[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/302 (2006.01)
G06F 9/318 (2006.01)

[12] 发明专利说明书

专利号 ZL 200610065301.8

[45] 授权公告日 2008 年 3 月 26 日

[11] 授权公告号 CN 100377075C

[22] 申请日 2006.3.17
[21] 申请号 200610065301.8
[30] 优先权
[32] 2005.3.18 [33] US [31] 11/083,543
[73] 专利权人 威盛电子股份有限公司
地址 台湾省台北县
[72] 发明人 葛伦·亨利 劳德利·虎克
泰利·派克
[56] 参考文献
CN1431584A 2003.7.23
CN1512324A 2004.7.14
CN1177138A 1998.3.25
CN1326566A 2001.12.12
US6014735A 2000.1.11
审查员 魏 峰

[74] 专利代理机构 北京市柳沈律师事务所
代理人 钱大勇

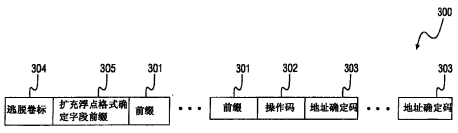
权利要求书 4 页 说明书 19 页 附图 11 页

[54] 发明名称

使浮点格式实现指令级规格的设备及方法

[57] 摘要

本发明涉及一种使浮点格式实现指令级规格的设备及方法，及具有超出其现有能力而提供浮点格式规格的设备。 该设备包含转换逻辑电路及扩充执行逻辑电路，其中转换逻辑电路将一扩充指令转换为对应微指令，扩充指令则具有一扩充前缀及一扩充前缀卷标。 扩充前缀用以确定浮点格式，扩充前缀卷标则指出该扩充前缀，其中该扩充前缀卷标为一微处理器所用的一指令集中在结构上确定的操作码。 扩充执行逻辑电路耦接至转换逻辑电路，并接收对应微指令，且依据该扩充前缀中确定的浮点格式执行相关浮点运算。 本发明可保证兼容性的同时，使浮点格式实现指令级规格。



1. 一种微处理器设备，用以实现一浮点格式在一相关浮点运算执行期间的使用，其特征在于，包含：

转换逻辑电路，将一扩充指令转换为对应微指令，其中该扩充指令包含：

一扩充前缀，以确定该浮点格式，所述扩充前缀还包括一精度码字段以及一舍入码字段，以分别确定一精度格式以及一舍入格式，所述扩充前缀还具有浮点例外屏蔽掩码，以阻止浮点例外的发生；

一扩充前缀卷标，用以指出该扩充前缀，并且是微处理器所用的一指令集在结构上以其它方式确定的操作码；

该转换逻辑电路包含：

逃脱指令检测逻辑电路，用以检测该扩充前缀卷标；

扩充前缀译码逻辑电路，耦接至该逃脱指令检测逻辑电路，用以转换该扩充前缀，并用以设定上述对应微指令中的一微操作码扩充字段，其中该微操作码扩充字段用于指定该浮点格式；

指令译码逻辑电路，用以设定上述对应微指令中的其它字段，其中该其它字段用以确定该相关浮点运算；以及

扩充浮点执行逻辑电路，耦接至该转换逻辑电路，并用以接收对应微指令，且依据该扩充前缀中确定的浮点格式执行该相关浮点运算。

2. 如权利要求 1 所述的微处理器设备，其特征在于，该扩充指令还包含，根据该指令集所在结构确定的部分。

3. 如权利要求 2 所述的微处理器设备，其特征在于，该所在结构确定的部分包含一操作码部分，用以确定该相关浮点运算。

4. 如权利要求 1 所述的微处理器设备，其特征在于，当该指令集包含一 x86 指令集时，该扩充前缀卷标包含该 x86 指令集的内部电路仿真断点操作码。

5. 一种扩充装置，用以对一现存微处理器指令集增加浮点格式掩码特征，其特征在于，包含：

一指令扩充器，用以令一微处理器根据一浮点格式执行一相关浮点运算，其中该扩充指令包含在该现存微处理器指令集之一选出操作码，其中

现存微处理器指令集之后有一 n 位扩充前缀，该经选出的操作码指出该扩充指令，且该 n 位扩充前缀指出该浮点格式，所述扩充前缀包括一精度码字段以及一舍入码字段，以分别确定一精度格式以及一舍入格式，所述扩充前缀还具有浮点例外屏蔽掩码，以阻止浮点例外的发生；以及

一转换器，用以接收该扩充指令，并用以产生一微指令序列，以令该微处理器根据该浮点格式执行该相关浮点运算；

该转换器还包含：

逃脱指令检测逻辑电路，用以检测扩充前缀卷标；

扩充前缀译码逻辑电路，耦接至该逃脱指令检测逻辑电路，用以转换该扩充前缀，并用以设定上述对应微指令中的一微操作码扩充字段，其中该微操作码扩充字段用于指定该浮点格式；

指令译码逻辑电路，用以设定上述对应微指令中的其它字段，其中该其它字段用以确定该相关浮点运算。

6. 如权利要求 5 所述的扩充装置，其特征在于，该扩充指令还包含根据该指令集所在结构确定的部分，用以指定该相关浮点运算。

7. 如权利要求 5 所述的扩充装置，其特征在于，当该现存微处理器指令集为 x86 微处理器指令集，该选出的操作码包含该 x86 微处理器指令集中的内部电路仿真断点操作码。

8. 一指令集扩充设备，用以对一现存微处理器指令集提供以指令级浮点格式掩码能力，其特征在于，该指令集扩充设备包含：

一转换逻辑电路，用以接收逃脱卷标，并用以指出一对应指令的伴随部分指定一为微处理器执行的一扩充运算，其中该逃脱卷标为该现存微处理器指令集中一第一操作码部分；该逃脱卷标耦接至一浮点格式确定字段，该浮点格式确定字段为该伴随部分之一，用以指定一待于该扩充运算执行期间所用的一浮点格式，所述浮点格式确定字段包括一精度码字段以及一舍入码字段，以分别确定一精度格式以及一舍入格式，所述逃脱卷标还具有浮点例外屏蔽掩码，以阻止浮点例外的发生；

该转换逻辑电路包含：

逃脱指令检测逻辑电路，用以检测扩充前缀卷标；

扩充前缀译码逻辑电路，耦接至该逃脱指令检测逻辑电路，用以转换该扩充前缀，并用以设定上述对应微指令中的一微操作码扩充字段，其

中该微操作码扩充字段用于指定该浮点格式;

指令译码逻辑电路，用以设定上述对应微指令中的其它字段，其中该其它字段用以确定该相关浮点运算；以及

扩充浮点执行逻辑电路，耦接至该转换逻辑电路，用以根据该浮点格式执行该扩充运算。

9. 如权利要求 8 所述的指令集扩充设备，其特征在于，该伴随部分的其余者包含一第二操作码部分及一选择性多个地址确定码字段部分。

10. 如权利要求 8 所述的指令集扩充设备，其特征在于，当该现存微处理器指令集为 x86 微处理器指令集，该第一操作码部分包含该 x86 微处理器指令集中的内部电路仿真断点操作码部分。

11. 一种扩充一微处理器指令集以提供一浮点格式指令集掩码的方法，其特征在于，该方法包含下列步骤:

提供一扩充指令，该扩充指令包含一扩充卷标及一扩充前缀，其中该扩充卷标为该微处理器指令集中的一第一操作码，所述扩充前缀还包括一精度码字段以及一舍入码字段，以分别确定一精度格式以及一舍入格式，所述扩充前缀还具有浮点例外屏蔽掩码，以阻止浮点例外的发生;

检测该扩充前缀卷标;

转换该扩充前缀，并用以设定上述对应微指令中的一微操作码扩充字段，其中该微操作码扩充字段用于指定该浮点格式;

设定上述对应微指令中的其它字段，其中该其它字段用以确定该相关浮点运算;

通过该扩充前缀及该扩充指令的其余部分指定一待执行运算，其中该指定步骤包含下列步骤:

第一确定步骤，用以确定该运算所用的精度格式及舍入格式；以及

根据该确定动作执行该运算。

12. 如权利要求 11 所述的扩充一微处理器指令集以提供一浮点格式指令集掩码的方法，其特征在于，该指定步骤还包含下列步骤:

第二确定步骤，用以确定该运算，其中该第二确定步骤包含使用该微处理器指应集中一第二操作码的步骤。

13. 如权利要求 12 所述的扩充一微处理器指令集以提供一浮点格式指令集掩码的方法，其特征在于，该提供步骤包含自 x86 微处理器指令集中

选出该第一操作码的步骤。

14. 如权利要求 13 所述的扩充一微处理器指令集以提供一浮点格式指令集掩码的方法，其特征在于，该选出步骤包含选择 x86 内部电路仿真断点为该扩充卷标的步骤。

使浮点格式实现指令级规格的设备及方法

技术领域

本发明涉及微电子领域，特别是涉及一种将指令级浮点格式规格应用于现有微处理器指令集结构中的技术。

背景技术

本案对应的美国案主张 2004 年 3 月 18 日提出美国临时申请案 60/554299 为优先权，在此将其并入本案以供参阅。

微处理器自 70 年代早期问世后，已逐渐获得推广，其应用领域已由原本的科技界扩增至商业消费界。其中商业消费界的产品包含桌上及膝上型计算机、视讯游戏控制器及其它诸多家用及商用装置。

随着微处理器使用的爆炸性增长，愈来愈多的产品面临技术上的更高要求，如速度要求、寻址能力要求、快速内存存取能力要求、大操作数要求、一般功能运算种类提升的要求(如浮点、单指令多数据流(SIMD)及条件式移动等)及附加特殊目的运算能力要求(如数字信号处理功能及其它多媒体运算等)。由于这些要求的推动，微处理器设计已使用了相当多的改进技术，如强力管线、超标量结构、高速缓存结构、失绪处理、猝发处理机制、分支预测及臆测执行等。相对于三十年前的产品而言，今天的微处理器实际上已是一相当复杂且功能强大的机器。

与其它产品不同的是，微处理器结构的进展仍有一项重要的受限因素，且该项因素持续对微处理器造成影响。这项因素即是前后代兼容性问题，也是目前的微处理器具有相当复杂度的主因之一。基于市场因素影响，许多制造商在新推出结构中加以新设计时都选择维持其所有性能以使与原有版本兼容。

如今，前后代兼容性问题已相对 x86 兼容微处理器发展时代明显许多。众所周知，现今虚拟模式 32/16 位 x86 微处理器仍可执行 80 年代所制造的

8位实模式应用程序。该种结构仍大量沿用于x86结构中，其目的只是为维持前后代应用及操作模式的兼容性。不过，虽然过去已经将新开发出的结构特征加至现有指令集结构上，但该特征实现所依赖的可程序化指令却变得相当少了。详细的说，目前某些指令集中并无闲置未用的指令可供设计者将较新的特征加入现有结构中。

以x86指令集结构而言，并无另外可用的未定义一位操作码，一位x86操作码中的全部256个操作码皆已为现有指令所用。故现今x86微处理器设计者必须选择加入新特征或维持前后代兼容性。若选择加入新的可程序化特征，那么这些特征必须被指定以操作码值，以使程序设计者运用这些特征。若有闲置操作码值未存于现有指令集结构中，则部份现有操作码值必须重新定义，以达到提供新特征的规格，然而新特征的加入却会使前后代兼容性受到影响。

