[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 7/72 (2006.01)

[12] 发明专利说明书

专利号 ZL 200510028915.4

[45] 授权公告日 2008 年 11 月 19 日

[11] 授权公告号 CN 100435090C

[22] 申请日 2005.8.18

[21] 申请号 200510028915.4

[73] 专利权人 上海微科集成电路有限公司

地址 200433 上海市国定路 335 号 5005 室

共同专利权人 复旦大学

[72] 发明人 曾晓洋 麻永新 范益波 顾叶华 陈 俊 郭亚炜

[56] 参考文献

US5742530A 1998.4.21

一种 Montgomery 模乘的硬件算法及其实现. 方颖立，高志强. 微电子学，第 32 卷第 4 期. 2002

一种新型硬件可配置公钥制密码协处理器的 VLSI 实现. 陈超，曾晓洋，章倩苓. 通信学报，第 26 卷第 1 期. 2005

审查员 徐 春

[74] 专利代理机构 上海正旦专利代理有限公司

代理人 陆 飞 盛志范

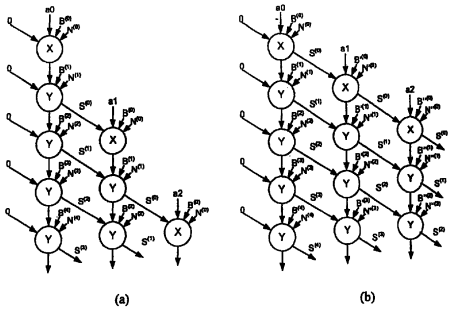
权利要求书 1 页 说明书 6 页 附图 2 页

[54] 发明名称

可扩展高基蒙哥马利模乘算法及其电路结构

[57] 摘要

本发明属集成电路技术领域，具体为一种可扩展高基蒙哥马利模乘算法及其电路结构。 本发明是对多字高基蒙哥马利模乘器的改进，其中，每一步对模数 N 和被乘数 B 进行左移位操作，对中结果 S 不作移位操作，使数据通路的流水线级间的延迟从二个时钟周期缩短为一个时钟周期。 其电路结构包括用于存放模乘运算 3 个操作数 A、B 和 N 的 3 个存储器、由第 1—第 P 级处理单元组成的流水线形式的数据通路模块、用于控制整个模乘器运算过程的控制模块和一个先进先出的存储器等。 本发明大大提高了模乘运算速度，同时对中间结果的存储单元进行了改进，使其硬件开销减小。



1、一种可扩展高基蒙哥马利模乘算法，其特征在于以多字高基蒙哥马利模乘算法为基础，每一步对模乘的被乘数 B 和模数 N 进行左移位操作，对中间结果 S 不作移位操作，从而使数据通路的流水线级间的延迟由 2 个时钟周期减小到一个时钟周期，具体步骤如下：

首先模数 $N^{(0)}$、被乘数 $B^{(0)}$以及 k+1 位的 a0 进入第一级流水线，经过一个时钟周期运算得到和结果 $SS^{(0)}$和进位 $SC^{(0)}$；第二个时钟周期将 $N^{(0)}$和 $B^{(0)}$左移 k 位，还有 $SS^{(0)}$和 $SC^{(0)}$传递到第二级流水线进行运算，同时 $N^{(1)}$、$B^{(1)}$进入第一级流水线进行运算；这样经过 p 级 PE 的流水线运算，中间运算结果从第 p 级的 PE 输出；下面分两种情况，如果 $e > p$，中间结果从第 p 级 PE 输出时，第 1 级 PE 仍然在运算高位的 $N^{(i)}$和 $B^{(i)}$,所以将输出的中间结果通过 CLA 加和后存入 FIFO，直到 $N^{(e-1)}$和 $B^{(e-1)}$通过第 1 级 PE 运算后，再将 FIFO 中的中间结果依次读入流水线第 1 级 PE 开始运算；如果 $e \le p$,则当中间结果从第 p 级 PE 输出时，第 1 级 PE 已经空闲，所以输出的中间结果 $SS^{(0)}$和 $SC^{(0)}$通过移位处理后直接进入第 1 级 PE 进行运算；这里，PE 为处理单元，N 为模数，B 为被乘数，SS 为和结果，SC 为进位结果，CLA 为提前进位加法器，FIFO 为先进先出存储器模块；这里，e=n/w，n 为模数 N 的倍数，w 为处理单元的数据宽度。

