[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 7/556

[12] 发明专利申请公开说明书

[21] 申请号 02808541.8

[43] 公开日 2004 年 6 月 9 日

[11] 公开号 CN 1503939A

[22] 申请日 2002.2.15 [21] 申请号 02808541.8
[30] 优先权
[32] 2001. 2.21 [33] US [31] 09/788,684
[86] 国际申请 PCT/US2002/004415 2002.2.15
[87] 国际公布 WO02/069136 英 2002.9.6
[85] 进入国家阶段日期 2003.10.20
[71] 申请人 美普思科技有限公司
地址 美国加利福尼亚州
[72] 发明人 M·斯特里贝克 K·D·基斯塞尔
P·帕里尔

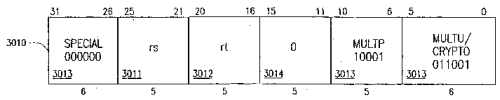
[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 杨 凯 张志醒

权利要求书 6 页 说明书 7 页 附图 3 页

[54] 发明名称 多项式算术运算

[57] 摘要

提供了指令集体系结构(ISA)中的多项式运算指令(3010)。 还提供了多项式乘 - 加(MADDP)指令和多项式乘(MULTP)指令(3013)。



ISSN1008-4274

1. 一种在指令集体系结构中用于执行多项式算术的指令，所述指令为所述指令集体系结构的一部分并包括:

一个或多个将所述指令识别为用于执行多项式算术运算的指令的操作码；和

一个或多个寄存器标识符;

其中，使用所述一个或多个寄存器标识符来执行所述多项式算术运算从而处理所述指令。

2. 如权利要求1所述的指令，其特征在于，所述多项式算术运算为二进制多项式加法。

3. 如权利要求2所述的指令，其特征在于，所述二进制多项式加法采用乘法器来执行。

4. 如权利要求1所述的指令，其特征在于，所述多项式算术运算的结果存于一个或多个结果寄存器中。

5. 如权利要求4所述的指令，其特征在于，所述多项式算术运算包括:

将由所述一个或多个寄存器标识符标识的所述寄存器中的内容相乘而得到一个中间值；以及

将所述一个或多个结果寄存器的内容与所述中间值相加而得到结果。

6. 如权利要求5所述的指令，其特征在于，所述结果存于所述一个或多个结果寄存器中。

7. 如权利要求1所述的指令，其特征在于，所述多项式算术运算的结果存于高阶结果寄存器和低阶结果寄存器中。

8. 如权利要求1所述的指令，其特征在于，所述多项式算术运算为多项式乘法。

9. 如权利要求8所述的指令，其特征在于，由所述一个或多个寄

存器标识符识别的各寄存器包含有多项式。

10. 如权利要求 9 所述的指令，其特征在于，将每个多项式编码为系数的二进制表示。

11. 如权利要求 1 所述的指令，其特征在于，所述指令集包括 RISC 指令集。

12. 一种使用指令来执行多项式算术的方法，所述方法包括:

接收指令，所述指令包括:

一个或多个将所述指令识别为用于执行多项式算术运算的指令的操作码；和

一个或多个寄存器标识符；和

通过处理所述指令利用所述一个或多个寄存器标识符来执行多项式算术运算。

13. 如权利要求 12 所述的方法，其特征在于，执行所述多项式算术运算包括执行二进制多项式加法。

14. 如权利要求 13 所述的方法，其特征在于，执行所述二进制多项式加法包括使用乘法器。

15. 如权利要求 12 所述的方法，其特征在于还包括将所述多项式算术运算的结果存于一个或多个结果寄存器中。

16. 如权利要求 15 所述的方法，其特征在于，执行所述多项式算术运算包括:

将由所述一个或多个寄存器标识符标识的所述寄存器中的内容相乘而得到一个中间值；和

将所述一个或多个结果寄存器的内容与所述中间值相加而得到结果。

17. 如权利要求 16 所述的方法，其特征在于还包括将所述结果存于所述一个或多个结果寄存器中。

18. 如权利要求 12 所述的方法，还包括将所述多项式算术运算的结果存于高阶结果寄存器和低阶结果寄存器中。

19. 如权利要求 12 所述的方法，其特征在于，执行所述多项式算术运算包括执行多项式乘法。

20. 如权利要求 19 所述的方法，其特征在于，由所述一个或多个寄存器标识符识别的媒体各寄存器包含多项式。

21. 如权利要求 20 所述的方法，其特征在于，将每个多项式编码为系数的二进制表示。

22. 如权利要求 12 所述的方法，其特征在于，所述指令为指令集的一部分，并且所述指令集包括 RISC 指令集。

23. 一种包括用软件实现的微处理器核心的计算机可读媒体，所述微处理器核心包括用于执行多项式算术的指令，所述指令包括:

将所述指令识别为一个或多个用于执行多项式算术运算的指令的操作码；和

一个或多个寄存器标识符;

其中，通过使用一个或多个寄存器标识符来执行所述多项式算术运算从而处理所述指令。

24. 如权利要求 23 所述的计算机可读媒体，其特征在于，所述多项式算术运算为二进制多项式加法。

25. 如权利要求 24 所述的计算机可读媒体，其特征在于，所述二进制多项式加法采用乘法器来执行。

26. 如权利要求 23 所述的计算机可读媒体，其特征在于，所述多项式算术运算的结果存于一个或多个结果寄存器中。

27. 如权利要求 26 所述的计算机可读媒体，其特征在于，所述多项式算术运算包括:

将由所述一个或多个寄存器标识符标识的所述寄存器中的内容相乘而得到一个中间值；和

将所述一个或多个结果寄存器的所述内容与所述中间值相加而得到结果。

28. 如权利要求 27 所述的计算机可读媒体，其特征在于，所述结

果存于所述一个或多个结果寄存器中。

29. 如权利要求23所述的计算机可读媒体，其特征在于，所述多项式算术运算的结果存于高阶结果寄存器和低阶结果寄存器中。

30. 如权利要求23所述的计算机可读媒体，其特征在于，所述多项式算术运算是多项式乘法。

31. 如权利要求30所述的计算机可读媒体，其特征在于，由所述一个或多个寄存器标识符识别的各寄存器包含多项式。

32. 如权利要求31所述的计算机可读媒体，其特征在于，将每个多项式编码为系数的二进制表示。

33. 如权利要求23所述的计算机可读媒体，其特征在于，所述指令是指令集的一部分，并且所述指令集包括RISC指令集。

34. 一种在公共密钥密码系统中用公共密钥对信息加密的方法，所述方法包括用于执行多项式算术的指令，所述指令包括:

一个或多个将所述指令识别为用于执行多项式算术运算的指令的操作码；和

一个或多个寄存器标识符;

其中，通过使用所述一个或多个寄存器标识符来执行所述多项式算术运算从而处理所述指令。

35. 如权利要求34所述的方法，其特征在于，所述多项式算术运算是二进制多项式加法。

36. 如权利要求35所述的方法，其特征在于，所述二进制多项式加法使用乘法器来执行。

37. 如权利要求34所述的方法，其特征在于，所述多项式算术运算的结果存于一个或多个结果寄存器中。

38. 如权利要求37所述的方法，其特征在于，所述多项式算术运算包括:

将由所述一个或多个寄存器标识符标识的所述寄存器中的内容相乘而得到一个中间值；和

将所述一个或多个结果寄存器的内容与所述中间值相加而得到结果。

39. 如权利要求 38 所述的方法，其特征在于，所述结果存于所述一个或多个结果寄存器中。

40. 如权利要求 34 所述的方法，其特征在于，所述多项式算术运算的结果存于高阶结果寄存器和低阶结果寄存器中。

41. 如权利要求 34 所述的方法，其特征在于，所述多项式算术运算是多项式乘法。

42. 如权利要求 41 所述的方法，其特征在于，由所述一个或多个寄存器标识符标识的媒体各寄存器包含多项式。

43. 如权利要求 42 所述的方法，其特征在于，将每个多项式编码为系数的二进制表示。

44. 如权利要求 34 所述的方法，其特征在于，所述指令是指令集的一部分，并且所述指令集包括 RISC 指令集。

45. 一种在微处理器中用于执行多项式算术的指令，所述指令包括:

一个或多个将所述指令识别为用于执行多项式算术运算的指令的操作码；和

一个或多个寄存器标识符;