程序设计者希望在现今的计算机中加入许多特征，但却因为上述原因而不具备可能性，其中一项特征就是对于指令级浮点格式规格的加入。

综上所述，本案发明人已了解到程序设计者在指令级指明浮点运算执行时所用的精度及/或舍入模式的需要，其中该浮点运算为一对应指令所激活。然而，本领域的普通技术人员皆知现有微处理器结构并无该种规格，其一般包含一执行浮点运算的浮点单元，且浮点运算时浮点单元所用的精度及舍入模式在浮点运算相关指令被执行前即先预定于一或多个相关硬件缓存器中。这些相关硬件缓存器在x86兼容微处理器中称作浮点控制字组，故x86兼容微处理器中浮点单元所执行的所有后续运算所用的浮点格式(即精度及舍入模式)为浮点控制字组中各不同字段值所确定。此外，欲对x86结构中浮点单元的精度及/或舍入模式加以改变时，需在一程序流中执行一特殊指令FLDCW。

由于浮点运算本身并非完全准确，故浮点操作数所用及结果所需的精度及/或舍入模式规格对于浮点算法的正确执行有着相当重要的影响，也因此算法中需加入兼容法则以确保能得到正确结果。例如，x86浮点控制字组可受程序化以确定单精度运算等，并使其舍入模式得到的舍入结果最接近但不小于无限准确值。此种在一浮点控制字组中确定的特殊浮点格式固足以真正为一些浮点算法所用，但对于需要不同精度或舍入模式的其它算法却是完全不够的，如程序语言JAVA通常都严格规定需使用单精度操作数，

此为熟习该项技术者皆知者。此外，本领域的普通技术亦知现有编译器一般对浮点控制字组加以设定以确定双精度为浮点运算的预定精度。再者，虽然典型指令集结构都能提供令微处理器自内存加载一新浮点控制字组以改变浮点格式的指令，但该项指令的执行速度非常慢，此乃因为微处理器中所有运算皆需在浮点控制字组改变之前受同步化。在实际使用时，微处理器中运算的同步化指微处理器必须被停止动作，浮点控制字组自内存中加载，且微处理器重新动作，此时该种运算的性能会有一严重性能瓶颈，甚至在只有一个浮点格式改变也是这样。实际上，本案发明人已注意到许多 JAVA 编译器利用间接确定新浮点格式的方式(其实上较快)避开这种性能瓶颈，即在双精度运算格式的浮点单元中执行单精度运算时，编译器利用使浮点单元在双精度模式下执行该等运算以进行该单精度运算、并接着利用储存该结果为所需精度形式及/或舍入模式的方式而将运算结果做舍入处理成单精度者，其中该结果储存于一内存中(多数指令集结构能在内存加载及储存动作进行时使浮点精度及/或舍入模式受到明确指明)。最后，该结果自内存(以所需精度储存)中被载至浮点单元中，以进行后续的运算动作。

上述浮点格式规格“workaround”的说明可见于 Takeshi Ogasawara 等人发表的论文” Optimizing Precision Overhead for x86 Processors”，截自 “Proceedings of the 2nd Java™ Virtual Machine Research and Technology Symposium,” August 1-2, 2002, Useix: San Francisco.此外，技术人员熟知，虽然将一浮点结果写出至内存、并接着将之读回至一浮点单元中的速度不会与执行一指令以加载一新浮点控制字组一般慢，但该种方式仍然会有性能瓶颈产生。

因此，本案发明人已注意到一种使浮点格式规格用于一不需运算同步化却不会有性能变差的浮点运算中的较佳技术确有提出的必要。

因此，需有多个加载浮点控制字组可被设定以指明多个所需浮点格式、并使程序设计者选择该等浮点控制字组之一者以为一相关浮点指令所确定的运算所用，其中相关浮点指令本身即确定了对该等浮点控制字组的一者的选择方式。此外，程序设计者需可直接确定一相关浮点运算的浮点格式，即浮点运算与运算执行时所用的浮点格式皆确定于一单一指令中。不过，虽然该等需求已被述明，但许多指令集结构(包含目前的 x86 指令集结构)

皆无在不牺牲一些现有操作数操作性的条件下提供所需特征的手段或装置。

基于上述需求，一种能将浮点格式特征规格加至一具有完全完整的操作码集的暨存微处理器结构的设备及方法是目前所需求的，其中浮点格式规格特征的加入使得一兼容微处理器得维持执行前后代应用程序的能力，并同时能使程序设计者与编译器具有在指令级上控制浮点格式及相关浮点运算规格的能力。

发明内容

本发明所要解决的问题在于克服上述及其它问题，并用以改善现有技术的缺点。

为实现上述目的，本发明提出一种使微处理器指令集具有超出其现有能力而能提供浮点格式规格。在一实施例中，所提出的一种微处理器设备用以确定一浮点格式在一相关浮点运算执行期间的使用，其包含转换逻辑电路及扩充执行逻辑电路，其中转换逻辑电路将一扩充指令转换为对应微指令，扩充指令则具有一扩充前缀及一扩充前缀卷标。扩充前缀用以确定浮点格式，所述扩充前缀还包括一精度码字段以及一舍入码字段，以分别确定一精度格式以及一舍入格式，所述扩充前缀还具有浮点例外屏蔽掩码，以阻止浮点例外的发生。扩充前缀卷标则指出该扩充前缀，其中该扩充前缀卷标为一微处理器所用的一指令集中在结构上以其它方式确定的操作码。转换逻辑电路包含逃脱指令检测逻辑电路、扩充前缀译码逻辑电路以及指令译码逻辑电路。逃脱指令检测逻辑电路用以检测该扩充前缀卷标。扩充前缀译码逻辑电路耦接至该逃脱指令检测逻辑电路，用以转换该扩充前缀，并用以设定上述对应微指令中的一微操作码扩充字段，其中该微操作码扩充字段用于指定该浮点格式。指令译码逻辑电路，用以设定上述对应微指令中的其它字段，其中该其它字段用以确定该相关浮点运算。扩充执行逻辑电路耦接至转换逻辑电路，并接收对应微指令，且依据该扩充前缀中确定的浮点格式执行相关浮点运算。

本发明公开了一扩充机制，用以对一现有微处理器指令集加以浮点格式规格特征，并具有一扩充指令及一转换器。扩充指令令微处理器根据一浮点格式执行一相关浮点运算，其中该扩充指令包含在该现有微处理器指

令集中一被选出的操作码及尾随的一 n 位扩充前缀。该被选出的操作码指出该扩充指令，而该 n 位元扩充前缀则指出该浮点格式，所述扩充前缀还包括一精度码字段以及一舍入码字段，以分别确定一精度格式以及一舍入格式，所述扩充前缀还具有浮点例外屏蔽掩码，以阻止浮点例外的发生；该转换器接收该扩充指令，并产生一微指令序列，用以令该微处理器依据该浮点格式执行该相关浮点运算。该转换器还包含逃脱指令检测逻辑电路、扩充前缀译码逻辑电路以及指令译码逻辑电路。逃脱指令检测逻辑电路用以检测该扩充前缀卷标。扩充前缀译码逻辑电路耦接至该逃脱指令检测逻辑电路，用以转换该扩充前缀，并用以设定上述对应微指令中的一微操作码扩充字段，其中该微操作码扩充字段用于指定该浮点格式。指令译码逻辑电路，用以设定上述对应微指令中的其它字段，其中该其它字段用以确定该相关浮点运算。

本发明的还公开一种指令集扩充设备，用以提供现有微处理器指令集以指令级浮点格式规格能力，并包含一转换逻辑电路，用以接收逃脱卷标，并用以指出一对应指令的伴随部份用于指定一为微处理器执行的一扩充运算，其中该逃脱卷标为微处理器指令集中一第一操作码部份；该逃脱卷标耦接至一浮点格式确定字段，该浮点格式确定字段为该伴随部份之一，用以指定一待于该扩充运算执行期间所用的一浮点格式，所述浮点格式确定字段包括一精度码字段以及一舍入码字段，以分别确定一精度格式以及一舍入格式，所述逃脱卷标还具有浮点例外屏蔽掩码，以阻止浮点例外的发生；及扩充浮点执行逻辑电路。该转换逻辑电路包含逃脱指令检测逻辑电路、扩充前缀译码逻辑电路以及指令译码逻辑电路。逃脱指令检测逻辑电路用以检测该扩充前缀卷标。扩充前缀译码逻辑电路耦接至该逃脱指令检测逻辑电路，用以转换该扩充前缀，并用以设定上述对应微指令中的一微操作码扩充字段，其中该微操作码扩充字段用于指定该浮点格式。指令译码逻辑电路，用以设定上述对应微指令中的其它字段，其中该其它字段用以确定该相关浮点运算。

本发明的还公开了一种扩充一微处理器指令集以提供一浮点格式指令级规格的方法，其包含下列步骤：提供一扩充指令，该扩充指令包含一扩充卷标及一扩充前缀，其中该扩充卷标为该微处理器指令集中的一第一操作码，所述扩充前缀还包括一精度码字段以及一舍入码字段，以分别确定

一精度格式以及一舍入格式，所述扩充前缀还具有浮点例外屏蔽掩码，以阻止浮点例外的发生；检测该扩充前缀卷标；转换该扩充前缀，并用以设定上述对应微指令中的一微操作码扩充字段，其中该微操作码扩充字段用于指定该浮点格式；设定上述对应微指令中的其它字段，其中该其它字段用以确定该相关浮点运算；通过该扩充前缀及该扩充指令的其余部份指定一待执行运算，其中该指定步骤包含先确定该运算所用的精度格式与舍入格式；及根据该确定动作执行该运算。

本发明可保证兼容性的同时，使浮点格式实现指令级规格。

附图说明

图 1 为一现有微处理指令格式的方块图；

图 2 为一指令集结构中指令映像至图 1 指令格式中一 8 位操作码字节的位逻辑电路状态的方式的表格；

图 3 为本发明的扩充指令格式的方块图；

图 4 为本发明的扩充结构特征映像至一 8 位扩充前缀实施例的逻辑状态的方式的表格；

图 5 为本发明中浮点格式加以规格化所用的管线微处理器的方块图；

图 6 为本发明中对浮点格式加以规格化所用的一扩充前缀实施例的方块图；

图 7 为本发明中对浮点格式加以规格化所用的扩充前缀另一实施例的方块图；

图 8 为图 5 的微处理器内转换级逻辑电路的方块图；

图 9 为图 5 微处理器中扩充执行逻辑电路的方块图；

图 10 为图 5 微处理器中扩充执行逻辑电路另一实施例的方块图；以及

图 11 为本发明的转换及执行指令的方法的流程图，且该方法用以对指令级浮点格式加以规格化。

其中，附图标记：

100	微处理器指令格式	101	前缀
102	操作码	103	地址确定码
200	8 位操作码地图	201	指令码
202	操作码	300	扩充指令格式

301	前缀	302	操作码
303	地址确定码	304	逃脱卷标
305	扩充浮点格式确定字段前缀	400	扩充前缀地图
401	浮点格式确定字段	501	提取逻辑电路
500	具浮点格式规格的微处理器	502	指令高速缓存
503	指令贮列	504	转换逻辑电路
505	扩充转换逻辑电路	506	微指令贮列
507	执行逻辑电路	508	扩充执行逻辑电路
600	扩充前缀	602	浮点控制字组字段
603	闲置字段	604	浮点控制字组 1
605	浮点控制字组 2	606	浮点控制字组 3
607	浮点控制字组 4	700	扩充前缀
701	舍入码字段	702	精度码字段
703	闲置字段	704	浮点控制字组
800	转换级逻辑电路	801	电源开启状态
802	机器确定缓存器	803	扩充特征字段
804	指令缓冲器	805	转换逻辑电路
806	转换控制器	807	关闭信号
808	逃脱指令检测器	809	扩充前缀译码器
810	指令译码器	811	只读存储器
812	微指令缓冲器	813	操作码扩充字段
814	微操作码字段	815	目的字段
816	来源字段	817	取代字段
900	扩充执行逻辑电路	901	运算位
902	微指令缓存器	903	操作码扩充字段
904	微操作码字段	905	运算位
906	浮点控制字组	910	浮点结果缓冲器
909	扩充浮点执行逻辑电路	1001	运算位 1
1000	扩充执行逻辑电路	1002	微指令缓存器
1003	操作码扩充字段	1004	微操作码字段
1005	运算位 2	1006	浮点控制字组