2、一种可扩展高基蒙哥马利模乘器电路结构，其特征在于包括：3 个存储器模块：存储器(5、6、7)，用来存放模乘运算的 3 个操作数 A、N 和 B；第 1-第 P 级处理单元组成的流水线形式的数据通路模块；控制模块(10)以及一个先进先出的存储器 (9)；模乘运算时，模数 N 和被乘数 B 分别以 w 位宽的 $N^{(i)}$和 $B^{(i)}$进入数据通路参与运算，同时乘数 A 以 k+1 位的宽度进入各个处理单元；第 p 级处理单元(2)输出的中间结果进入 FIFO(9)；FIFO(9)的输出结果和 2w 位的零通过一个选择器(8)进入第一级处理单元(4)；控制模块(10)控制整个模乘器的运算过程，包括三个存储器(5、6、7)的读写、数据通路中数据的流向、FIFO(9)中的数据读写。

3、根据权利要求 2 所述的模乘器电路结构，其特征在于先进先出存储器模块(9)由输出寄存器(11)和寄存器(15)、两输入与门(12)、两输入选择器(13)、寄存器堆(14)和提前进位加法器(16)组成，当 bypass 信号为高时，进位结果 SC 通过选择器(13)直接输出到输出寄存器(11)，和结果 SS 通过一个两输入与门(12)输出到输出寄存器(11)；当 bypass 信号为低时，和结果 SS 和进位结果 SC 进入提前进位加法器(16)相加后再通过寄存器(15)进入寄存器堆(14)，然后寄存器堆的输出再通过两输入选择器(13)输出到输出寄存器(11)。

4、根据权利要求 2 所要求的模乘器电路结构，其特征在于其数据通路内的处理单元，采用触发器(17、18)和寄存器(19)对被乘数 B 和模数 N 进行左移 2 位后再传递到下一级处理单元。

可扩展高基蒙哥马利模乘算法及其电路结构

技术领域

本发明属集成电路技术领域，具体涉及一种改进的可扩展高基蒙哥马利（Montgomery）模乘算法及其电路结构。

背景技术

随着电子通讯技术的飞速发展，信息安全越来越受到人们的关注。为了保障传输数据的安全，人们提出了很多密码算法与协议。因为用软件实现加密算法既耗时又存在安全隐患，所以近年来 RSA 等密码算法的硬件实现成为信息安全技术研究的一个热点，众多的国内外学者在这方面已经取得很多研究成果，并且已有很多成果应用于各种信息安全产品中。

有限域的乘法运算广泛应用于各种加密算法中，如 RSA 算法，椭圆曲线密码算法（ECC）等。因为随着密钥长度的增加和需要大量的模乘运算来实现模幂运算，加密运算越来越耗时，所以如何用硬件实现高效率的模乘器成为密码处理器设计的关键。Montgomery 算法仅仅通过加法运算和移位操作实现模乘运算，避免了除法运算，因此很适合硬件实现。

Montgomery 模乘是一种利用整数的余数表示系统（Residue Number System，RNS）来求模乘的方法，通过操作数到 RNS 的变换，在 RNS 除法求模转化为每次扫描乘数后的移位操作，最后再从 RNS 变换回整数，实现模乘运算。下面对 Montgomery 模乘算法进行介绍。

算法 1. Montgomery 模乘算法

Montgomery 模乘：$MM(A, B, N) = A \times B \times R^{-1} \pmod N$，式中，N 为 n 位，A、B 也是 n 位且小于 N，$R=2^n$，其算法如下：

输入：A，B，N

输出：S

MM Algorithm:

S = 0

for i = 0 to n-1

$q_i = (S + A_i \times B) \bmod 2$

$S = (S + q_i \times N + A_i \times B)/2$

end for

if S ≥N then S=S-N

算法中 $q_i = (S+A_iB) \bmod 2$，由 S、A_i 和 B 三者的最低位决定，它的引入是为了使得累加结果的最低位为 0，从而在进行运算 $S=(S+ q_i \times N + A_iB)/2$ 时，也就是要右移一位的时候不会带来误差。从上面的算法可以看出只需要做加法和移位运算就可以得到模乘结果，非常适合硬件实现。

加密算法的模乘运算的操作数都比较大，目前对于 ECC 算法，密钥长度从 128 位到 256 位，对于 RSA 算法，密钥长度则从 512 位到 2048 位，甚至更高位数。目前大部分模乘运算器的硬件设计都是针对固定的密钥长度，也就是说模乘操作数不能超出一个固定位数。为了使同一个电路结构可以根据需要完成任意位宽要求的模乘运算，有参考文献提出了一种采用基于字操作的 Montgomery 模乘算法。同时为了提高模乘运算的速度，有文献提出了高基的 Montgomery 模乘算法，即对于被扫描的乘数 A 每次扫描一位以上，运用布思编码（Booth encoding）进行计算。下面算法是采用多字操作的高基 Montgomery 模乘算法。

算法 2. 多字高基的 Montgomery 模乘算法

输入：A，B，N

输出：S

MWR2^{k}MM Algorithm:

$S = 0$

$a_{-1} = 0$

for j = 0 to n-1 STEP k

$q_{Bj} = Booth(a_{j+k-1..j-1})$

$(C_a, S^{(0)}) = S^{(0)} + (q_{Bj} * B)^{(0)}$

$q_{Nj} = S^{(0)}_{k-1..0} * (2^k - N^{(0)-1}_{K-1..0}) \bmod 2^k$

$(C_b, S^{(0)}) = S^{(0)} + (q_{Nj} * N)^{(0)}$

for i=1 to e-1

$(C_a, S^{(i)}) = C_a + S^{(i)} + (q_{Bj} * B)^{(i)}$

$(C_b, S^{(i)}) = C_b + S^{(i)} + (q_{Nj} * N)^{(i)}$

$S^{(i-1)} = (S^{(i)}_{k-1..0}, S^{(i-1)}_{w-1..k})$

end for

$C_a = C_a$ or C_b

$S^{(e-1)} = sign\ ext(C_a, S^{(e-1)}_{w-1..K})$

end for

if S ≥N then S=S-N

其中模数N的位数为n位，w为处理单元的数据宽度，n=e*w。有关多字高基Montgomery模乘算法详细内容可参考A. F. Tenca , G. Todorov, and C. K. Koc, "High-radix design of a scalable modular multiplier" in Cryptographic Hardware and Embedded Systems –CHES 2001, C. K. Koc and C. Paar, Eds. 2001,Lecture Notes in Computer Science, No.1717, pp.189-206, Springer, Berlin, Germany。

上述的多字高基Montgomery模乘算法的问题在于，如果采用多级处理单元的流水线电路结构，因为$S^{(i-1)}$传递到下一级运算必须等到$S^{(i)}_{k-1..0}$计算出来，因此数据从上一级流水线传递到下一级流水线需要经过两个时钟周期的延迟，这导致模乘运算的速度降低。

发明内容

本发明的目的是提出一种改进的多字高基的可扩展 Montgomery 模乘算法及其电路结构，使流水线间的延迟只有一个时钟周期，从而提高模乘运算的速度，同时对中间结果的存储单元进行改进使其硬件开销减小。

上文提到的A.F.Tenca设计的模乘器结构是每次对中间结果S进行右移操作，这样上一级流水线的一组结果$S^{(i-1)}$要等到$S^{(i)}_{k-1..0}$计算出来以后才可以传递到下一级参与运算，因此这样设计的流水线之间的延迟为两个时钟周期。当流水线级数较多时这种结构的模乘运算速度会大大降低。本发明提出的可扩展高基蒙哥马利模乘算法，是对上述多字高基蒙哥马利模乘算法(算法2)的改进，其作法是每一步对模数N和被乘数B进行左移位操作，对中间结果S不作移位操作，这样改进了流水线组织形式，使数据通路的流水线级间的延迟只有一个时钟周期，因此可以大大提高运算速度。

