|  |
| --- |
| **Algorithm: Trade-off for DEC** |
| **Input:** W, A, b |
| R = W % A; |
| **Parameter:** R1, R2, |
| h1, h2 |
| **Output:** messageN |
|  |
| **Procedure:** |
| 1. Initialize R1 ← 0, R2 ← 0, h1 ← 0, h2 ← 0; |
| 2. **If** R ==0: output W **/**A |
| 3. **else:** |
| 4. **for** p = 0 to b **do** |
| 5. **for** s1 in {-1,1} **do** |
| 6. p1 = s1 \* p |
| 7. R1 ← **P\_LUT**[p1] |
| 8. R2 ← R - R1 |
| 9. **if** R2 == 0: output (W – s1\*2p1) **/** A |
| 10. **else if** R2 < 0**:** R2 += A |
| 11. **else:** p2 ← **R\_LUT**[R2] |
| 12. **if** p2 is found: break |
| 13. **if** found: break |
| 14. **if** found: output (W – s1 \* 2p1 – s2 \* 2p2) |
| 15. **else:** multiple-AW errors are seldom occurred |

1 Given: A, W', b # b = bit length

2 R = W' % A

3 if R == 0:

4 output W'

5 else:

6 for *i* in range(b):

7 for s1 in (-1, +1):

8 lookup h-table to find r1

9 r2 = R – r1

10 if r2 == 0: output (W' – s1\*2*i*) % A

11 else if r2 < 0: r2 += A

12 else: lookup r-table and find *j* and s2

13 if found: break

14 if found: break

15 if found: output (W' – s1\*2*i* – s2\*2*j*) % A

16 else: (multiple AW errors are seldom occurred)