1009	扩充浮点执行逻辑电路	1010	浮点结果缓冲器
1100	浮点格式的指令级规格	1011	总线

步骤 S1102 开始

步骤 S1104 提取下一指令

步骤 S1106 判断扩充逃脱码

步骤 S1108 译码扩充前缀中浮点格式确定字段

步骤 S1110 指定微指令序列中扩充字段浮点格式

步骤 S1112 译码/转换前缀、操作码及地址确定码字段

步骤 S1114 设定微指令序列中微操作码及操作数字段

步骤 S1116 提供微指令序列至贮列中以为执行所用

步骤 S1118 取得执行用的扩充微指令

步骤 S1120 根据指定的浮点格式执行预定运算并产生结果

步骤 S1122 完成

具体实施方式

通过以下说明可使本领域的普通技术人员借之制造及使用本发明。以下介绍将针对一特定应用及其需求进行说明，不过本领域的普通技术人员可轻易对该实施例进行各种变更，该变更仍以本发明公开的内容为基本原理，并因此得到其它的实施例。因此，本发明的范围不仅限于所述实施例，而应以其原理及技术特征所对应的最大范围来确定。

鉴于背景技术一节对于现有微处理器用以扩充结构特征至具有超出原本相关指令集能力的技术的讨论，以下将通过图 1 及图 2 所示的现有技术中的范例进行说明，并特别针对现今微处理器设计者常碰到的问题加以说明。一方面，该二图相关说明希望将近期发展的结构特征加入微处理设计当中；另一方面，市场因素又要求该技术必须能应用于前后代应用程序中。在图 1 及图 2 中，完整定义的操作码地图的提出使得结构不需再定义新操作码，因此设计者只能选择加入新特征及牺牲一定程度的兼容性，或放弃研发出的结构改进，以确保执行较早期的应用程序。在现有技术的探讨后，以下配合图 3 至图 11 介绍本发明。通过将现存但已老式的操作数作为扩充指令的前缀卷标，使使用者克服完全完整指令集结构的限制，并使其能对浮点格式加以规格化，且同时使该指令前后兼容性较高。

请参阅图 1，其所示为一相关现有微处理器指令格式 100 的方块图。该相关现有微处理器指令格式 100 具有一数目可变的指令部份 101-103，且各被设以一特定值，共同构成微处理器的一特定微处理器指令格式 100。特定微处理器指令格式 100 可使微处理器执行特定运算，如对二运算相加或将操作数移至一内部寄存器中。最典型的是，微处理器指令格式 100 的操作码 102 部份用于指定待执行的特定运算，而可选择的地址确定码 103 部份则跟随在前缀 101 后，其中前缀 101 用于指定关于特定运算的其它数据，如运算执行方式及运算所置放处等等数据。微处理器指令格式 100 还使得程序设计者得以前缀 101 部份取代操作码 102 为用，前缀 101 使该已确定的结构特征在操作码 102 所指定的运算期间得以执行。一般说来，该结构特征可使用于指令集中操作码 102 所对应的多数运算上。如现有微处理器中前缀 101 使运算可在使用不同位数(如 8 位及 32 位)条件下执行，故该处理器被设为一预定操作数大小(如 32 位)，且前缀 101 被设计于其对应指令集当中，以令使用者得以逐项指令的方式改变该预定操作数大小(如执行 16 位运算)，不过运算位数可选择只是构成现有微处理器中操作码 102 对应的众多运算(如加法、减法、乘法及布尔逻辑等)的结构特征的一例。

图 1 所示为熟识的 x86 微处理器指令格式 100，其可应用于所有现有 x86 兼容的微处理器上。详细的说，x86 微处理器指令格式 100(x86 指令集结构)使用 8 位前缀 101、8 位操作码 102 及 8 位地址确定码 103 字段。x86 微处理器指令格式 100 同样具有多个前缀 101，且其中二个改变了 x86 微处理器的预定地址/数据位数(即操作码状态 66H 及 67H)，另一个使微处理器依据不同的转换规则(即前缀值 0FH，其使转换依所谓的 2 位操作码规则进行)解译一后续操作码 102，其它前缀 101 则使相关运算重复进行直至重复条件满足为止(即 REP 码：F0H、F2H 及 F3H)。

现请参阅图 2，其为一说明指令集结构中指令映像至图 1 指令格式中一 8 位操作码字节的位逻辑电路状态的方式的表格。该表格显示一 8 位操作码地图 200 范例，其与一 8 位操作码 102 部份的 256 个值相关，其中该 8 位操作码 102 部份具有微处理器指令码 201。8 位操作码地图 200 将一操作码 102 部份的特定值(如 02H)映像成一指令码 201(即指令 I02 及 201)，此时技术人员皆知操作码值 14H 映像为 x86 进位加法(ADC)指令操作码，并令一 8 位中间操作码加至结构寄存器 AL 的内容中。熟习该项技术者都知道，前

文提到的 x86 前缀 101(即 66H、67H、0FH、F0H" F2H 及 F3H)实际上是为确定下一操作码 102 部份指定的运算所加的特定结构的内容扩充，如前一操作码 14H(一般而言为上述的进位加法操作码)在前文为 0FH 时构成一执行一 UNPCKLPS 运算而非进位加法的 x86 处理器，此乃因 x86 处理器在遇上前缀 0FH 时会有不同转换规则的缘故。由于微处理器中指令转换/译码逻辑电路依序对微处理器指令格式 100 的部份 101-103 加以解译，故上述 x86 范例等特征会部份被激活。因此，定操作码值作为指令集结构中的前缀 101 已令微处理器设计者将相当数量的最先进结构特征加至一兼容微处理器设计中，同时不会使不利用该特定操作码状态的较早期程序的执行受阻。例如，一不使用 x86 操作码的早期程序 0FH 仍能在现有 x86 微处理器上执行，且新一代应用程序可利用将 x86 操作码 0FH 当作前缀 101 而被近期已加至结构中的相当数量的 x86 结构特征所执行，如此得执行单指令多数据流(SIMD)运算及条件移动运算。

过去，最新结构特征的加入已通过将可用/闲置指令码 201 当作前缀 101(也称为结构特征卷标/指示器或逃脱指令)的方式实现，然而许多微处理器指令格式 100 在提升上已再无更多的指令集资源可用，其相当直接的理由即为所有的可用/闲置操作码状态皆已受利用。即，8 位操作码地图 200 中的所有的操作码值在结构中皆已受到预定的使用，且当所有的可用操作码值已被指定为操作码 102 部份或前缀 101 部份时，已再无操作码值可接受新特征的加入。此项大问题是现今微处理器结构所共有的，并使设计者必须在于设计中加入结构特征及维持与较旧程序的兼容性之间做出选择。

图 2 所示的指令码 201 为广义方式(即 I24 及 I86)而非特定方式(即加法进位、减法或互斥)，这是因为完全被指定的 8 位操作码地图 200 目前不接受更多最新结构的加入之故。另，虽图 2 所示例中提及 8 位操作码 102 部份，但本领域的技术人员都知道操作码 102 的位数是无关的，该 8 位大小只是为教示全 8 位操作码地图 200 所用。因此，完全填满的 6 位操作码地图具有 64 个结构上定义的指令码 201 (前缀)，并无法提供现有/闲置操作码值以供扩充用。

然而，本案发明人已观察到操作码全部被使用的 8 位操作码地图 200 中特定指令码 201 在多种执行于兼容微处理器上的应用程序中皆得到应用，他们还持续指出一些非结构上定义的操作码 202 并未使用在能为微处理器

执行的应用程序中。图 2 中指令 IF1 操作码 202 即为此种特殊范例。事实上，非常相同的操作码 202 值 (即 F1H)映像于目前所有应用程序均未使用过的 x86 指令集结构中的一有效操作码 202，而未使用过的 x86 操作码 202 虽为一可令 x86 微处理器执行一特定结构的运算的有效 x86 操作码 202，但其未被可在现有 x86 微处理器上执行的应用程序所使用。该特殊 x86 操作码 202，被称作内部电路仿真断点(即 ICE BKPT，操作码值 F1H)，其在早期专门使用于目前已不存在的微处理器仿真设备中。ICE BKPT 操作码 202 在过去从未用于内部电路仿真器外的应用程序中，而早期使用 ICE BKPT 操作码 202 的内部电路仿真设备形式已淘汰。因此，本案发明人已确认一种在 x86 系列中指令完全受到预先指定的 8 位操作码地图 200 内的一种机制，藉由该种机制本案发明人可开发一种有效但已淘汰的操作码 202 来让最新的结构特征加入至一微处理器设计中，同时不牺牲前后代兼容性。在一指令已完全受预用的 8 位操作码地图 200 中，本发明以一结构上确定但仍未使用过的操作码 202 当作一为下一 n 位前缀所用的指示卷标，故能让微处理器设计者加入高达 2^n 个最新开发出结构特征至一现有微处理器设计中，同时能维持所有前后代软件的完全兼容性。

本案发明人利用了前缀卷标/扩充前缀观念，其方式为提供一 n 位元浮点格式确定字段前缀，借以令程序设计者能通过微处理器预定执行一传统运算(如加法、减法、布尔运算、操作数处理等)，并利用同一指令确定待于下一浮点运算(若与该 n 位前缀相关的指令预定执行非浮点运算时)或另一浮点运算期间所当使用的浮点格式，其中该另一浮点运算指包含该 n 位前缀的相同指令所定义的浮点运算工作。在其它实施例中，还具有程序设计者可经由该 n 位前缀直接确定浮点格式、或确定多个浮点控制自组之一者的特点，其中该多个浮点控制字组之一者确定一浮点格式。以下，本发明的说明将配合图 3 至图 11 的配合说明继续说明如下。

请参阅图 3，其为本发明的扩充指令格式的方块图。与图 1 中微处理器指令格式 100 非常相近的是，扩充指令格式 300 具有指令部份 301-305，其位数可变，且每一部分皆被设定为一特定值，以共同构成微处理器之一特定扩充指令格式 300。该特定扩充指令格式 300 可使微处理器执行诸如两操作码的相加或将一操作码移至微处理器中之一寄存器中之类特定处理动作。典型的，扩充指令格式 300 中一操作码 302 部份用于指定该欲执行的

特定处理动作，且选择性地址确定码 303 部份跟随预定的其它与该特定处理动作相关的信息出现，其中该其它信息可为该处理动作的执行方式及操作数位置等。扩充指令格式 300 令程序设计者得以在前缀 301 部份加以一操作码，以在由操作码 302 所指定的特定处理动作执行期间将现存结构特征加入。

不过，本发明的扩充指令格式 300 为图 1 中微处理器指令格式 100 的一超集(superset)，其具有两个外加部份 304, 305。该二外加部份 304, 305 具有可选择性，其作为一指令扩充部而取代一经格式化的扩充指令格式 300 的所有其它部份 301-303，其目的在于让一待使用的浮点格式的规格在一或多个浮点运算执行期间得到使用，并用以加入指令级浮点规格特征，这是一全部指令都已被使用的指令集结构所没有提供的。选择性部份 304, 305 为一逃脱卷标 304 及一扩充浮点格式确定字段前缀 305，其中扩充浮点格式确定字段前缀 305 是以另一种方式在微处理器指令集结构中确定的操作码。在一 x86 实施例中，逃脱卷标 304(或称扩充指令卷标)为操作码状态 F1H，并为先前所使用的内部电路仿真断点。逃脱卷标 304 向微处理器逻辑电路指出，后续会出现扩充浮点格式确定字段前缀 305 或扩充特征确定字段，此时扩充浮点格式确定字段前缀 305 用于指定在一或多相关浮点运算执行期间当中使用的浮点格式。在一实施例中，逃脱卷标 304 指出一对应扩充指令格式 300 的伴随部份 301-303, 305 预定一有待被微处理器执行的扩充运算。n 位经扩充浮点格式确定字段前缀 305(或称浮点格式确定字段)，在一实施例中为多个浮点控制字组中之一，此时该每个浮点控制字组皆预定一种有待于一浮点运算执行期间使用的浮点格式，其中该浮点运算包括可由扩充指令格式 300 所确定或由一下一指令所确定。在一不同实施例中，该 n 位扩充浮点格式确定字段前缀 305 直接指出一有待于扩充指令格式 300 或一下一指令确定的浮点运算执行期间所使用的浮点格式。