有关A.F.Tenca和本发明的流水线算法设计的比较如附图1所示。由图1可以看出，由于本发明的流水线的组织是左移B和N，因此前一级流水线计算得到的$S^{(i-1)}$直接进入下一级流水线进行运算而不再需要等待移位，所以本发明的流水线的组织形式可以大大提高模乘运算速度。

本发明提出的模乘乘法器的电路结构如附图2所示，主要包括：3个存储器（RAM）模块：存储器5、存储器6和存储器7，用来存放模乘运算的3个操作数A、N和B；第1-第P级处理单元(PE)组成的流水线形式的数据通路(Datapath)模块1；控制模块10(Control unit）以及一个先进先出的存储器（FIFO）9。存储器6存放模数N，存储器7存储被乘数B，存储器5存储乘数A。模乘运算时，模数N和被乘数B分别以w位宽的$N^{(i)}$和$B^{(i)}$进入数据通路参与运算，同时乘数A以k+1位的宽度进入各个处理单元。第p级处理单元2输出的中间结果进入FIFO9。FIFO9的输出结果和2w位的零通过一个选择器8进入第一级

处理单元 4。控制模块 10 控制整个模乘器的运算过程，包括三个存储器 5、6、7 的读写、数据通路中数据的流向、FIFO9 中的数据读写。

为了减小关键路径延时，运算单元内采用进位保留加法器（CSA），所以计算中间结果都是冗余形式表示，即加法的结果以和结果（SS）和进位结果(SC)存在。为了减小存储中间结果的 FIFO, 本发明将最后一级运算单元计算输出的和结果 SS 和进位结果 SC 通过一个提前进位加法器(CLA)16 加和后再存入 FIFO9，这样 FIFO9 的大小可以减小一半，从而减小了硬件开销。FIFO 电路结构如附图 3 所示，其组成包括：一个提前进位加法器 16、一个寄存器 15、一个寄存堆 14、一个选择器 13、一个两输入与门 12 和一个输出寄存器 11。

当 bypass 信号为高时，进位结果 SC 通过选择器 13 直接输出到输出寄存器 11，和结果 SS 通过一个两输入与门 12 输出到输出寄存器 11。当 bypass 信号为低时，和结果 SS 和进位结果 SC 进入提前进位加法器 16 相加后再通过寄存器 15 进入寄存器堆 14，然后寄存器堆的输出再通过两输入选择器 13 输出到输出寄存器 11。

以基 4 为例的处理单元（PE）如图 4 所示，其中取运算位宽为 32 位。处理单元主要包括 Aj 编码器模块 31 和 N 编码器模块 28，两个进位保留加法器 27 和 34，两个反相器模块 21 和 32，若干寄存器、选择器和触发器。被乘数 B 和模数 N 通过触发器 17 和 18 左移两位，再通过寄存器 19 输出到下一级处理单元。乘数 Aj 通过 Aj 编码模块 31 后产生信号 double、zero 和 neg，由 double、zero 和 neg 控制三输入选择器 30 和反相器 32 产生相应的被乘数的倍数。输入 SC、SS 与反相器 32 的输出通过进位保留加法器(CSA)34 加和，CSA34 的输出 SS 和 SC 的其中两位通过加法器 37 求和的结果和 N[1]一起控制 N 编码单元产生信号 double、zero 和 neg，double、zero 和 neg 控制选择器 20 和反相器 21 产生相应 N 的倍数。上一级的 CSA 的输出 SS、SC 和反相器的输出通过 CSA27 加和，同时其结果经过寄存器 24 后输出给下一级处理单元。

