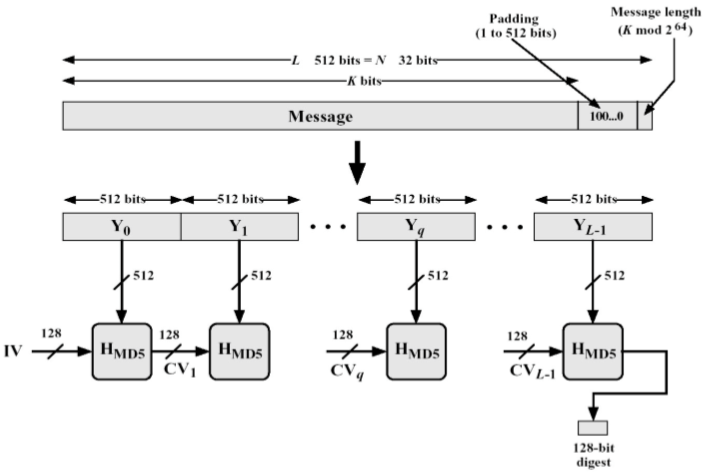
**MD5算法的程序设计和实现——实验报告**

1. **算法原理概述**

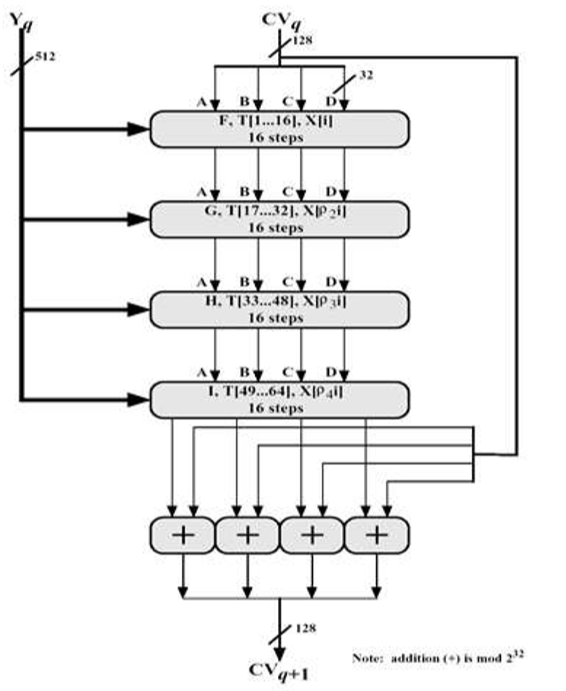
MD5即Message-Digest Algorithm5(信息-摘要算法5)，由Ron Rivest发明，是广泛使用的Hash算法，用于确保信息传输的完整性和一致性。MD5使用little-endian(小端模式)，输入任意不定长度信息，以512-bit进行分组，每一分组又被划分为16个32位子分组，经过了一系列的处理后，生成四个32-bit的数据，最后联合输出固定128-bit的信息摘要。MD5算法的基本过程为：填充、分块、缓冲区初始化、循环压缩、得出结果。

1. **算法总体结构**

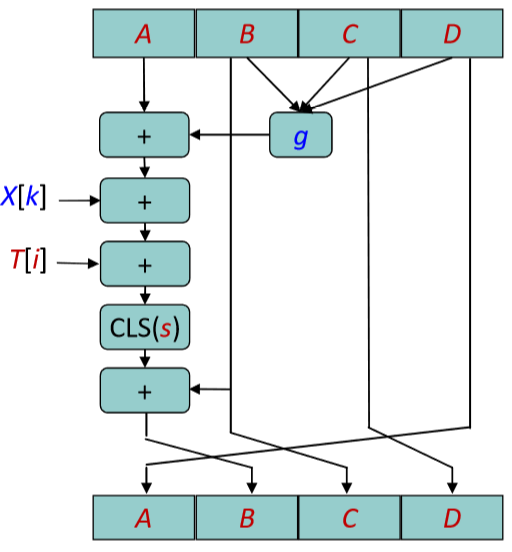
**MD5算法总流程**



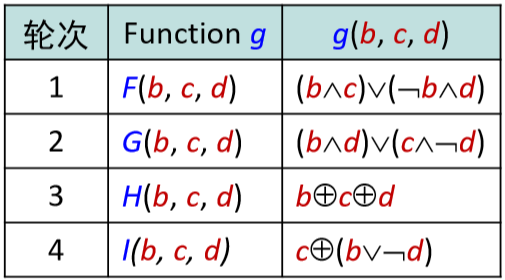
**循环压缩总体结构**



**循环压缩每轮循环中一次迭代运算逻辑**



**4轮循环所使用的生成函数**



**三、模块分解**

**①填充字符串函数：**在长度为K bits的原始消息尾部填充长度为P bits的标识（1000....00）,其中1 <= P <= 512（即至少需要填充一个bit），使得填充后的消息位数为 K+P ≡ 448（mod 512）。注意，当K ≡ 448（mod 512）时，填充的字节数 P = 512 bit。填充得到上述消息后，在尾部附加K值的低64位，最后得到一个长度为 K+P+64 ≡ 0（mod 512）的消息。

**②字符串分块函数：**将填充好的字符串分割成L个长度为512bit的分组

**③循环压缩函数：**对每个512-bit分组进行64轮迭代运算

1. **对分组（A,B,C,D）中的A进行迭代运算**

**公式为：A <= B + ((A + g(B,C,D) + X[k] + T[i])) << S[i]**

**其中：**

**·** A,B,C,D代表MD5缓冲区当前的数值

· g为轮函数，1-16轮迭代使用F函数，17-32轮迭代使用G函数，33-48轮迭 代使用H函数，49-64轮迭代使用I函数

· X[k]代表当前处理消息分组的第k个32位字，X[k]由第n轮迭代对应的顺序表决定

· T[i]代表T表的第i项的值，T[i] = int(2^32 \* |sin(i)|)

· S[i]对应第i轮的左循环移位的s值

1. **对分组（A,B,C,D）作循环轮换**

**公式为：（B,C,D,A）<=（A,B,C,D）**

**④MD5编码函数：**用于调用前面的功能函数进行MD5编码

1. 输入待加密的明文字符串
2. 对明文字符串进行填充
3. 对填充后的明文字符串进行分块（Yq）
4. 使用预设的初始值初始化MD5缓冲区间（IV）
5. 对各个分块字符串利用公式HMD5(CVi-1, Yi)进行循环压缩，运算结果作为下一块的输入（CVi）
6. 当所有的分块迭代完成后，输出结果CVL，L表示最后一个分块的序号

**四、数据结构**

程序的输入使用C++的string类型，在MD5的编码过程中将字符串转化成类型为32位unsigned int数组，每个分块使用长度为16的unsigned int数组，循环压缩运算过程中所有的操作都是基于unsigned int类型。最后输出时将unsigned int数组中的数据按照16进制格式输出成长度为32的字符串。

1. **编译运行结果**

|  |  |
| --- | --- |
| **本地测试** |  |
| **网上比对** |  |
| **本地测试** |  |
| **网上比对** |  |
| **结果分析** | 32位加密结果输出完全一样，MD5算法成功实现。 |