为对本发明的浮点格式规格技术做详细说明，取一现存指令集结构的一个逃脱卷标 304 及一扩充浮点格式确定字段前缀 305 构成一指令扩充部。其中该选出操作码系指令作为一逃脱卷标 304，而扩充指令格式 300 为一扩充特征指令 (即，其预定微处理器结构的扩充部份)，而扩充浮点格式确定字段前缀 305 用于确定一在相关浮点运算执行期间所当用的浮点格式，其中扩充浮点格式确定字段前缀 305 在一实施例中为 8 位。

现请参阅图 4，其为本发明的浮点格式规格扩充部份特征映像至一 8 位扩充前缀实施例的逻辑状态的方式的表格。与图 2 讨论的 8 位操作码地图 200 相当，图 4 的扩充前缀地图 400 显示一 8 位浮点格式确定字段扩充前缀地图 400，该扩充前缀地图 400 与一具对应浮点格式确定字段 401 的 8 位扩充浮点格式确定字段前缀 305 部份的 256 个值相关(如 E34 及 E4D 等)。其中该浮点格式确定字段 401 将会在一相关浮点运算执行期间被使用到。在 x86 例中，本发明的 8 位扩充浮点格式确定字段前缀 305 系用以为浮点格式确定字段 401(即 E00-EFF)所用，但其未为现今 x86 指令集结构所提供。

图 4 中扩充浮点格式确定字段 401 的特征绘示以较上位概念而非较明确的方式，因为本发明的技术可用于多种不同结构的浮点格式确定字段 401 部及特定指令集结构中。本领域的普通技术人员皆能了解许多上述及其它不同结构特征的浮点格式确定字段 401，可将本案的逃脱卷标 304/扩充浮点格式确定字段前缀 305 技术增添至一现存指令集。图 4 的 8 位前缀实施例提供 256 个不同特征的浮点格式确定字段 401，而一 n 位前缀实施例可对 2^n 个不同特征的浮点格式确定字段 401 加以程序化。

在对应现有微处理器所共享的浮点格式规格及表示法的实施例中，格式结合可确定操作数及结果精度等参数，如单精度、双精度、扩充精度及舍入模式等，其中舍入模式可为舍至最近者、往下舍去(往负无限大方向舍去)、往上舍去(往正无限大方向舍去)及舍至零(截尾)。不过，上述特定浮点格式规格参数并不限定本发明的范围为一特定浮点格式确定字段组，而仅用以作为本发明的扩充浮点格式确定字段前缀 305 的编码方式。普通技术人员将能了解一特定扩充浮点格式确定字段前缀 305 的设计是依对应浮点格式规格的表示及在一对应微处理中的存储方式而定。

现请参阅图 5，其为本发明的管线微处理器的方块图。具浮点格式规格的微处理器 500 具有三个主要操作级，即提取、转换及执行，其中提取级具有提取逻辑电路 501，其自一指令高速缓存 502 或外部内存取得指令，所取得的指令经一指令贮列 503 送至转换级；转换级具有转换逻辑电路 504，该转换逻辑电路耦接至一微指令贮列 506，并包含扩充转换逻辑电路 505；执行级具有执行逻辑电路 507，执行逻辑电路 507 中具有扩充执行逻辑电路 508。

在实际操作时，提取逻辑电路 501 从指令高速缓存 502/外部内存提取

到的具本发明的格式的指令，并将这些指令以执行次序置于该指令贮列 503 中。接着，上述指令可自指令贮列 503 中取得，并被送至转换逻辑电路 504 里，具浮点格式规格的微处理器 500 执行上述指令所指定的运算或动作。转换逻辑电路 504 转换/译码每一上述指令并生成一对应微指令序列，用以令具浮点格式

规格的微处理器 500 执行上述指令所指定的运算或动作。经扩充转换逻辑电路 505 检测上述具一本发明的扩充前缀卷标的指令，并提供对对应浮点格式确定字段前缀的转换/译码。在一 x86 实施例中，扩充转换逻辑电路 505 被设计为用以检测--值为 F1H 的扩充前缀卷标，其为 x86 内部电路仿真断点(ICD BKPT)操作码。微指令字段设于微指令贮列 506，用以确定一应当于一相关浮点运算期间使用的浮点格式。

微指令自微指令贮列 506 送至执行逻辑电路 507，其中该扩充执行逻辑电路 508 检测具浮点格式规格的微指令，且该特征能否启动由微指令字段指出。扩充执行逻辑电路 508 建立由该微指令所指定的浮点格式，并执行该微指令所指定的运算或动作，以产生一对应结果。若微指令所指定的运算或动作为一浮点运算，则扩充执行逻辑电路 508 执行该浮点运算，并根据该微指令指定的浮点格式产生对应结果。

本领域的普通技术人员将能了解图 5 所描述的具浮点格式规格的微处理器 500 为现今管线微处理器的一种简化代表形式，实际上现今具浮点格式规格的微处理器 500 包含至 20-30 个管线级，不过这些管线级一般可分作如图 5 所示的的三级群，故图 5 的方块图的提出用以教示本发明上述实施例的必要组件，具浮点格式规格的微处理器 500 中其它不属本发明讨论的范围的部分则未添加图标，以求图面的简洁。

现请参阅图 6，其为本发明中对浮点格式加以规格化所用的一扩充前缀实施例的方块图。该扩充前缀 600 为一 8 位扩充前缀 600，并具有一浮点控制字组字段（FPSEL） 602，该浮点控制字组字段 602 包含第 0、1 位及一闲置字段（SPARE）603，该闲置字段（SPARE）603 包含第 2 至 7 位。根据浮点控制字组字段 602，该四浮点控制字组 604-607 中之一将用于一相关浮点运算的执行，且每个浮点控制字组（604-607）都包含用以利用上述的方式确定的一对应浮点式的字段(未显示)。在另一实施例中，并未提供闲置字段 603，且浮点控制字组字段 602 为 8 位，故任意数目的浮点控制字组变

为多至最大数目 256。除确定浮点格式(精度及舍入模式)外，上述每个浮点控制字组皆额外有浮点例外屏蔽掩码，用以阻止典型浮点例外的发生，该等例外包含精度例外、下溢、零除、不正常操作数、不成立运算及现有微处理器中的种种例外。

图 7 为本发明中对浮点格式规格加以规格化所用的扩充前缀另一实施例的方块图。该扩充前缀 700 具有 2 位精度码字段（PC）702、一 2 位舍入码字段（RC）701 及 4 位闲置字段 703。根据舍入码字段 701 及精度码字段 702 的值，四种精度格式之一及四种舍入模式之一可被直接确定用于一相关浮点运算执行，其中该直接确定的精度格式及舍入模式在实际操作中取代处理器一浮点控制字组 704 目前所确定的舍入模式及精度格式，不过其它数据(如浮点例外屏蔽位)为该浮点格式字组所提供。字段 701-702 的位数可交替增加，以根据所提供的一特定指令集结构而可容置另增的舍入模式及精度。

现请参阅图 8，其为图 5 的微处理器内转换级逻辑电路详细的方块图。该转换级逻辑电路 800 具有一指令缓冲器 804，该指令缓冲器 804 提供一本发明的扩充指令至转换逻辑电路 805，并被耦接至一机器确定缓存器 802 中。其中该机器确定缓存器 802 具有一扩充特征字段 803。转换逻辑电路 805 具有一转换控制器 806，该转换控制器 806 提供一关闭信号 807 至一逃脱指令检测器 808 及一扩充前缀译码器 809，其中逃脱指令检测器 808 被耦接至扩充前缀译码器 809 及一指令译码器 810，扩充前缀译码器 809 及指令译码器 810 对应一控制只读存储器(ROM) 811，其中该只读存储器 811 存储对应部份的上述扩充指令的微指令序列样本。转换逻辑电路 805 还具有一微指令缓冲器 812，该微指令缓冲器 812 具有一操作码扩充字段 813、一微操作码字段 814、一目的字段 815、一来源字段 816 及一取代字段 817。

在实际操作中，在微处理器被供以验源期间，机器确定缓存器 802 中扩充特征字段 803 的状态是经由电源开启状态 801 所控制，以指出该特定微处理器是否能根据本发明的方式转换及执行扩充指令。在一实施例中，电源开启状态 801 由一特征控制缓存器(未显示)控制，其中该缓存器读取一熔线阵列(未显示)或读取在该部份制造前间设定的类似组件。机器确定缓存器 802 提供扩充特征字段 803 的状态至转换控制器 806，转换控制逻辑电路根据浮点格式规格转换规则，对是否对指令缓冲器 804 送出的指令加

以转换进行控制。该种控制特征的存在使得监督应用程序(如 BIOS)得以启动/关闭微处理器的扩充执行特征；若扩充特征被关闭，则具被选择为扩充特征卷标的运算状态的指令将依照现有转换规则转换。在一操作码状态 F1H 被选为卷标的 x86 实施例中，以现有转换方式所得到的 F1H 可能会形成不合理的指令例外，而卷标在扩充转换规则的处理下则会为逃脱指令检测器 808 所检测。此时，逃脱指令检测器 808 会在下一扩充浮点格式确定字段前缀期间利用扩充前缀译码器 809 关闭指令译码器 810 的动作，并使指令译码器 810 以转换/译码该扩充指令的其它部份。欲获取对应的微指令序列样本时，只要以数个特定指令即可通过存取控制只读存储器 811 实现。微指令缓冲器 812 的操作码扩充字段 813 由扩充前缀译码器 809 所设定，用以指定浮点格式确定字段前缀所确定的浮点格式。在一实施例中，该操作码扩充字段 813 用于确定该浮点控制字组。在另一不同实施例中，该操作码扩充字段 813 则直接用于确定浮点格式。其余缓冲器字段 814-817 用于确定对应运算或处理动作，并为指令译码器 810 所设定。经设定的微指令缓冲器 812 被送至一微指令贮列(未显示)中，用于执行微处理器进行的后续动作。

现请参阅图 9，其为图 5 微处理器中扩充执行逻辑电路的方块图。扩充执行逻辑电路 900 具有扩充浮点执行逻辑电路 909，该扩充浮点执行逻辑电路 909 耦接至多个浮点控制字组 906 及一浮点结果缓冲器 910。二操作数 OPERAND1 及 OPERAND2 由扩充浮点执行逻辑电路 909 所取得，其在一实施例中用以确定一包含相关浮点运算执行期间所用的浮点操作数的扩充浮点执行逻辑电路 909 的浮点寄存器(未显示)。一微指令缓存器 902 提供一微指令至扩充浮点执行逻辑电路 909，并具有一操作码扩充字段 903 及一微操作码字段 904。

在实际操作中，扩充微指令在一使用浮点格式规格特征的扩充指令被依本发明的方式转换成一微指令序列时，被送至扩充执行逻辑电路 900，且于缓存器中可使用的运算位 901 及 905 (或运算位寄存器确定字段)一起经由微指令缓存器 902 送至扩充执行逻辑电路 900 中。操作码扩充字段 903 确定该多个浮点控制字组 906 中的一特定者在一相关浮点运算执行期间被使用。在一实施例中，该相关浮点运算被微操作码字段 904 所确定。因此，扩充浮点执行逻辑电路 909 执行相关浮点运算，并根据该多个浮点控制字