从图 4 中可以看出，本级处理单元包括 4 个 w 位的寄存器，它们用来存储模乘被乘数 $B^{(i)}$、模数 $N^{(i)}$以及中间运算结果 $SS^{(i)}$和 $SC^{(i)}$。其中 $B^{(i)}$和 $N^{(i)}$各左移两位再传递到下一级运算，而中间运算结果 $SS^{(i)}$和 $SC^{(i)}$则直接经过寄存器进入下一级流水线。

附图说明

图 1 为本发明和其他参考文献的流水线结构比较。其中，（a）为 A.F.Tenca 设计的流水线结构，（b）为本发明设计的流水线结构。

图 2 为可扩展的 Montgomery 模乘器结构。

图 3 先进先出存储器结构图。

图 4 基为 4 的处理单元（PE）结构图。

图中标号：1 为模乘器的数据通路模块，2 为第 p 级处理单元，3 为第 2 级处理单元，4 为第 1 级处理单元，5、6 和 7 为存储器，8 为两输入选择器，9 为先进先出存储器模块（FIFO），10 为模乘控制模块，11 和 15 为寄存器，12 为两输入与门，13 为两输入选择器，14 为寄存器堆，16 为提前进位加法器（CLA），17、18、23、26、33、36、38 和 39 为 D 触发器，19 和 24 为寄存器，20 和 30 为三输入选择器，21 和 32 为反相器模块，22、25、29、35、40 和 41 为两输入选择器，28 为 N 编码模块，31 为 Aj 编码模块，27 和 34 为进位保留加法（CSA），37 为加法器。

具体实施方式

下面结合附图进一步描述本发明。

本发明的可扩展高基模乘器的结构可以应用于任何加密强度要求，可以处理任意长位数的有限域的模乘运算。并且可以根据实际应用需要的运算速度和硬件开销调节运算单元的流水线级数，从而达到模乘运算速度和面积的折衷。如附图 1 的结构所示，如果需要运算速度较快，可以增加流水线的级数、提高处理单元的位宽 w，或者采用更高基（如基为 2^3、2^4）。反之，如果需要较少的硬件来实现一个对速度要求不高的模乘器，则可以通过减小流水线的级数、减小处理单元的位宽 w 或采用较低的基。同时对于已经设计好的模乘运算单元，只需要增加用来存储中间运算结果的 FIFO，增加运算流水线循环的次数就可以处理更高位数的模乘运算。

附图 1 所示的流水线结构中，每一竖列代表流水线的一级（即一个 PE），横行每一行代表一个时钟周期。进行模乘运算时，首先 $N^{(0)}$、$B^{(0)}$以及 a0(k+1 位)进入第一级流水线，经过一个时钟周期运算得到 $SS^{(0)}$和 $SC^{(0)}$。第二个时钟周期将 $N^{(0)}$和 $B^{(0)}$左移 k 位，还有 $SS^{(0)}$和 $SC^{(0)}$传递到第二级进行运算，同时 $N^{(1)}$、$B^{(1)}$进入第一级进行运算。这样经过 p 级 PE 的流水线运算，中间运算结果从第 p 级的 PE 输出。下面分两种情况，如果 $e > p$，中间结果从第 p 级 PE 输出时，第 1 级 PE 仍然在运算高位的 $N^{(i)}$和 $B^{(i)}$,所以将输出的中间结果通过 CLA 加和后存入 FIFO，直到 $N^{(e-1)}$和 $B^{(e-1)}$通过第 1 级 PE 运算后，再将 FIFO 中的中间结果依次读入流水线第 1 级 PE 开始运算。如果 $e \le p$,则当中间结果从第 p 级 PE 输出时，第 1 级 PE 已经空闲，所以输出的中间结果 $SS^{(0)}$和 $SC^{(0)}$可以通过移位处理后直接进入第 1 级 PE 进行运算。

以基 4 为例的处理单元 PE 的具体结构如附图 4 所示，Aj[2:0]通过布思编码后输出三个信号用于选择需要加的 B 的倍数，其中 double 为 1 代表加 2 倍的 B，neg 为 1 代表加负数，zero 为 1 表示加 0。同样，当 firstcycle=1 时，即 $N^{(0)}$和 $B^{(0)}$运算时，CSA 运算的结果的最低 2 位加和的结果和 N[1]通过编码产生所需加 N 的倍数的选择信号，其中 double 为 1

