

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 9/32 (2006.01)

G06F 9/38 (2006.01)



## [12] 发明专利申请公开说明书

[21] 申请号 200380104345.1

[43] 公开日 2006 年 1 月 4 日

[11] 公开号 CN 1717654A

[22] 申请日 2003.10.31

[74] 专利代理机构 中国专利代理(香港)有限公司

[21] 申请号 200380104345.1

代理人 程天正 王 勇

[30] 优先权

[32] 2002.11.28 [33] EP [31] 02079975.5

[86] 国际申请 PCT/IB2003/004962 2003.10.31

[87] 国际公布 WO2004/049154 英 2004.6.10

[85] 进入国家阶段日期 2005.5.27

[71] 申请人 皇家飞利浦电子股份有限公司

地址 荷兰艾恩德霍芬

[72] 发明人 P·P·E·穆维斯森 N·恩吉

C·H·范伯克

M·J·G·贝库伊

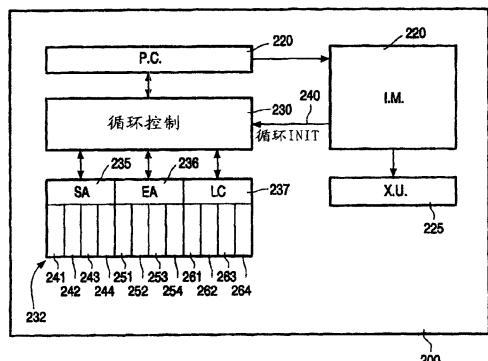
权利要求书 2 页 说明书 16 页 附图 4 页

### [54] 发明名称

数据处理器的循环控制电路

### [57] 摘要

一种数据处理器(200)包括一个操作执行单元(225)，用于执行由一程序计数器(220)指示的、来自指令存储器(210)的指令。循环控制电路(230)在一寄存器组(232)中存储用于多个指令循环的各个相关的循环信息。所述循环信息至少包括循环结束标识和用于指示循环应被执行次数的循环计数。所述循环控制电路(230)检测循环之一需要被执行并响应所述检测，为相应的循环加载循环信息，并根据加载的循环信息控制程序计数器执行相应的循环。响应于循环初始化指令(240)初始化循环信息，其中所述初始化指令在由循环信息初始化的循环开始之前被发送并且与所述循环开始无关。



1. 一种数据处理器，用于执行存储在指令存储器中的指令，所述指令由程序计数器指定，所述处理器包括：
  - 操作执行单元，用于执行由所述程序计数器指示的指令；和
  - 循环控制电路，可用于：
    - 为多个指令循环存储各个相关的循环信息；一个指令循环的循环信息至少包括循环结束标识和用于指示循环应被执行次数的循环计数；
    - 检测循环之一需要被执行并响应所述检测，为相应的循环加载循环信息，并根据加载的循环信息控制程序计数器执行相应的循环；
    - 响应于循环初始化指令初始化循环信息，其中所述初始化指令在由循环信息初始化的循环开始之前发送并且与所述循环开始无关。
- 15 2. 根据权利要求 1 所述的数据处理器，其中所述循环控制电路可用于以嵌套的形式执行多个指令循环，其中在开始执行紧邻的周围循环之前对内循环进行初始化。
3. 根据前述任何一个权利要求所述的数据处理器，其中用于操作执行单元的每个指令包括一循环开始字段，其能够指出所述指令是形成将由操作执行单元执行的指令循环的指令序列的第一指令。
- 20 4. 根据权利要求 3 所述的数据处理器，其中所述循环控制电路可响应于检测到所述循环开始字段指出一个指令循环的开始，而在与所述循环相关的循环信息中存储该循环的开始地址标识。
5. 根据权利要求 2 和 3 所述的数据处理器，其中根据连续的循环嵌套等级来存储循环信息，其中对于各个嵌套等级上的循环信息，每个时刻至多只能指定一个循环；所述循环控制电路可用于存储正被执行的指令的当前嵌套等级；和响应于下述过程来更新嵌套等级：
  - 通过检验循环开始字段检测循环的开始；和
  - 通过将程序计数器与为该循环存储的循环结束标识进行比较来检测循环的结束。
- 30 6. 根据权利要求 3 或 4 所述的数据处理器，其中所述循环开始字段能够指出需要启动多个可指定循环中的哪一个。