组 906 的该特定者确定的浮点格式以产生浮点结果，所产生的浮点结果则被送至浮点结果缓冲器 910。在另一实施例中，该相关浮点运算不为微操作码字段 904 所确定。因此，扩充浮点执行逻辑电路 909 选择一为该多个浮点控制字组 906 的该特定者确定的浮点格式，并使用该特定者确定的浮点格式，并使用该格式以执行后续所需进行的浮点运算。

现请参阅图 10，其为图 5 微处理器中扩充执行逻辑电路另一实施例的方块图。扩充执行逻辑电路 1000 具有扩充浮点执行逻辑电路 1009，该扩充浮点执行逻辑电路 1009 被耦接至一浮点控制字组 1006 及一浮点结果缓冲器 1010。二操作数 OPERAND1 及 OPERAND2 为扩充浮点执行逻辑电路 1009 自运算位 1001,1005 缓存器取得。在一实施例中，运算位 1001,1005 缓存器可确定包含一相关浮点运算执行期间所需使用的浮点操作数的扩充浮点执行逻辑电路 1009 内的浮点缓存器(未显示)。一微指令缓存器 1002 提供一微指令至扩充浮点执行逻辑电路 1009，并具有一操作码扩充字段 1003 及一微操作码字段 1004。

在实际操作中，扩充指令在一使用浮点格式规特征的扩充指令被转换成为一微指令序列时，扩充微指令被送至扩充执行逻辑电路 1000，且与缓存器中可使用的运算位 1001 及 1005 (或运算位存器确定字段)一起经由微指令缓存器 1002 送至扩充执行逻辑电路 1000 中。操作码扩充字段 1003 用于直接确定一相关浮点运算执行期间所使用的浮点格式。在一浮点格式包含浮点精度及舍入模式的实施例中，其它为一相关浮点运算执行所需的规格参数(如浮点例外屏蔽) 为浮点控制字组 1006 经由总线 1011 送至扩充浮点执行逻辑电路 1009。在一仅浮点精度为操作码扩充字段 1003 部所确定的不同实施例中，当时为一相关浮点运算执行所需的舍入模式及其它格式参数为浮点控制字组 1006 经由总线 1011 送至扩充浮点执行逻辑电路 1009。其它实施例则利用操作码扩充字段 1003 构成的联合浮点格式参数规格及利用浮点控制字组 1006 而提供其它格式参数。在一实施例中，该相关浮点运算为微操作码字段 1004 所确定，故扩充浮点执行逻辑电路 1009 执行相关浮点运算，并根据操作码扩充字段 1003 及浮点控制字组 1006 直接确定的浮点格式产生浮点结果(必要时)，且该浮点结果被送至浮点结果缓冲器 1010。在另一实施例中，该相关浮点运算不为微操作码字段 1004 所确定，故扩充浮点执行逻辑电路 1009 选择一为操作码扩充字段 1003 及浮点控制

字组 1006 所确定的浮点格式(必要时)，并利用该格式执行后续需进行的浮点运算。

现请参阅图 11，其为本发明的转换及执行指令的方法的流程图，且该方法用以对指令级的浮点格式进行规格化。该流程始于开始（步骤 S1102），其中一为具浮点格式规格特征的扩充指令程序被送至一微处理器。接着，提取下一指令（步骤 S1104）。

在步骤 S1104 处，一下一指令自高速缓存/内存中被提出，接着判断扩充逃脱码（步骤 S1106）。

在步骤 S1106 处，于提取下一指令（步骤 S1104）所提得的该下一指令被用以判定其中是否包含一扩充逃脱卷标/码。若否，则行进译码/转换前缀、操作码及地址确定码字段（步骤 S1112）；若是，译码该扩充前缀中的浮点格式确定字段（步骤 S1108）。

在步骤 S1108 处，由于一扩充逃脱卷标已在步骤 S1106 时被检测得，故一扩充确定字段前缀被执行以转换/译码工作，以判定一有待于一相关浮点运算执行期间得到使用的浮点格式。接着，指定微指令序列中扩充字段浮点格式（步骤 S1110）。

在步骤 S1110 处，一微指令序列的对应字段被设定以指出该扩充前缀所指定的浮点格式。接着，进行译码/转换前缀、操作码及地址确定码字段（步骤 S1112）。

在步骤 S1112 处，指令的其它部份(如前缀部份、操作码及地址确定码字段)被加转换/译码，以判定该待执行运算或动作及相关的操作数属性。接着，设定微指令序列中微操作码及操作数字段（步骤 S1114）。

在步骤 S1114 处，对一微指令序列的其它字段进行设定以指定该经确定运算或动作及其操作数规格。接着，提供微指令序列至贮列中以为执行所用（步骤 S1116）。

在步骤 S1116 处，微指令序列(包含步骤 S1110 处设定的操作码扩充字段及步骤 S1114 内设定的其它字段)被送至一微指令贮列中，以为微处理器所执行。接着，取得执行用的扩充微指令（步骤 S1118）。

在步骤 S1118 处，微指令序列为本发明的扩充执行逻辑电路所取得。接着，流程行进至根据指定的浮点格式执行预定运算并产生结果（步骤 S1120）。

在步骤 S1120 处，扩充执行逻辑电路执行预定的运算或动作，并依据 1110 指定的浮点格式产生结果。接着，流程行进至完成（步骤 S1122）。

在步骤 S1122 处，本发明的方法完全执行完毕。

虽然本发明的目的、特征及优点已详述如上，但本发明还包含其它实施例。例如，本发明在一完全指定的指令集结构中使用一单一、未使用操作码状态作为一卷标、以指出一下一扩充特征前缀的技术已说明于上，本发明在任何一方面的范围皆不限于完全指令的指令集结构、或未使用的指令或独立卷标。相反地，本发明可适用于未被完全指定的指令集、具有已使用操作码的实施例及使用过一指令卷标的实施例。对不存在未使用操作码状态的指令集结构而言，本发明的一实施例包含选择一目前被用作为逃脱卷标的操作码状态的步骤，其中该选择标准由市场因素所决定；另一不同实施例则包含使用一特定操作码组合为该卷标的步骤，其中特定操作码组合可为连续出现的操作码 7FH 组合等。因此，本发明的基本特性体现了一后续有一 n 位扩充前缀的卷标序列的使用，以令程序设计者可排除对对应一扩充指令的其它部份所确定的运算或动作执行的结果加以回写的可能性，其中该结果的回写会于符合一特定标准时成立。

此外，本发明的技术已通过一具有单一精度、双精度、双扩充精度及上述舍入模式等的常用浮点格式确定字段组的微处理器说明如上，因该种微处理器为现今普遍使用者，但不代表本发明仅能用于上述字段种类。

再者，虽然上述通过一种微处理器的设定公开本发明及其目的、特征及优点，但本领域的普通技术人员皆了解其范围超越微处理器结构的界线，应包含所有形式的可程序化装置，如信号处理器、工业控制器、阵列处理机等。

