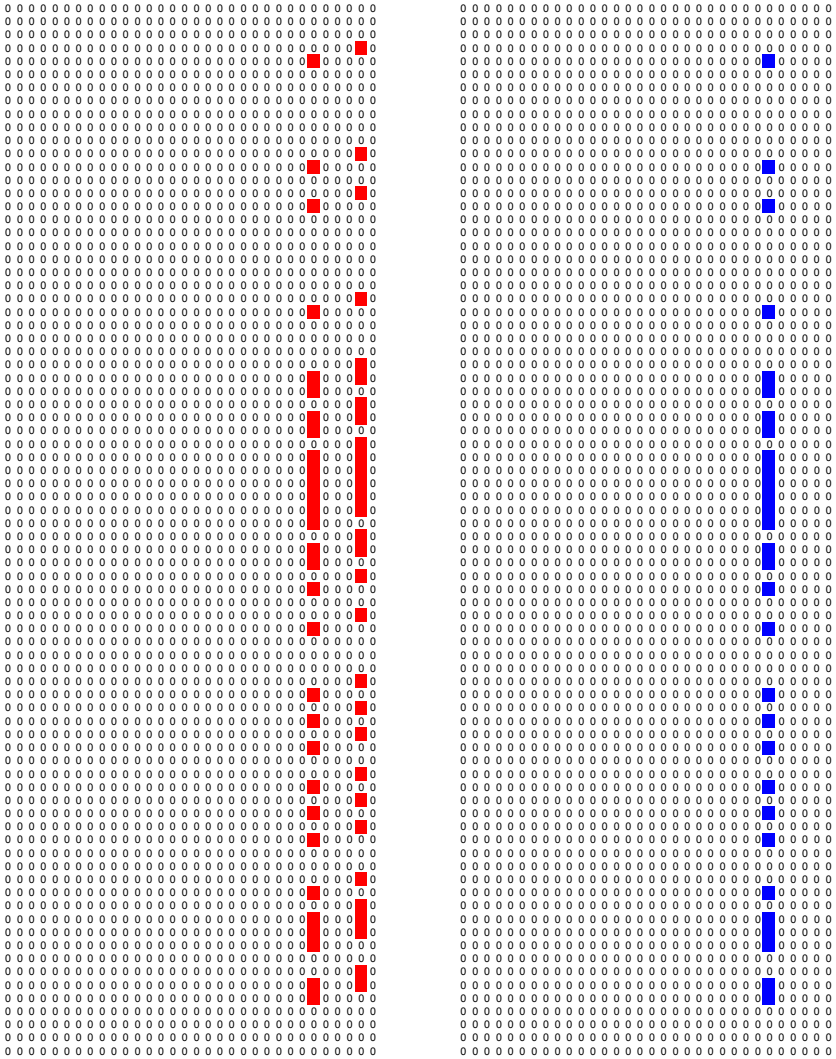


- Perturbation pattern
  - Low weight

$$W_t = W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}$$

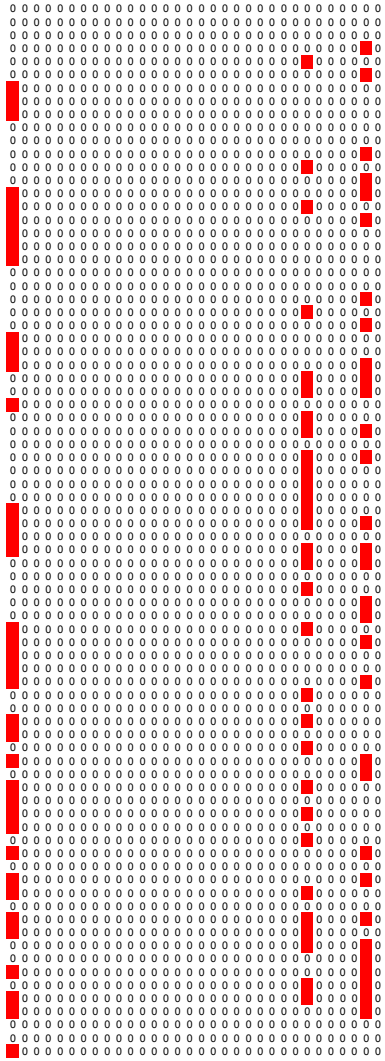
- Last 5 words are zero

A collision-producing difference pattern



- Apply 5 corrections with the same pattern
  - displaced over steps
  - rotated over bit positions