其中，通过使用所述一个或多个寄存器标识符来执行所述多项式算术运算从而处理所述指令。

46. 如权利要求 45 所述的指令，其特征在于，所述多项式算术运算是二进制多项式加法。

47. 如权利要求 46 所述的指令，其特征在于，所述二进制多项式加法采用乘法器来执行。

48. 如权利要求 45 所述的指令，其特征在于，所述多项式算术运算包括:

将由所述一个或多个寄存器标识符标识的所述寄存器中的内容

相乘而得到一个中间值；和

将所述一个或多个结果寄存器的内容与所述中间值相加而得到结果。

49. 如权利要求45所述的指令，其特征在于，所述多项式算术运算是多项式乘法。

50. 一种提供了一条或多条用于执行多项式算术的指令的微处理器，所述微处理器包括：

指令存储器；

执行单元，所述执行单元从所述指令存储器中取微处理器指令并处理所取指令；和

多项式算术单元，如果所取指令是所述一条或多条用于执行多项式算术的指令之一，则所述多项式算术单元在处理所取指令中由所述执行单元使用。

51. 如权利要求50所述的微处理器，其特征在于，所述微处理器还包括乘/除单元。

52. 如权利要求51所述的微处理器，其特征在于，所述多项式算术单元是所述乘/除单元的部件。

53. 如权利要求50所述的微处理器，其特征在于，所述多项式算术单元可用于执行二进制多项式加法。

54. 如权利要求50所述的微处理器，其特征在于，所述多项式算术单元可用于执行二进制多项式乘法。

55. 如权利要求50所述的微处理器，其特征在于，所述微处理器还包括用于存储来自所述多项式算术单元的结果的结果寄存器。

56. 如权利要求55所述的微处理器，其特征在于，通过执行二进制多项式乘法运算以确定中间结果并将所述中间结果加到所述结果寄存器中，所述多项式算术单元可用于执行二进制多项式乘-加操作。

多项式算术运算

技术领域

本发明涉及用于执行多项式算术的微处理器指令，具体说涉及用于执行多项式乘法运算的微处理器指令。

背景

当业界朝向更庞大更复杂的指令集发展时，开发出了精简指令集计算机（RISC）体系结构。通过简化指令集设计，RISC 体系结构使得应用例如流水线和高速缓冲技术更为容易，从而提高了系统的性能。

RISC 体系结构一般具有在指令格式上几乎没什么变化的固定长度指令（例如 16 位，32 位或 64 位）。指令集体系结构（ISA）中的每个指令可具有总处于相同位置的源寄存器。例如，32 位 ISA 可具有总由位 16-20 和 21-25 指定的源寄存器。对每条指令而言，这允许无需任何复杂的指令译码就可读取指定寄存器。

概述

加密的系统（“密码系统”）越来越多地用于确保交易安全，对通信加密，对用户进行认证以及保护信息。许多专用密钥密码系统（例如数字加密算法（DES）），在计算上相对简单，并且常常可简化为对数据块进行一系列异或（XOR）、循环和置换运算的硬件解决方案。而另一方面，公共密钥密码系统比专用密钥密码系统在数学上更巧妙且在计算上更困难。

虽然不同的公共密钥密码系统方案依据不同的数学基础，但是它们往往都需要在数量级为 1024 位的大数值范围内进行整数运算。扩展精度运算常常基于模数（即以某个值域为模进行的运算），而

在某些情况下则是基于二进制多项式而非二进制补码。例如，RSA公共密钥密码系统采用扩展精度模取幂来对信息加密和解密，而椭圆曲线密码系统采用扩展精度模多项式乘法对信息加密和解密。

公共密钥密码系统已经广泛用于用户认证和安全密钥交换，而专用密钥密码系统广泛用于加密通信通道。随着公共密钥密码系统的应用增多，希望能够提高扩展精度模算术运算的性能。

一般而言，指令集体系结构包括用于执行多项式算术的指令。该指令包括一个或多个将此指令识别为用于执行多项式算术运算的指令的操作码。此外，该指令识别一个或多个寄存器。通过使用所识别的寄存器来执行多项式算术运算，从而可对该指令加以处理。