本领域的普通技术人员可轻易以上述内容作为基础，进行进一步设计或修改，该改变及代换皆不脱离发明附于后文的权利要求书所定义的范围。

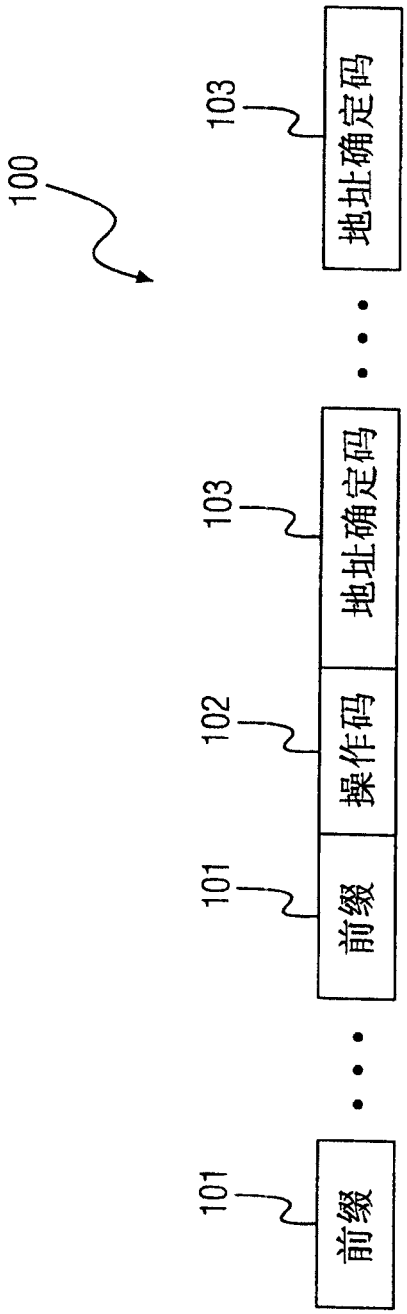


图1

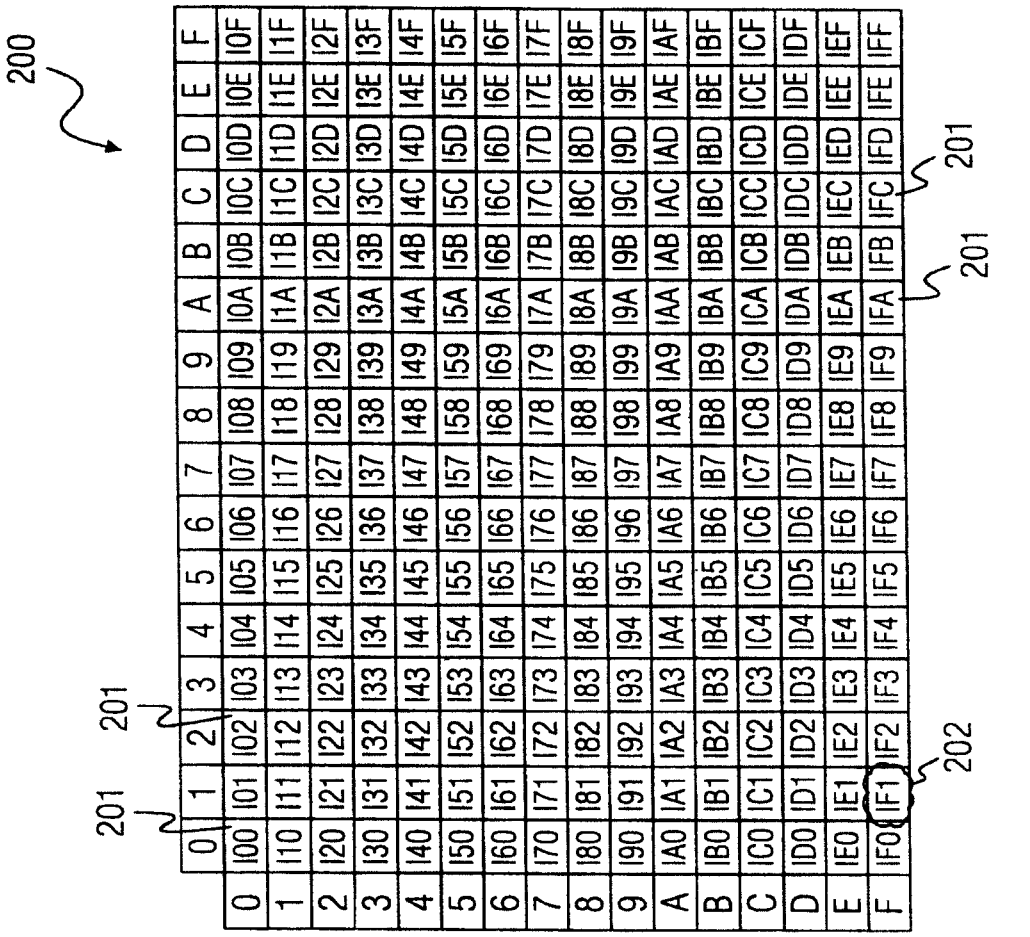

200
201
201
201
202
0 1 2 3 4 5 6 7 8 9 A B C D E F
0 I00 I01 I02 I03 I04 I05 I06 I07 I08 I09 I0A I0B I0C I0D I0E I0F
1 I10 I11 I12 I13 I14 I15 I16 I17 I18 I19 I1A I1B I1C I1D I1E I1F
2 I20 I21 I22 I23 I24 I25 I26 I27 I28 I29 I2A I2B I2C I2D I2E I2F
3 I30 I31 I32 I33 I34 I35 I36 I37 I38 I39 I3A I3B I3C I3D I3E I3F
4 I40 I41 I42 I43 I44 I45 I46 I47 I48 I49 I4A I4B I4C I4D I4E I4F
5 I50 I51 I52 I53 I54 I55 I56 I57 I58 I59 I5A I5B I5C I5D I5E I5F
6 I60 I61 I62 I63 I64 I65 I66 I67 I68 I69 I6A I6B I6C I6D I6E I6F
7 I70 I71 I72 I73 I74 I75 I76 I77 I78 I79 I7A I7B I7C I7D I7E I7F
8 I80 I81 I82 I83 I84 I85 I86 I87 I88 I89 I8A I8B I8C I8D I8E I8F
9 I90 I91 I92 I93 I94 I95 I96 I97 I98 I99 I9A I9B I9C I9D I9E I9F
A IA0 IA1 IA2 IA3 IA4 IA5 IA6 IA7 IA8 IA9 IAA IAB IAC IAD IAE IAF
B IB0 IB1 IB2 IB3 IB4 IB5 IB6 IB7 IB8 IB9 IBA IBB IBC IBD IBE IBF
C IC0 IC1 IC2 IC3 IC4 IC5 IC6 IC7 IC8 IC9 ICA ICB ICC ICD ICE ICF
D ID0 ID1 ID2 ID3 ID4 ID5 ID6 ID7 ID8 ID9 IDA IDB IDC IDD IDE IDF
E IE0 IE1 IE2 IE3 IE4 IE5 IE6 IE7 IE8 IE9 IEA IEB IEC IED IEE IEF
F IF0 IF1 IF2 IF3 IF4 IF5 IF6 IF7 IF8 IF9 IFA IFB IFC IFD IFE IFF

图2

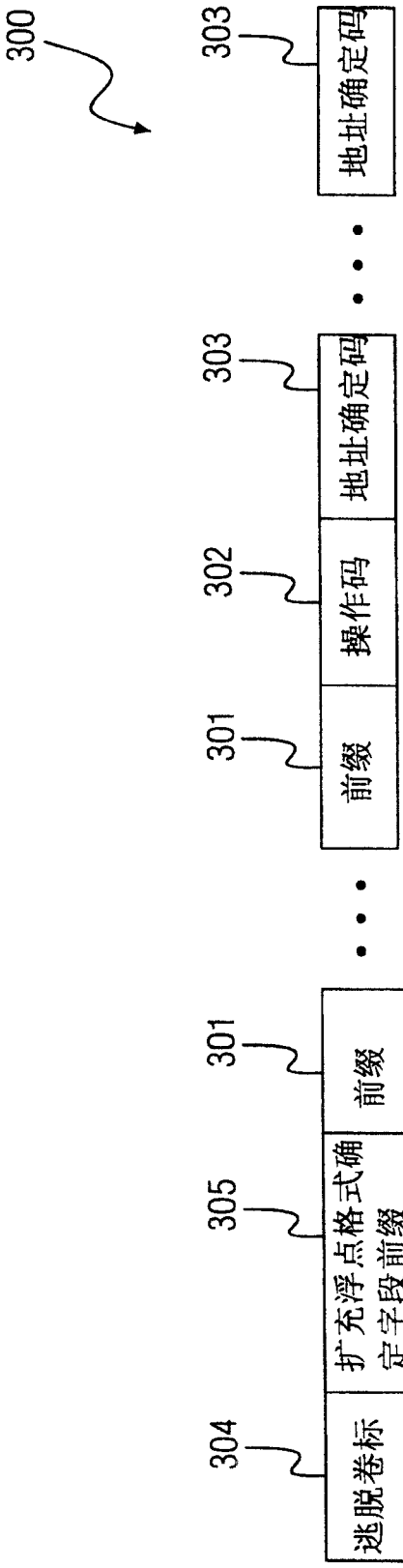

300
303
地址确定码
303
地址确定码
302
操作码
301
前缀
301
前缀
305
扩充浮点格式确定字段前缀
304
逃脱卷标

图3

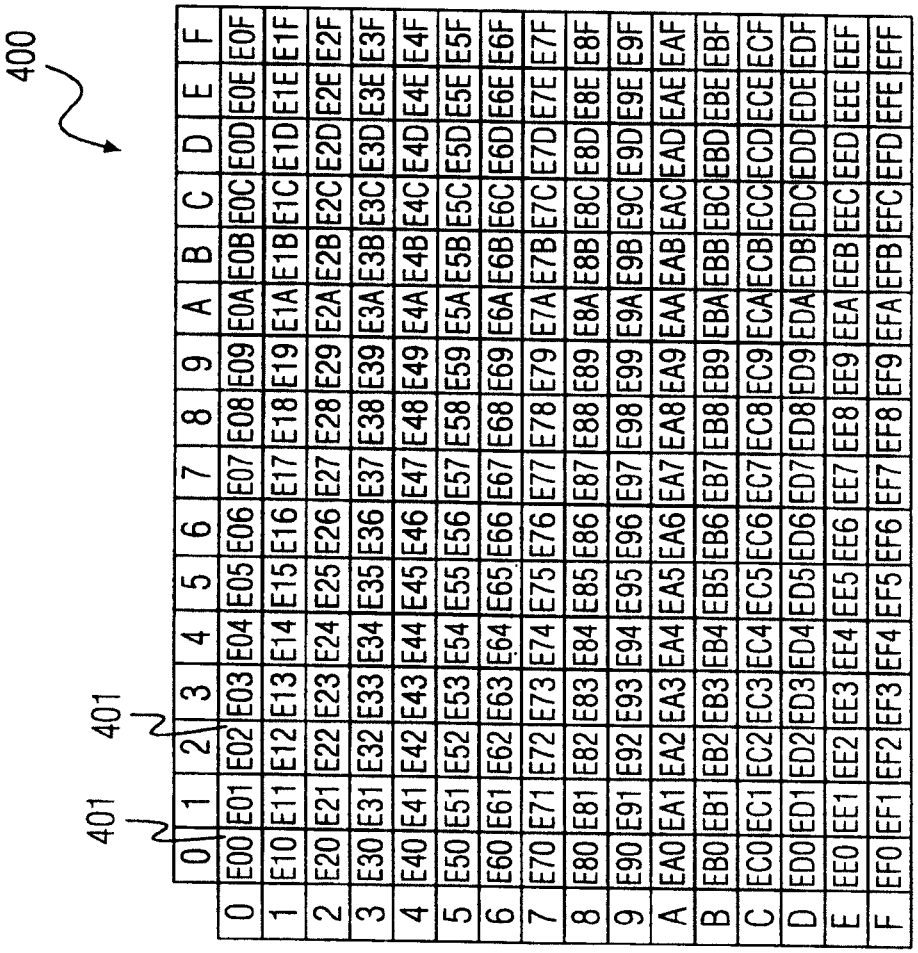

400
401
401
0 1 2 3 4 5 6 7 8 9 A B C D E F
0 E00 E01 E02 E03 E04 E05 E06 E07 E08 E09 E0A E0B E0C E0D E0E E0F
1 E10 E11 E12 E13 E14 E15 E16 E17 E18 E19 E1A E1B E1C E1D E1E E1F
2 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E2A E2B E2C E2D E2E E2F
3 E30 E31 E32 E33 E34 E35 E36 E37 E38 E39 E3A E3B E3C E3D E3E E3F
4 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49 E4A E4B E4C E4D E4E E4F
5 E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E5A E5B E5C E5D E5E E5F
6 E60 E61 E62 E63 E64 E65 E66 E67 E68 E69 E6A E6B E6C E6D E6E E6F
7 E70 E71 E72 E73 E74 E75 E76 E77 E78 E79 E7A E7B E7C E7D E7E E7F
8 E80 E81 E82 E83 E84 E85 E86 E87 E88 E89 E8A E8B E8C E8D E8E E8F
9 E90 E91 E92 E93 E94 E95 E96 E97 E98 E99 E9A E9B E9C E9D E9E E9F
A EA0 EA1 EA2 EA3 EA4 EA5 EA6 EA7 EA8 EA9 EAA EAB EAC EAD EAE EAF
B EB0 EB1 EB2 EB3 EB4 EB5 EB6 EB7 EB8 EB9 EBA EBB EBC EBD EBE EBF
C EC0 EC1 EC2 EC3 EC4 EC5 EC6 EC7 EC8 EC9 ECA ECB ECC ECD ECE ECF
D ED0 ED1 ED2 ED3 ED4 ED5 ED6 ED7 ED8 ED9 EDA EDB EDC EDD EDE EDF
E EE0 EE1 EE2 EE3 EE4 EE5 EE6 EE7 EE8 EE9 EEA EEB EEC EED EEE EEF
F EF0 EF1 EF2 EF3 EF4 EF5 EF6 EF7 EF8 EF9 EFA EFB EFC EFD EFE EFF