代表加 2 倍的 N，neg 为 1 代表加负数，zero 为 1 表示加 0。运算结束后 $SS^{(i)}$和 $SC^{(i)}$经过寄存器进入下一级 PE 进行运算，同时 $B^{(i)}$和 $N^{(i)}$左移 2 位后进入下一级 PE 进行运算。

本发明的多字 Montgomery 模乘器的流水线组织结构可以使每级流水线之间的延迟只有一个时钟周期，对于 $e \leq p$ 时，可以大大提高模乘运算速度，对于 $e > p$ 时，由于省去级间寄存器可以减小硬件开销，两种情况下模乘器的速度面积比都提高了将近 2 倍。

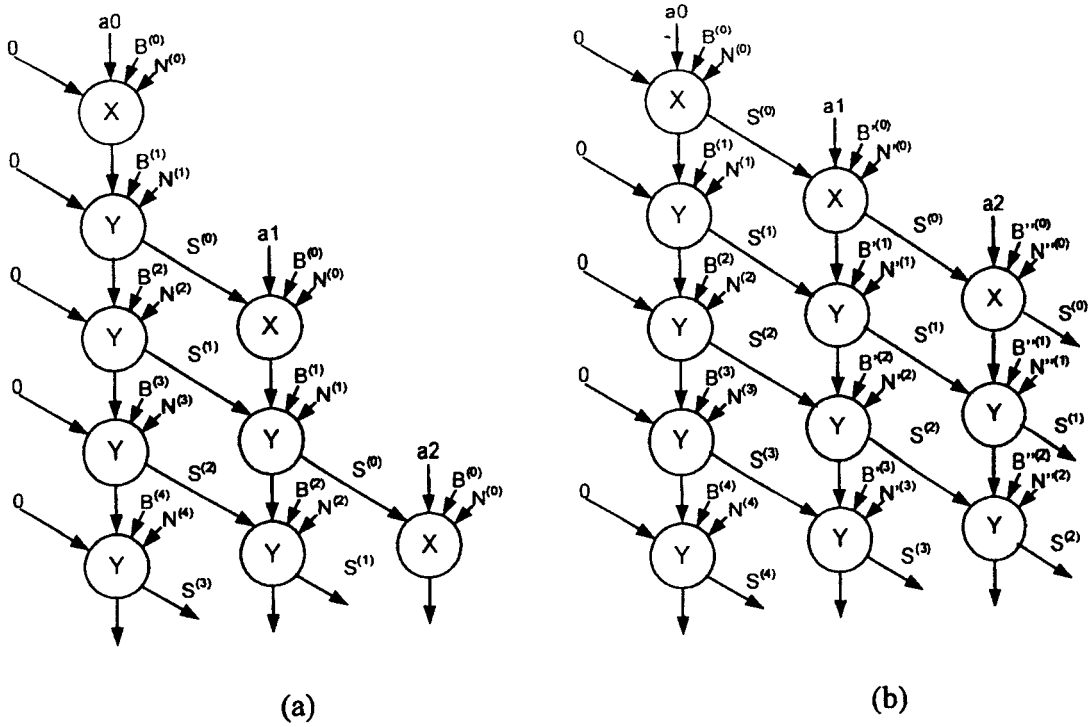

a0
a1
a2
X
Y
(a)
(b)