实现可提供执行二进制多项式加法的指令，该指令可采用乘法器来实施。多项式算术运算的结果可存于一个或多个结果寄存器中。多项式算术运算可包括乘法，其中，使所标识的寄存器的内容相乘。运算还可以包括多项式的乘加运算，其中，所标识的寄存器的内容相乘然后加到一个或多个结果寄存器中。结果寄存器可包括高阶寄存器和低阶寄存器。可对存于寄存器中的多项式进行多项式算术运算。多项式可编码为系数的二进制表示。

以下附图和说明书中对一个或多个实现加以阐述。从说明书和附图以及权利要求中可明显看出本发明的其他特征和优点。

附图说明

图 1 是可用于 RISC 体系结构中的五级流水线示例的框图，

图 2 是包括执行核心和乘/除运算单元的处理器核心的框图。

图 3A 和 3B 是执行多项式乘法和加法的例示指令的指令编码。

详细说明

许多公共密钥密码系统采用扩展精度模运算来对数据加密和解密。例如，多数椭圆曲线（EC）密码系统大量采用二进制多项式乘

法和加法来对数据加密和解密。椭圆曲线密码系统的性能可通过修改编程 CPU 乘法器以响应新定义的专用于多项式运算的指令来加以提高。

当（如 IEEE1363-2000 标准所推荐的那样）采用定义于 GF（2^{163}）上的椭圆曲线时，所需的主要运算是在 GF（2^{163}）域上的乘法运算。在 2^{163} 个元素中每一个元素可表示为系数为 0 或 1 的至多 163 次幂的多项式。在此表示中，两个元素可以采用简单的按位异或相加，而两个多项式 a（X）和 b（X）可通过计算 a(X)b(X) mod P(X)而得到相乘的结果，乘积 a(X)b(X)是 326 次多项式，P（X）是 IEEE1363-2000 标准规定的既约多项式。

多项式乘法与模数乘法具有相同的形式，在整数范围内执行 ab mod p，不同之处在于：（1）常规的加法由异或替代；和（2）常规的 32 位乘法由 32 位不带进位的乘法替代。因此，可采用移位和异或而不是移位和加法运算来执行多项式模数乘法运算。

参考图 1，可用来实现多项式乘法运算的示范性的微处理器结构包括五级流水线，其中，指令可在每个时钟周期发出并在固定的时间例如 5 个时钟周期内执行。每条指令的执行分成 5 步：取指（IF）级 1001，读寄存器（RD）级 1002，算术/逻辑单元（ALU）级 1003，存储（MEM）级 1004 和写回（WB）级 1005。在 IF 级 1001 中，从指令高速缓冲器中取出指定指令。所取指令的一部分用于指定可用于执行指令的源寄存器。在读寄存器（RD）级 1002 中，系统将指定源寄存器的内容取出。所取得的值可用在 ALU 级 1003 中执行算术或逻辑运算。在 MEM 级 1004 中，执行指令可读/写数据高速缓冲器中的存储器。最后，在 WB 级 1005 中，通过执行指令而获得的值可写回到某个寄存器中。

因为有些运算，例如浮点运算和整数乘/除运算未必能够在一个时钟周期内完成，某些指令仅仅开始指令的执行。在经过足够的时钟周期后，另一指令可用于取回结果。例如，当整数乘法指令花费 5

个时钟周期时，一条指令可启动乘法计算，而另一条指令可在乘法运算完成后将乘积装入寄存器中。如果在需要结果的时候乘法运算还未完成，流水线可停止直到结果可得。

参考图 2，作为示例给出示范性的 RISC 体系结构。处理器核心 2000（也称为“微处理器核心”）包括如下单元：执行单元 2010、乘/除运算单元（MDU）2020、系统控制协处理器（CPO）2030、存储管理单元 2040、高速缓存控制器 2050 和总线接口单元（BIU）2060。

执行单元 2010 是在处理器核心 2000 内执行指令的主要机构。执行单元 2010 包括寄存器阵列 2011 和算术逻辑单元（ALU）2012。在一种实现中，寄存器阵列 2011 包括 32 个可用于例如标量整数运算和地址计算的 32 位通用寄存器。可将包括两个读端口和一个写端口的寄存器阵列 2011 完全旁路，以使流水线中的运算延迟最小。ALU 2012 支持逻辑和算术运算，例如加法、减法和移位。