图4

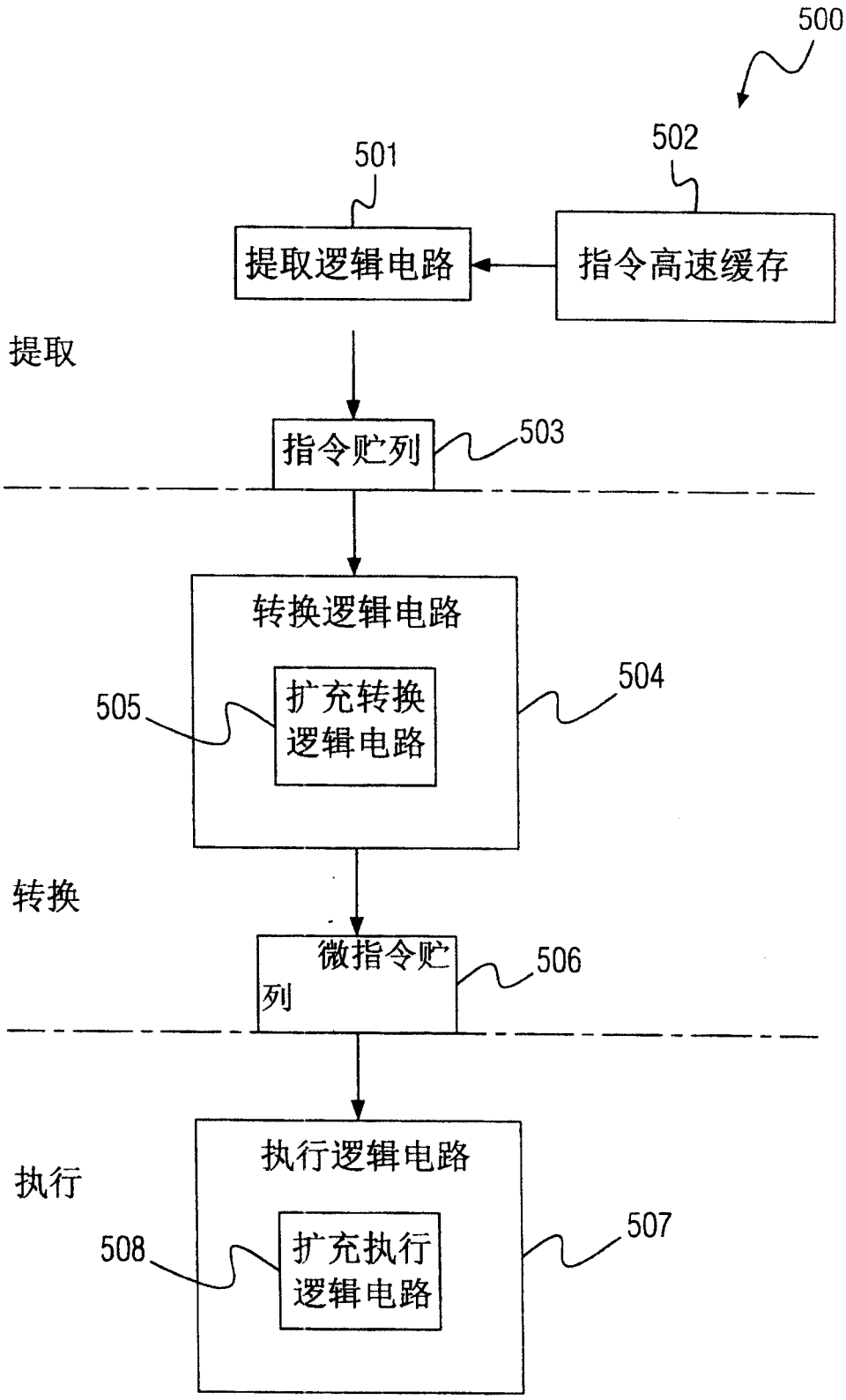

500
502
501
提取逻辑电路
指令高速缓存
提取
指令贮列
503
转换逻辑电路
扩充转换
逻辑电路
505
504
转换
微指令贮
列
506
执行逻辑电路
执行
扩充执行
逻辑电路
508
507

图5

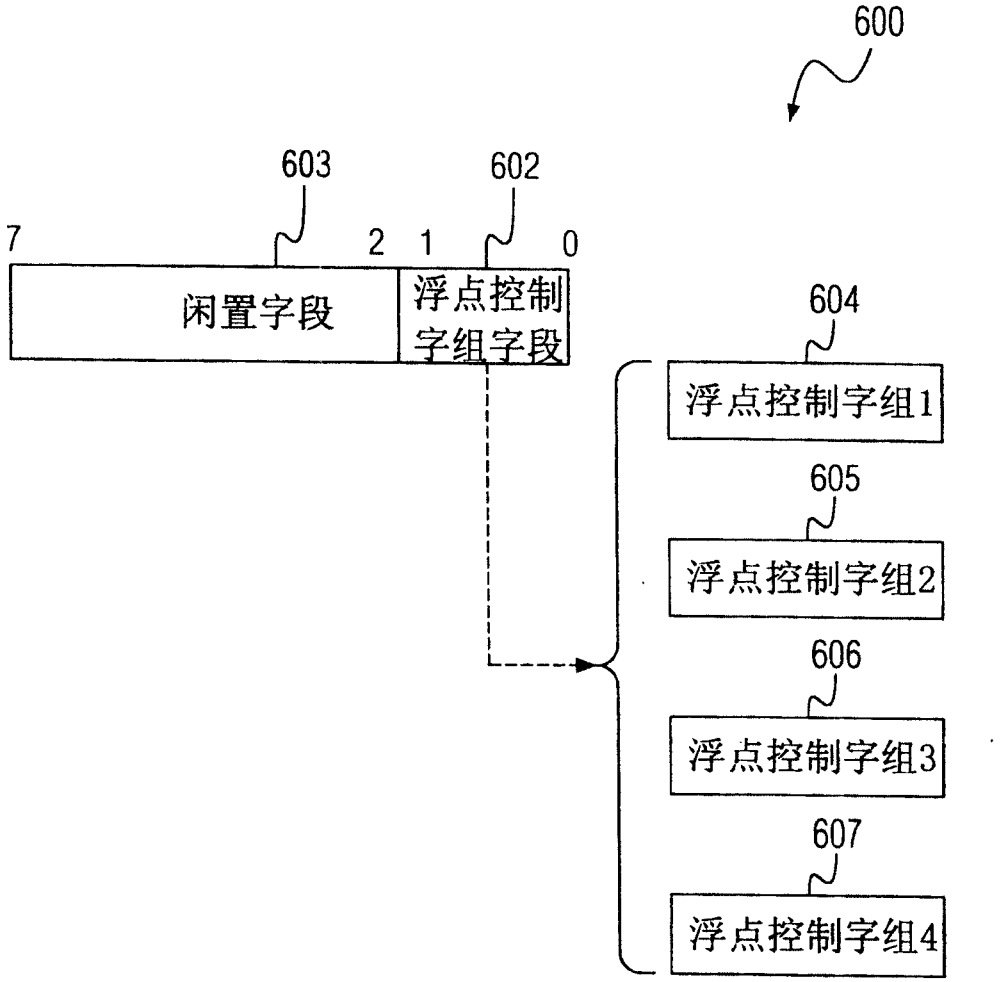

600
603
602
7
2
1
0
闲置字段
浮点控制字组字段
604
浮点控制字组1
605
浮点控制字组2
606
浮点控制字组3
607
浮点控制字组4

图6

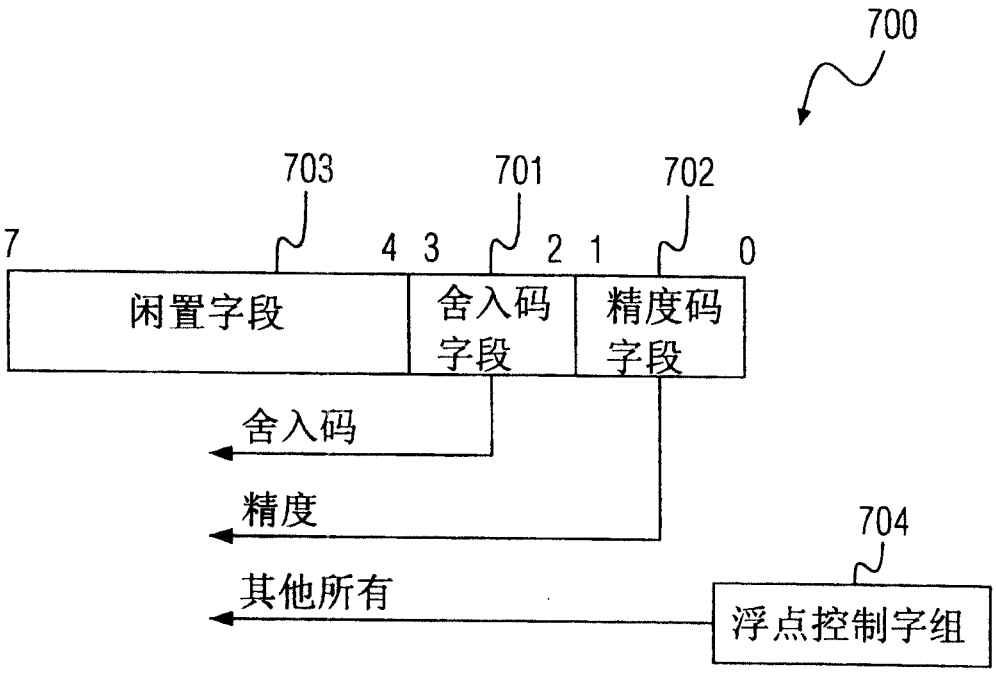

700
703
701
702
7
4 3
2 1
0
闲置字段
舍入码字段
精度码字段
舍入码
精度
704
其他所有
浮点控制字组

图7

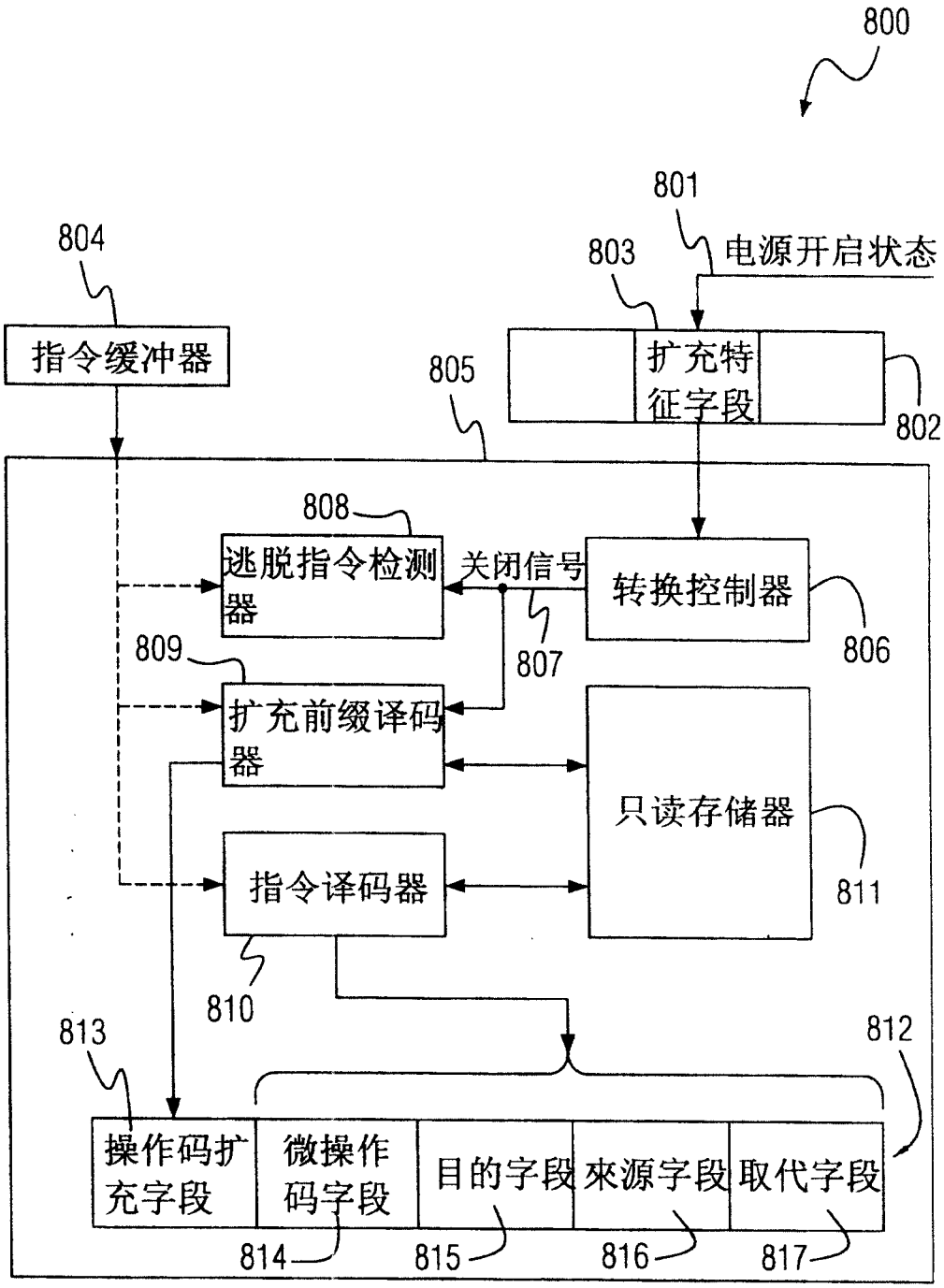

800
801
804
803
电源开启状态
指令缓冲器
805
扩充特征字段
802
808
逃脱指令检测器
关闭信号
转换控制器
806
809
807
扩充前缀译码器
只读存储器
811
指令译码器
810
813
812
操作码扩充字段
微操作码字段
目的字段
來源字段
取代字段
814
815
816
817