A collision-producing difference pattern



- Completed difference pattern consisting of
  - 1 perturbation pattern
  - 5 correction patterns

2. Construct right pair

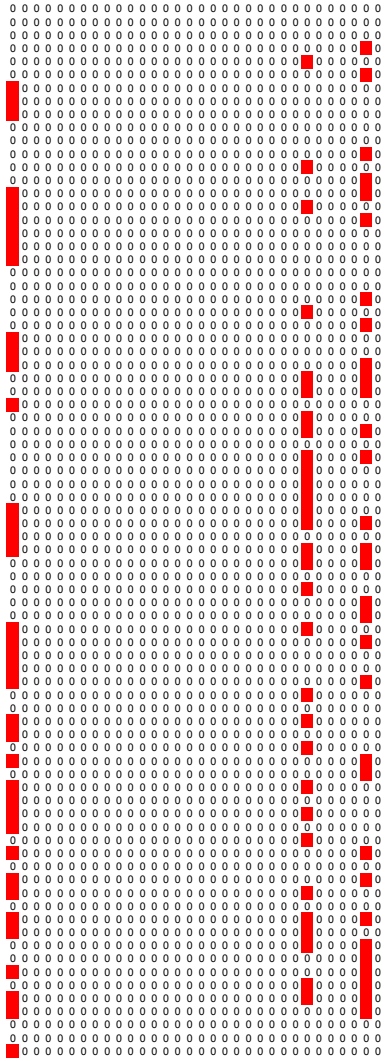
- With specified input difference
- Such that differences propagate as specified
- Conditions
  - On inputs of IF, MAJ
  - On inputs of addition

Example

$$IF(x,y,z) = xy \oplus (1 \oplus x)z$$

- Input difference: (0,0,1)
  - Desired output difference: 1
  - Condition: $x = 0$
- Input difference: (1,0,1)
  - Desired output difference: 0
  - Condition: $x \oplus z = 1$

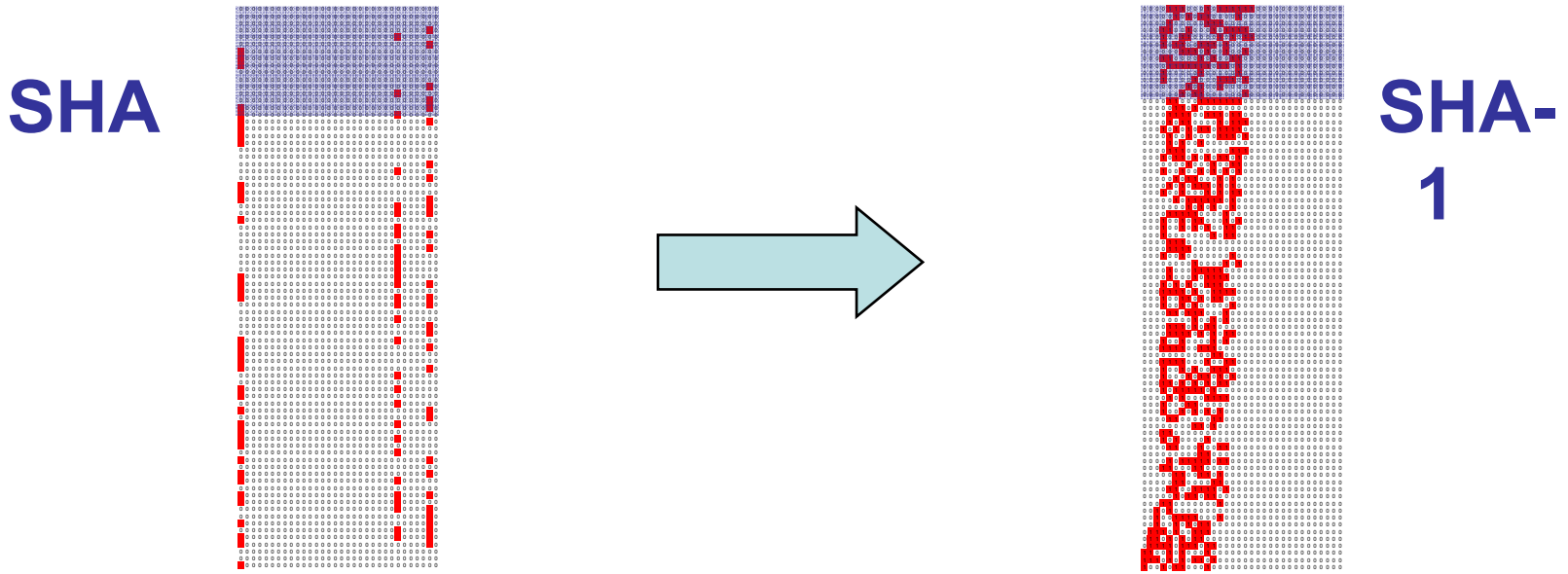
Conditions



- Every perturbation gives 2-5 conditions on the message
- Most conditions are nonlinear and complicated to express in terms of the input message
- Goal is to minimize these conditions (to make final search easier)