MDU 2020 执行乘法和除法运算。在一种实现中，MDU 2020 包括 32 位乘 16 位 (32×16)Booth 编码的（Booth-encoded）乘法器（未显示）、结果寄存器（HI 寄存器 2021 和 LO 寄存器 2022）、除法状态机以及执行这些功能的所需的所有多路复用器和控制逻辑。在一种流水线式实现中，每时钟周期可将 32×16 乘法运算发送给 MDU2020，以便每个时钟周期 32 位的数可同 16 位的数相乘。但是直到乘法运算完成后 HI/LO 寄存器（2021 和 2022）中才有可用结果。可用 MFHI 和 MFLO 指令来访问结果。这些指令将结果从 HI 寄存器 2021 和 LO 寄存器 2022 中分别移至指定的寄存器。例如“MFHI $7”将 HI 寄存器 2021 的内容移至通用寄存器$7 中。

乘-加（MADD/MADDU）和乘-减（MSUB/MSUBU）这两条指令可用于执行乘-加和乘-减运算。MADD 指令使两个数相乘后再将乘积加到 HI 寄存器 2021 和 LO 寄存器 2022 的当前内容中。然后将结果存于 HI/LO 寄存器（2021 和 2022）中。类似地，MSUB 指令使两个操作数相乘后再将乘积从 HI 寄存器 2021 和 LO 寄存器 2022 中

减去，然后将结果存于 HI/LO 寄存器（2021 和 2022）中。MADD 和 MSUB 指令对符号数执行运算。MADDU 和 MSUBU 指令对无符号数执行类似运算。

参考图 3A，提供了多项式乘法（MULTP）指令 3010 的示范性的指令编码。MULTP 指令 3010 有两个寄存器字段，即 rs 3011 和 rt 3012，用于指定包含将要参与相乘的多项式的源寄存器。在乘运算完成之后，结果存在 HI 寄存器 2021 和 LO 寄存器 2022 中。MULTP 指令 3010 还包括一个或多个用于识别将要执行的运算的操作码 3013。在一些实现中，可以不用指令字段的一部分，例如字段 3014。

在一种实现中，由 rs3011 和 rt3012 标识的寄存器包含二进制多项式（即模 2 化简的多项式系数）。因此，各系数或为“1”或为“0”。在 32 位寄存器中对多项式进行编码，其中每一位表示一个多项式系数。例如将多项式“x^4+x+1”编码为“10011”，因为 x^3 和 x^2 的系数为“0”，其余系数为“1”。

MULTP 指令 3010 允许将两个多项式相乘。例如，$(x^4+x+1)(x+1)=x^5+x^4+x^2+2x+1$。模 2 化简多项式得到 $x^5+x^4+x^2+1$。如果多项式以上述二进制编码，那么同一乘法可表示为 (10011)(11)=110101。

指令和操作数的长度可任意改变；所描述的 32 位设计仅为一个例子。在 32 位的实现中，存于 rs 3011 中的 32 位字的值可以同存于 rt 3012 中的 32 位字的值进行多项式的乘法，即将两个操作数均作为二进制多项式值，以产生 64 位的结果。低阶 32 位字可放在 LO 寄存器 2022 中，高阶 32 位字结果可放在 HI 寄存器 2021 中。在某些实现中，不会发生运算异常。如果由 rs 3011 和 rt 3012 所指定的寄存器不包含 32 位带符号扩展的值，那么运算的结果可能无法预料。

参考图 3B，提供了多项式乘加（MADDP）指令 3020 的示范性指令编码。MADDP 指令 3020 有两个参数字段，即 rs 3021 和 rt 3022，用于指定源寄存器，该源寄存器包含要执行乘法运算并采用多项式

加法（异或）加到 HI 2021 和 LO 2022 的内容中的多项式。乘法和加法运算完成之后，其结果存在 HI 寄存器 2021 和 LO 寄存器 2022 中。MADDP 指令 3020 还可包括一个或多个识别所要执行的运算的操作码 3023。在一些实现中，可以不用指令段的一部分，例如字段 3024。

MADDP 指令 3020 执行如上讨论的乘法运算。二进制多项式加法类似于按位异或运算。例如，二进制多项式加（x^4+x+1）+（x+1）的结果是 x^4+2x+2。模 2 化简系数得到 x^4，x^4 可表示为“10000”。