图8

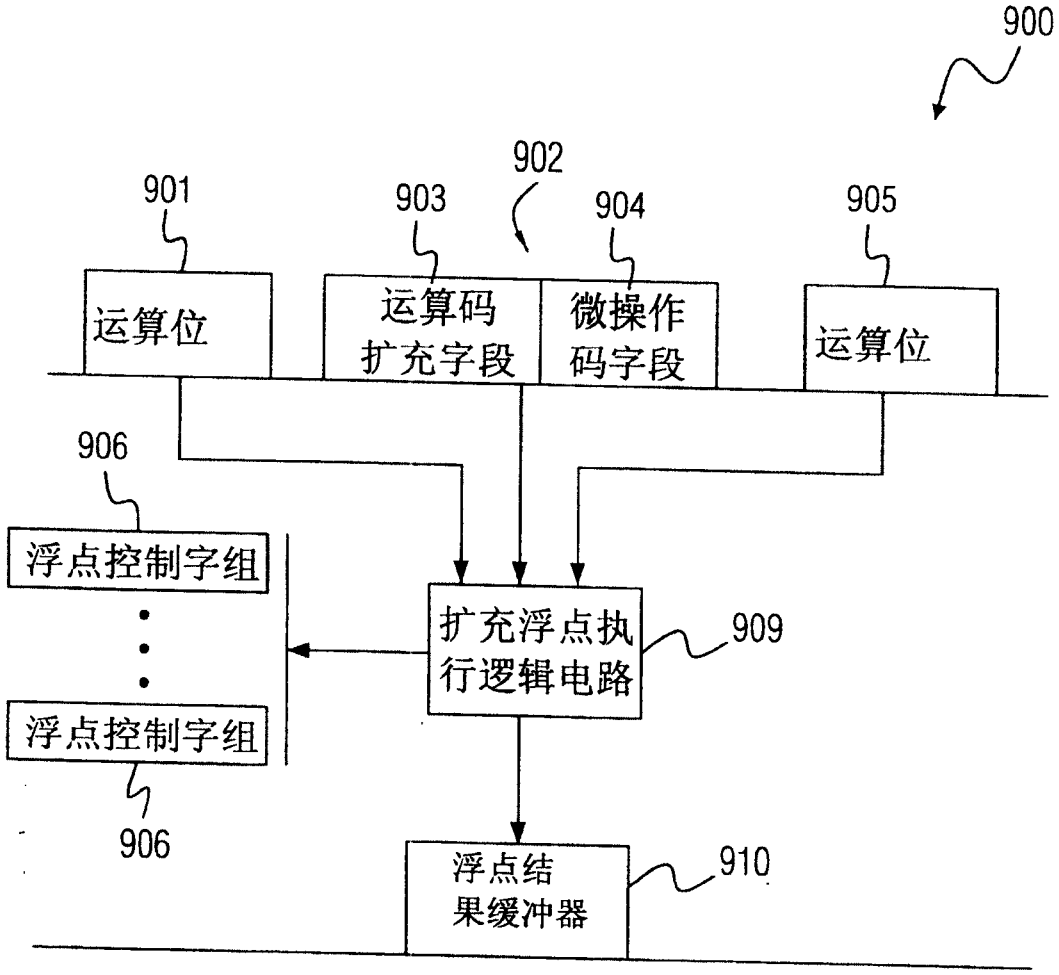

900
901
902
903
904
905
运算位
运算码
扩充字段
微操作
码字段
运算位
906
浮点控制字组
浮点控制字组
906
扩充浮点执
行逻辑电路
909
浮点结
果缓冲器
910

图9

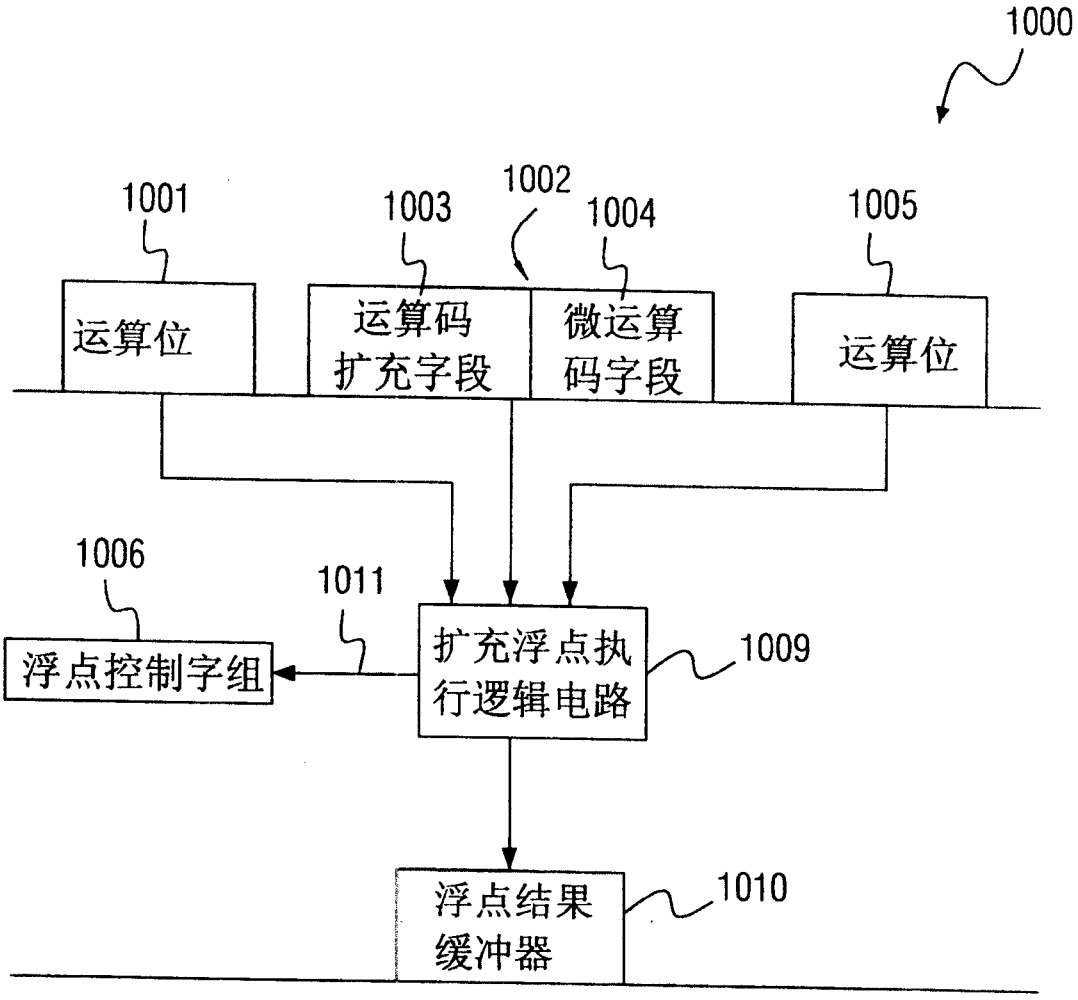

1000
1001
1003
1002
1004
1005
运算位
运算码
扩充字段
微运算
码字段
运算位
1006
1011
浮点控制字组
扩充浮点执
行逻辑电路
1009
浮点结果
缓冲器
1010

图10

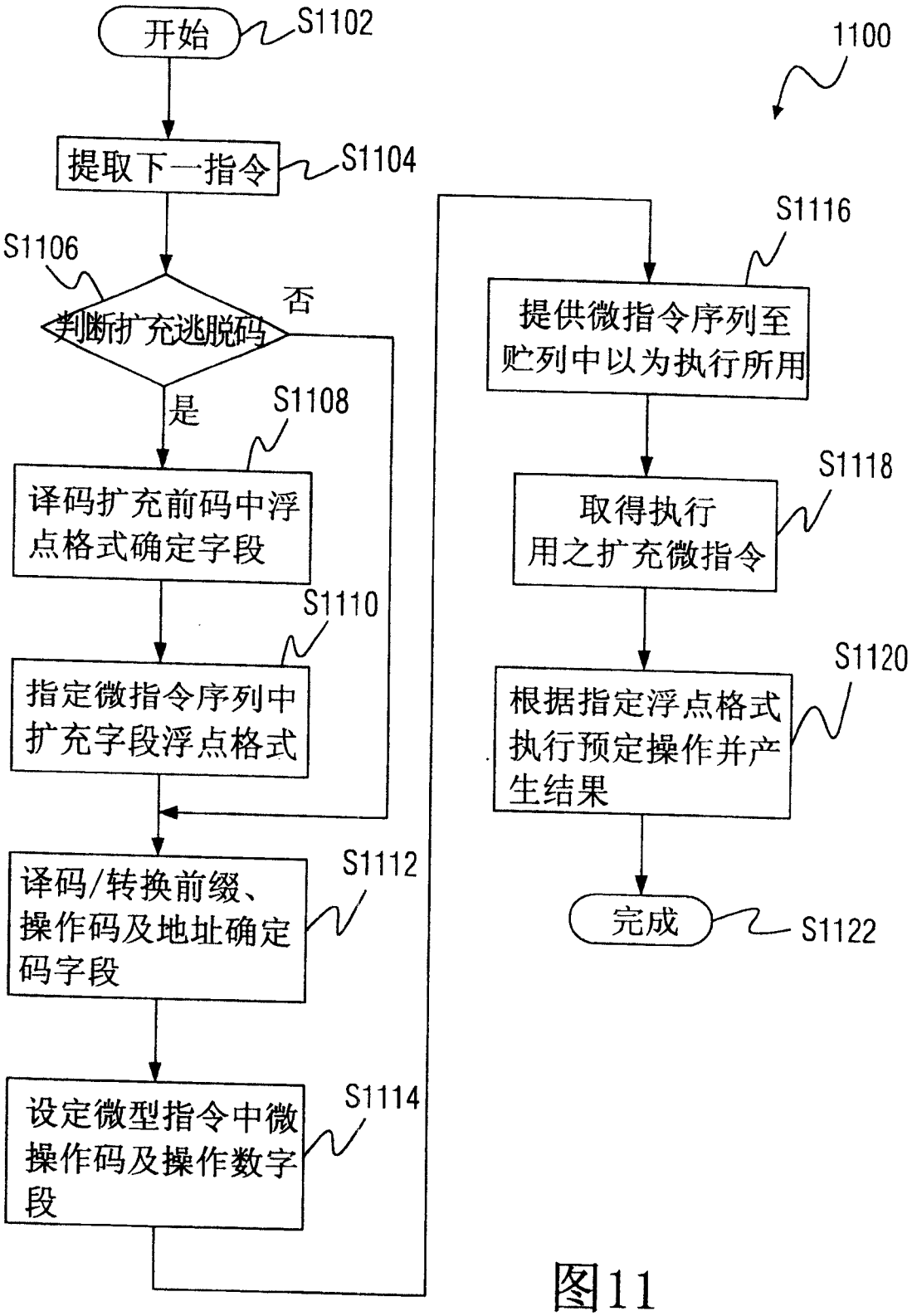

开始
S1102
1100
提取下一指令
S1104
S1106
判断扩充逃脱码
否
是
S1108
译码扩充前码中浮点格式确定字段
S1110
指定微指令序列中扩充字段浮点格式
S1112
译码/转换前缀、操作码及地址确定码字段
S1114
设定微型指令中微操作码及操作数字段
S1116
提供微指令序列至贮列中以为执行所用
S1118
取得执行用之扩充微指令
S1120
根据指定浮点格式执行预定操作并产生结果
完成
S1122

图11