图 1

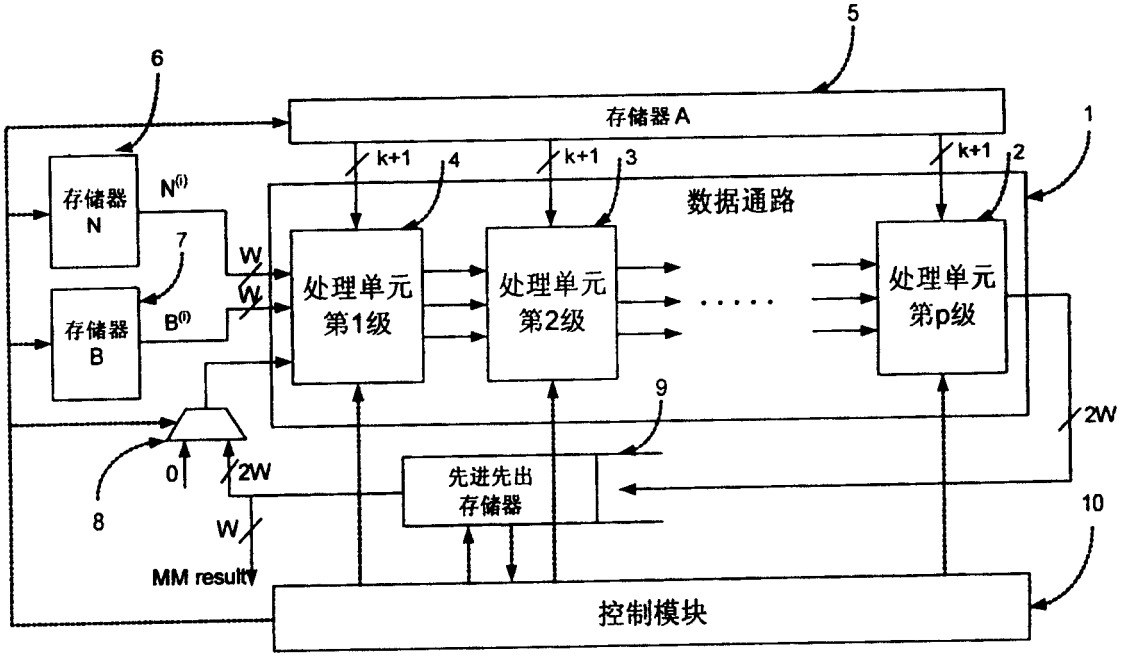

存储器 A
存储器 N
存储器 B
处理单元 第1级
处理单元 第2级
处理单元 第p级
数据通路
先进先出 存储器
控制模块
MM result
k+1
W
2W