同样地，指令和操作数的长度可任意改变。在一种实现中，存于 rs 3021 的 32 位字的值可与存于 rt 3022 中的 32 位字的值作基于多项式的乘法运算，即把两个操作数均作为二进制多项式的值，从而得到 64 位结果。然后该结果可采用多项式加法加到 HI 寄存器 2021 和 LO 寄存器 2022 中的内容中。64 位的结果包括低阶 32 位字和高阶 32 位字。低阶 32 位字可放在 LO 寄存器 2022 中，高阶 32 位字结果可放在 HI 寄存器 2021 中。如果由 rs 3021 和 rt 3022 指定的寄存器不包含 32 位带符号扩展的值，那么运算的结果可能无法预料。

多项式算术的实现除可采用硬件（如在微处理器或微控制器内）实现，还可用软件实现，所述软件设置在例如经配置用于存储软件（即计算机可读程序代码）的计算机可用（如可读的）媒体中。所述程序代码可使本说明书所公开的系统和技术的功能或构造或者二者均得以实现。例如，这可以通过使用通用编程语言（如 C，C++）、硬件描述语言（HDL）（包括 Verilog HDL、VHDL、AHDL(Altera HDL)等）、或者其它可用的编程和/或电路捕获工具来完成。程序代码可配置在任何已知的计算机可用媒体中，包括半导体、磁碟片、光盘（例如 CD-ROM，DVD-ROM），以及配置为实现于计算机可用（如可读）传输媒体（如载波和任何别的包括数字的、光学的或基于模拟的媒体）中的计算机数据信号。这样，代码可通过包括因特网和企业内部网的通信网络来传输。

应理解，上述系统和技术所完成的功能和/或所提供的结构可以

用于程序代码来实现的核心（如微处理器核心）来表示，并且可转换成作为集成电路产品一部分的硬件。所述系统和技术还可体现为硬件和软件的组合。因此，其它实现方式也在以下权利要求范围之内。