Application to SHA-1

- Low- weight patterns exist for SHA => break



- For SHA-1: weight is too high

Improvements

- Better ways to construct right pairs
  - Message modification
- Better characteristics
  - 1-block $\rightarrow$ multi-block
  - Better suited for hash functions

Constructing right pairs

- Equations following from nonlinear operations
  - Every step increases the complexity
- First 15 steps: easily solvable
- Next steps: mostly guess and verify
- Deterministic solving of eqs. in steps 16 and ff.
  - Neutral-bit technique [Biham,Chen]
  - Advanced message modification [Wang et al.]

Neutral bits

- Weak diffusion
  - Based mostly on carry propagation effects
- After small number of steps, not all output bits depend on all input bits
- When trying pairs, only vary bits that don't change the outputs which are already right

Advanced message modification

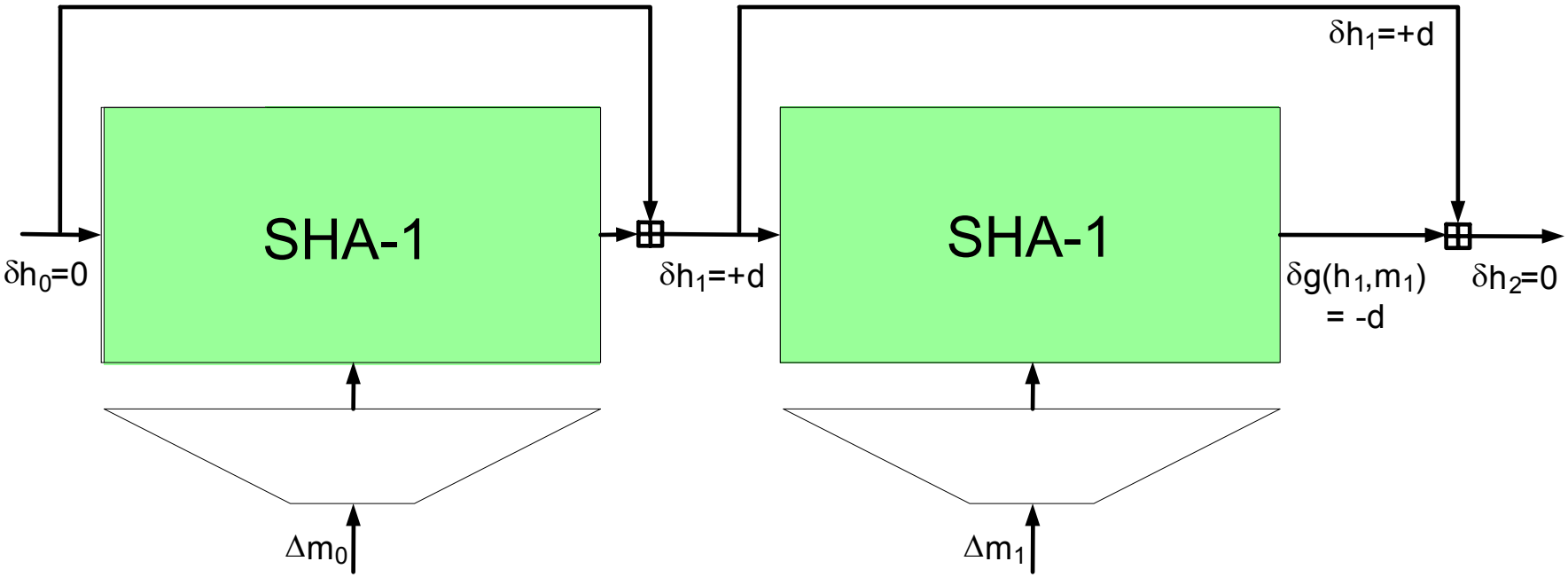
- Solve equations deterministically
  - Iterative solving strategy
- Again helped by weak diffusion

