Définition des variables

1. Création du msg ⬄ longueur + n\*0 + 1 + msg ⬄ n%512
   1. Conversion du texte en binaire ⬄ *m[b]*
   2. Écriture de la longueur *(b)* en binaire ⬄ nom[64]

* Si longueur sup à 264 prend que les bits de poids faible
  1. Ajout du 1 et des 0, systématique, meme si déjà modulo

On a donc M[N-1] ⬄ msg total multiple de 512 bit

* Soit M [0 ... N-1] les mots du message résultant, où N est un multiple de 16.

1. Buffer / variable : 32 bits en hexa

* A: 01 23 45 67
* B: 89 ab cd ef
* C: fe dc ba 98
* D: 76 54 32 10

1. Fonctions : 32 bits en entrée et en sortie

* F(X,Y,Z) = XY v not(X) Z ⬄ si X alors Y sinon Z
* G(X,Y,Z) = XZ v Y not(Z) ⬄ si X alors Y sinon si Y alors non Z
* H(X,Y,Z) = X xor Y xor Z ⬄ XOR : X et Y et Z
* I(X,Y,Z) = Y xor (X v not(Z)) ⬄ XOR : Y et X sinon non Z

1. Table : sur 64 éléments entier

* I en radian

Calcul

1. Boucle a faire pour chaque bloc de 512 bits
   1. Boucle à faire 16 fois ⬄ 32 bits chacun **pas sure**
      1. Copier dans X
   * For j = 0 to 15 do
   * Set X[j] to M[i\*16+j].
     1. Copier les valeurs des variables
   * AA / BB / CC / DD
     1. Round 1 [abcd k s i] :
   * [ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4]
   * [ABCD 4 7 5] [DABC 5 12 6] [CDAB 6 17 7] [BCDA 7 22 8]
   * [ABCD 8 7 9] [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]
   * [ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]
     1. Round 2 [abcd k s i] :
   * [ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]
   * [ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]
   * [ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]
   * [ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]
     1. Round 3 [abcd k s i] :
   * [ABCD 5 4 33] [DABC 8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]
   * [ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]
   * [ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]
   * [ABCD 9 4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA 2 23 48]
     1. Round 4 [abcd k s i] :
   * [ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]
   * [ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]
   * [ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]
   * [ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]
     1. Calculs des buffer
        1. A = A + AA
        2. B = B + BB
        3. C = C + CC
        4. D = D + DD
   1. , we begin with the low-order byte of A, and end with the high-order byte of D.