图 1

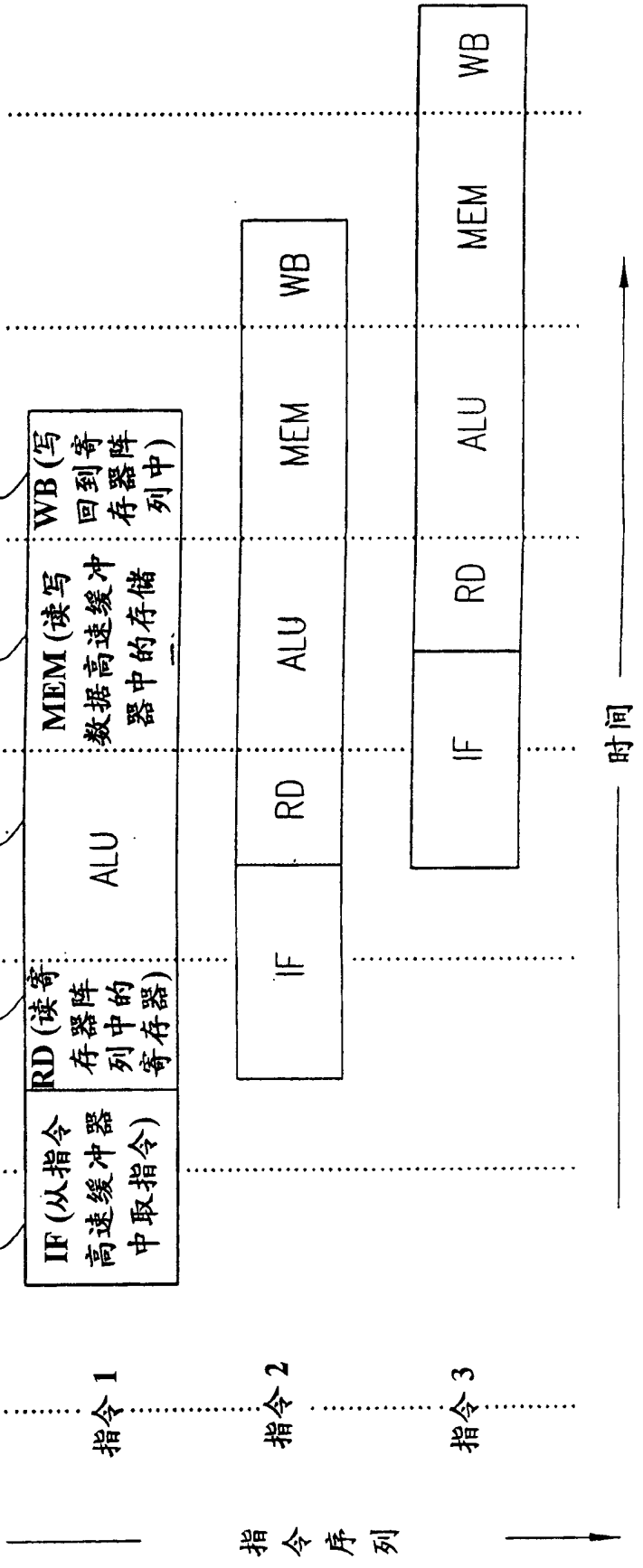

1001
1002
1003
1004
1005
IF (从指令高速缓冲器中取指令)
RD (读寄存器阵列中的寄存器)
ALU
MEM (读写数据高速缓冲器中的存储)
WB (写回到寄存器阵列中)
IF
RD
ALU
MEM
WB
IF
RD
ALU
MEM
WB
指令1
指令2
指令3
指令序列
时间

