**SHA-512**

**步骤1：填充附加位**

填充消息使其长度≡896（mod 1024）。填充由一个1 和后续的0组成。

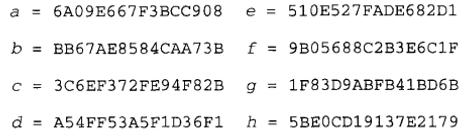
**步骤2：附加长度**

　　在填充后的消息后附加128位的块，将其视为无符号整数，它包含前消息的长度。

　　前两步的结果是产生了一个长度为1024整数倍的消息，以便分组。

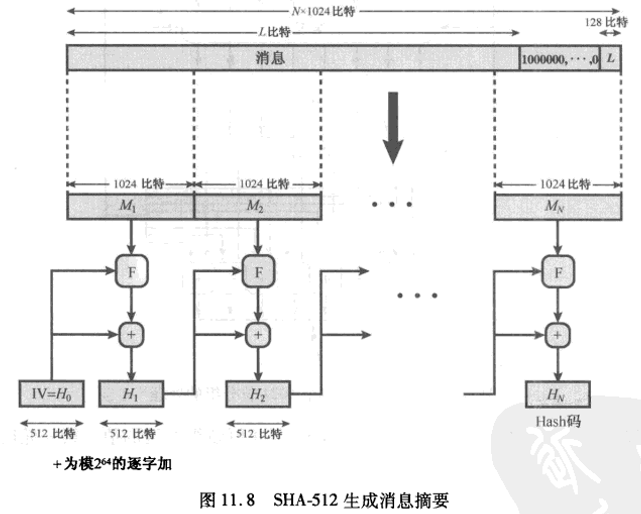
**步骤3：初始化Hash缓冲区**

　　Hash函数中间结果和最终结果保存在512位的缓冲区，缓冲区由8个64位的寄存器（a、b、c、d、e、f、g、h）表示，并将这些寄存器初始化为下列64位的整数（十六进制）



　　每个寄存器内容获取的方式是：取前8个素数（2、3、5、7、11、13、17、19）取平方根，取小数部分的前64位。

**步骤4：以1024位分组（128个字节）为单位处理消息并输出结果**

****

　　总结SHA-512的运算如下：

　　　　H0 = IV

　　　　Hi = Hi-1+F(Hi-1,Mi)

　　　　MD = HN

　　IV是第三步中定义的abcdefgh缓冲区的初始值

　　Hi是第i个消息分组处理的最后一轮的输出

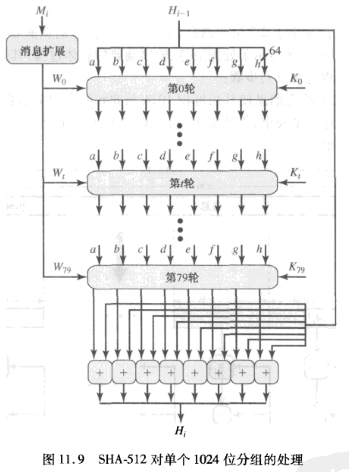
　　N为消息（包括填充和长度域）中的分组数

　　MD表示最后的消息摘要值

　　F表示轮函数

 　　+为模264位加

**轮函数**的核心是具有80轮运算的模块，在图11.8中，该模块标记为F。下图是它的逻辑原理。



　　Kt是轮常数，每一轮的轮常数均不相同，用来使每轮的计算不同。这些常数获得方法如下：对前80个素数开立方根，取小数部分前64位。这些常数提供了64位随机串集合，可以初步消除输入数据中的统计规律。

　　对分组Mi进行消息扩展生成Wt，每个Wt64位，前16个Wt直接取自当前分组。余下的值按如下方式导出

　　　　Wt= δ1512(Wt-2) + Wt-7 + δ0512(Wt-15) + Wt-16

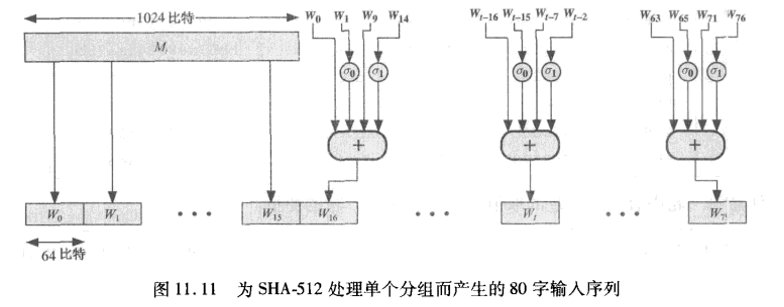
　　　　　　δ1512(x) = ROTR1(x)⊕ROTR8(x)⊕SHR7(x)

　　　　　　δ0512(x) = ROTR19(x)⊕ROTR61(x)⊕SHR6(x)

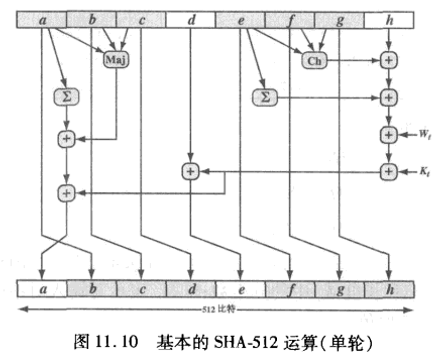
　　　　　　　　ROTRn(x)为对64位的变量x循环右移n位

　　　　　　　　SHRn(x)为对64位变量x向左移n位，右边填充0

　　　　　　　　+为模264位加



　　轮函数每一轮操作如下图



　　从上图可以看出，轮函数有两个特点：

　　　　1、轮函数输出的8个字中的6个是通过简单的轮置换实现的。如上图的阴影部分。

　　　　2、输出中只有2个字通过替代操作产生。

　　　　　　字e是将输入变量(d,e,f,g,h)以及轮常数Kt和轮消息Wt作为输入的函数。

　　　　　　字a是将除d之外的输入变量以及轮常数Kt和轮消息Wt作为输入的函数。

　　　　　　　　T1 = h + Ch(e,f,g) + (∑1512e) + Wt + Kt

　　　　　　　　T2 = (∑0512a) + Maj(a,b,c)

　　　　　　　　e = d + T1

　　　　　　　　a = T1 + T2

　　　　　　其中：

　　　　　　　　t为步骤数，0≤t≤79

　　　　　　　　Ch(e,f,g) = (e AND f)⊕(NOT e AND g)  　　　　　　条件函数（位运算）：如果e，则f，否则g

　　　　　　　　Maj(a,b,c) = (a AND b)⊕(a AND c)⊕(b AND c)　　  当且仅当变量的多数（2个或者3个）为真时函数为真

　　　　　　　　(∑0512a) = ROTR28(a)⊕ROTR34(a)⊕ROTR39(a)

　　　　　　　　(∑1512e) = ROTR14(e)⊕ROTR18(e)⊕ROTR41(e)

SHA-512算法具有如下特性：Hash码的每一个位都是全部输入位的函数。基本函数F多次复杂重复运算使得结果充分混淆，从而使得随机选择两个消息，甚至于这两个消息有相似的特征，都不太可能产生相同的Hash码。除非SH-512中存在目前未公开的隐藏缺陷，找到两个具有相同摘要的消息的复杂度需要2256次操作，给定摘要寻找消息的复杂度需要2512次操作。