图 2

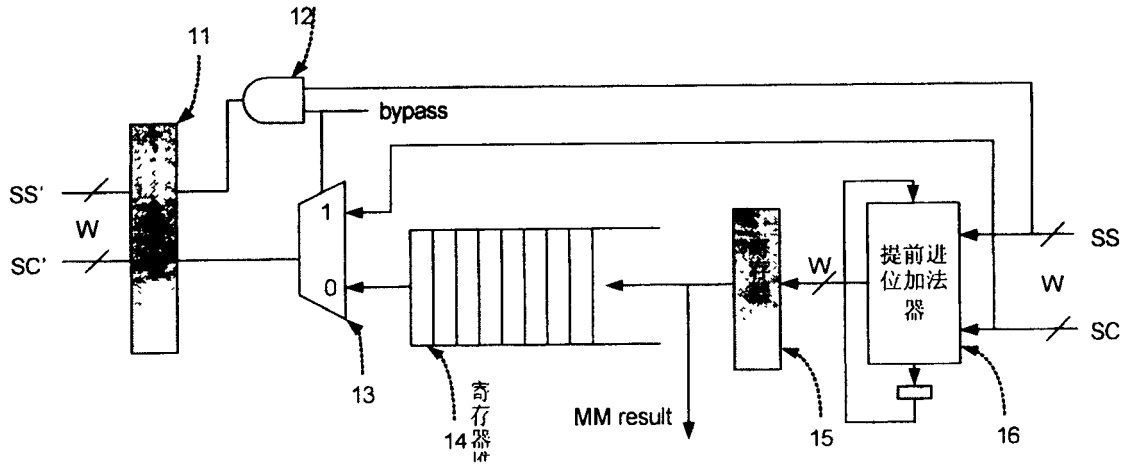

11
12
bypass
SS'
W
SC'
1
0
13
寄存器
14
MM result
W
提前进位加法器
15
16
SS
W
SC

图 3

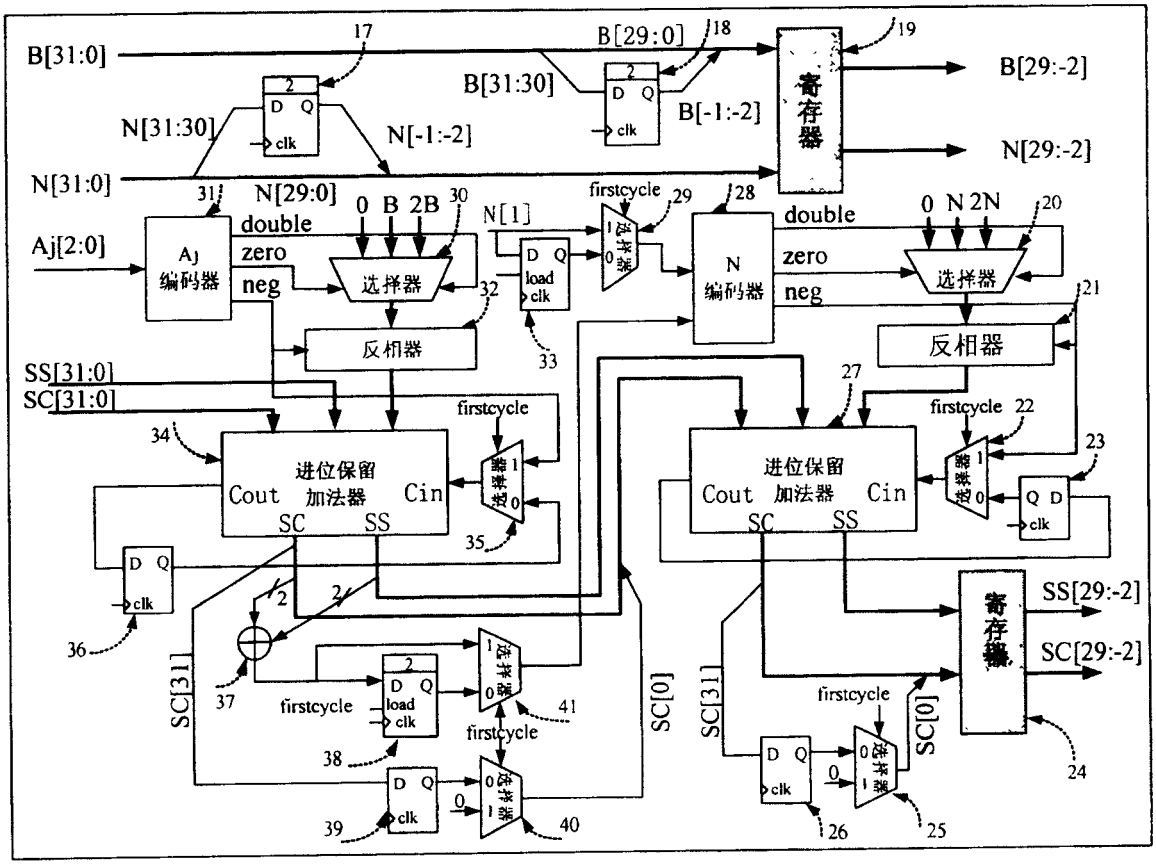

B[31:0]
17
B[29:0]
18
19
B[29:-2]
B[31:30]
B[-1:-2]
寄存器
N[31:30]
N[-1:-2]
N[29:-2]
N[31:0]
31
N[29:0]
0 B 2B
30
firstcycle
29
28
double
0 N 2N
20
Aj[2:0]
Aj 编码器
zero
neg
选择器
N[1]
load
clk
N 编码器
选择器
32
33
反相器
反相器
21
SS[31:0]
SC[31:0]
27
firstcycle
firstcycle
22
23
34
进位保留 Cout 加法器 Cin
SC SS
进位保留 Cout 加法器 Cin
SC SS
35
36
37
38
39
40
41
SC[31]
SC[0]
SC[31]
SC[0]
SS[29:-2]
SC[29:-2]
寄存输出
24
25
26

图 4