图 2

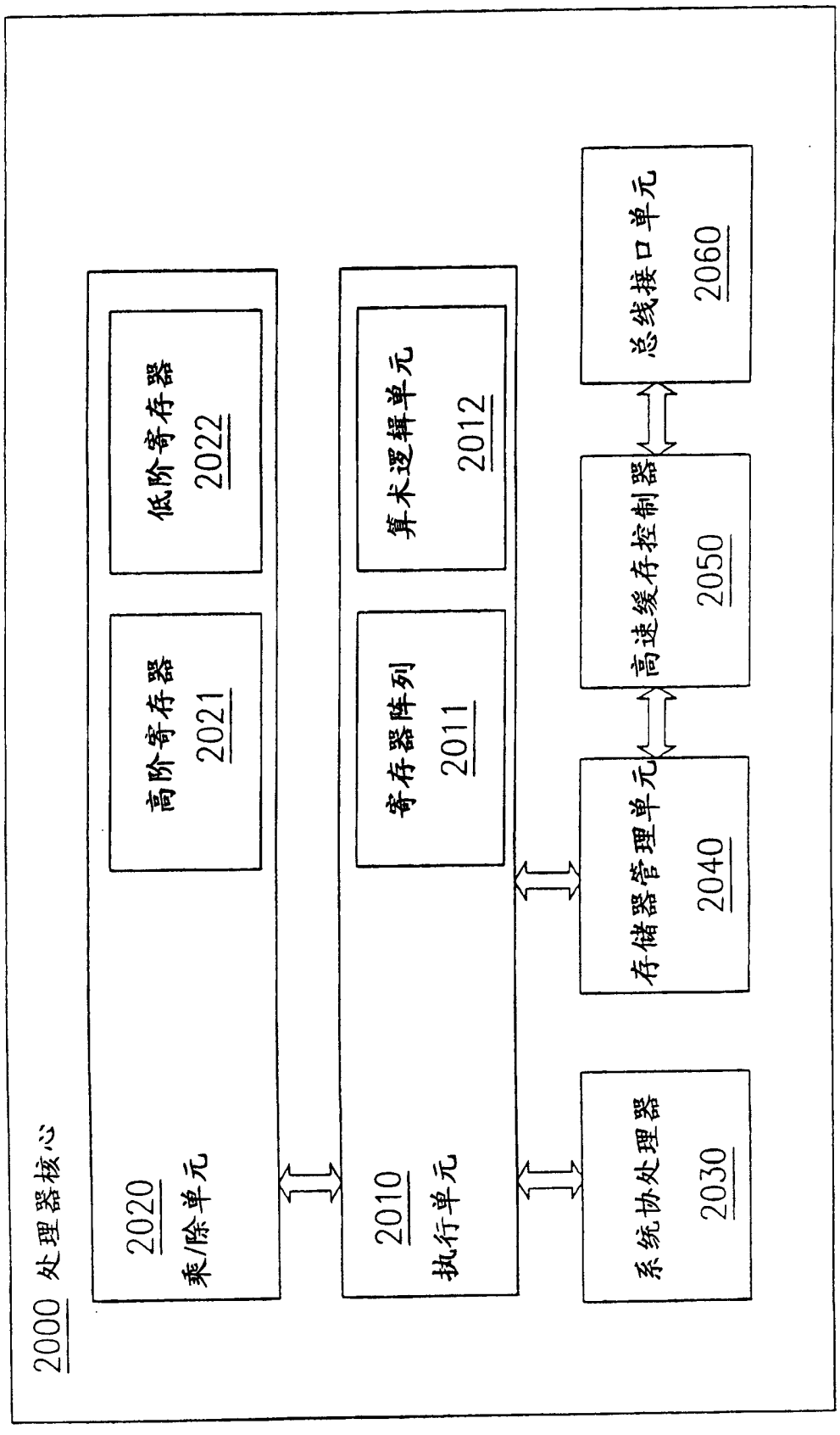

2000 处理器核心
2020
乘/除单元
高阶寄存器
2021
低阶寄存器
2022
2010
执行单元
寄存器阵列
2011
算术逻辑单元
2012
系统协处理器
2030
存储器管理单元
2040
高速缓存控制器
2050
总线接口单元
2060

图 3A

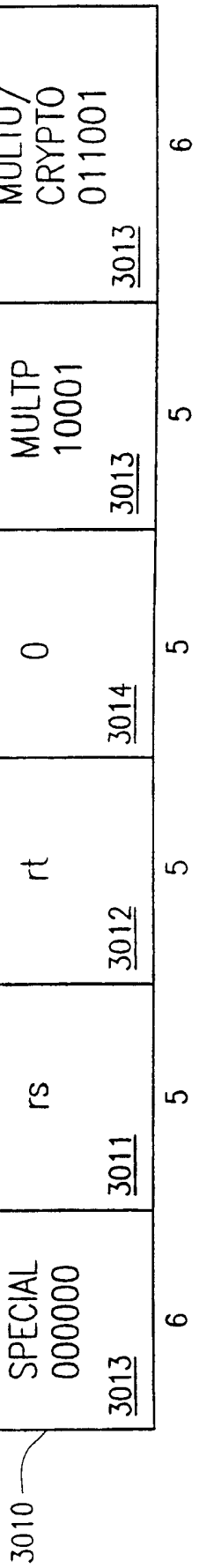
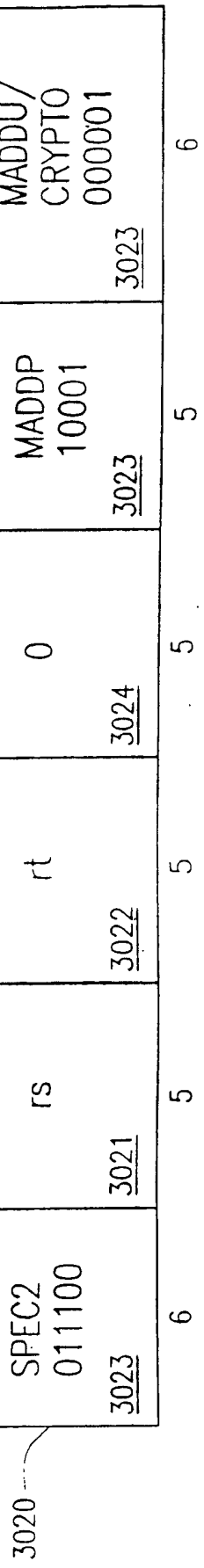

3010
SPECIAL
000000
3013
rs
3011
rt
3012
0
3014
MULTP
10001
3013
MULTU/
CRYPTO
011001
3013
31
26
25
21
20
16
15
11
10
6
5
0
6
5
5
5
5
6

图 3B


3020
SPEC2
011100
3023
rs
3021
rt
3022
0
3024
MADDP
10001
3023
MADDU/
CRYPTO
000001
3023
31
26
25
21
20
16
15
11
10
6
5
0
6
5
5
5
5
6