- Boundary: we need to leave some degrees of freedom for last phase

Multi-block collisions

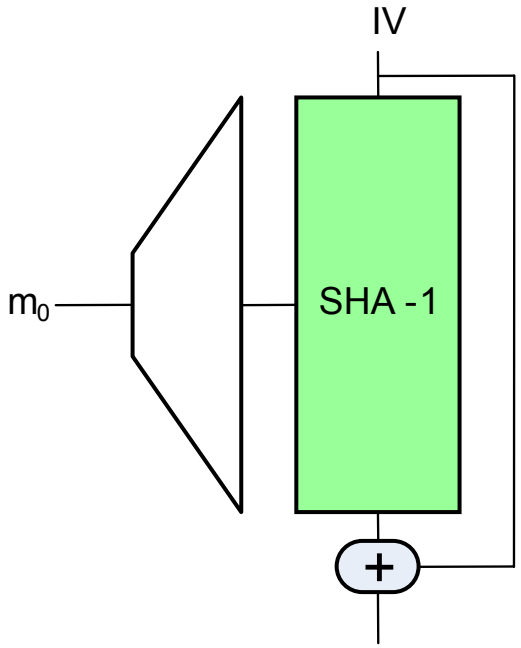
- Near-collision: outputs differ in only a few bits
- Observation: much easier than collisions
  - Characteristics with higher probability

Use of near-collisions



- Two related near-collisions give a 2-block collision
- Work effort of two blocks is only double of one block

Minimize difficult equations



First 20 steps: # conditions not so important; almost arbitrary difference propagation

Last 60 steps: minimize conditions; follow the approximation

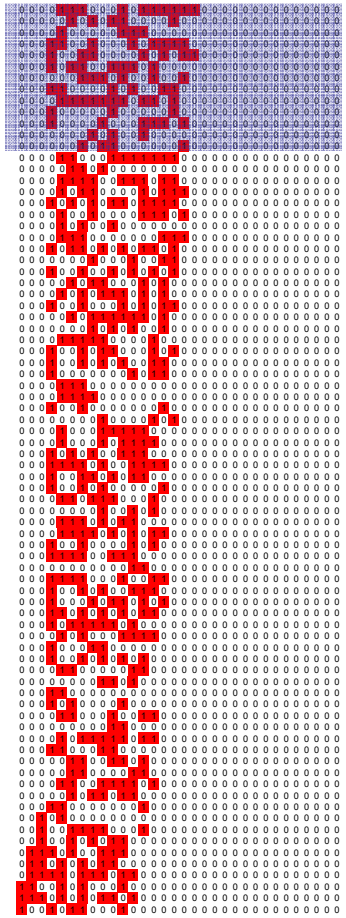
Use of pseudo-collisions

- Pseudo-collision: starting with different IVs
  - Again much more easy than (near) collisions
- Construct characteristic for pseudo-near collision over last 60 steps

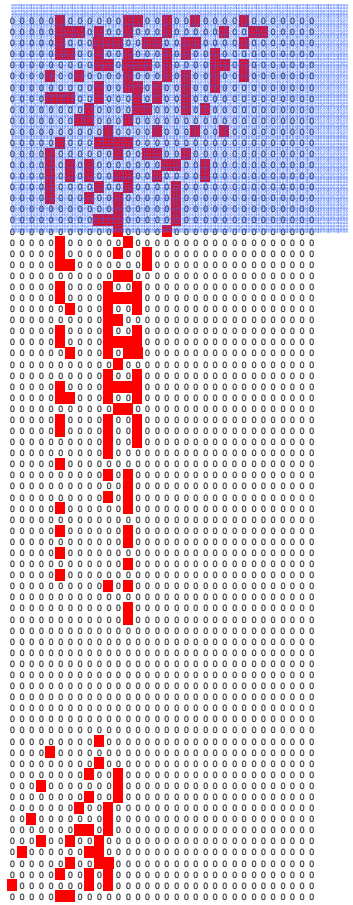
- First 20 steps:
  - From correct IV to required difference
  - Low-probability characteristic (advanced message modification)

Result for SHA-1

SHA-1 old approach



SHA-1 new approach



Final search complexity: 2^{69}

Conclusions & outlook [anno 2004]

- Collisions for SHA have been found
- Academic break of SHA-1
  - Collisions (messages without meaning)
- Attacks are optimized versions of 15-year old techniques
- Same techniques break MD4, MD5, RIPEMD, HAVAL
- What about SHA-256?