7. 根据权利要求 1 所述的数据处理器，其中所述循环信息包括循环开始标识。

8. 根据权利要求 7 所述的数据处理器，其中所述循环控制电路可用于通过将程序计数器与存储在循环信息中的循环开始标识进行比较来检测循环的开始。  
5

9. 根据前述任何一个权利要求所述的数据处理器，其中所述循环初始化指令包括多个用于在一项操作中初始化多个循环的循环信息的字段。

10. 根据前述任何一个权利要求所述的循环控制电路。

10 11. 一种使处理器执行由程序计数器指定的指令循环的方法，该方法包括：

-在循环开始之前并独立于该循环开始为多个指令循环存储各个相关的循环信息；一个指令循环的循环信息至少包括循环结束标识和循环计数；以及

15 -检测循环之一需要被执行并响应所述检测，为相应的循环加载循环信息，并根据加载的循环信息控制程序计数器执行相应的循环。

12. 根据权利要求 11 所述的方法，其中能够以嵌套的形式执行多个指令循环，并且所述方法包括在开始执行紧邻的周围循环之前存储一个内循环的循环信息。

20 13. 一种可使处理器执行权利要求 11 中所述的步骤的计算机程序产品。

## 数据处理器的循环控制电路

### 技术领域

5 本发明涉及一种数据处理器的循环控制电路、一种具有循环控制  
电路的数据处理器，和一种在数据处理器中执行循环的方法。

### 背景技术

处理器的性能日益增强。这使得传统上使用硬件实现的功能通过  
10 适当程序控制下的处理器就能得以执行。还能够以提高的质量对新的  
功能或现有功能进行基于软件的信号处理。新功能性的例子是第三  
代无线通信，例如基于 UMTS/FDD、TDD、IS2000 和 TD-SCDMA 标准。  
这些系统工作在非常高的频率。用于 3G 移动通信标准例如 UMTS 的调  
制解调器（收发器）需要约比 GSM 高 100 倍的数字信号处理能力。期  
望使用可编程的架构来实现这种标准的收发器以便能够处理不同的标  
准和能够灵活的适用于新的标准。使用工作在传统频率下的传统 DSP  
15 技术可能需要多至 30 个 DSP 以提供所需的性能。将清楚看到与传统的  
单一标准的收发器的基于硬件的方案相比，这样一种方案既非成本高  
效的，也非能力高效的。处理器的数字信号处理能力可通过使用流水  
线技术得到提高。  
20

US4792892 描述了一种流水线式处理器。为了执行循环控制指令，  
该指令指定对“*I*”指令序列重复执行 *N* 次，所述处理器包括一具有指  
令计数器的环路电路，所述计数器对循环序列中的指令的执行进行计  
数并且在每次完成循环时产生一序列结束信号。使用寄存器来响应于  
25 每个序列结束信号用所述循环中的第一指令的地址刷新程序计数器。  
使用一循环计数器对所述循环的完成数量进行计数并发出表示整个程  
序的循环部分的结尾的信号，并且使程序计数器对程序的剩余部分继  
续操作。流水线式计算的关键在于，必须与窄结构一致地给出和读取  
自变量和结果。所披露的流水线式处理器允许循环控制指令在所述循  
环开始之前对将要执行的循环初始化多个“*D*”指令。所述循环控制电  
路并入一个计数器以便在启动执行所述循环序列“*N*”次之前对“*D*”  
30 指令进行计数。所述已知的系统对包括一个循环的流水线式操作提供

了更多的调度自由度。

改进处理器性能的另一种方法是使用向量处理器。一个向量包括一个以上的数据元素，例如十六个16位的元素。处理器的功能单元对由一个指令触发的向量的所有个体数据元素同时起作用。传统的向量处理器体系结构对没有高度向量化的应用无效。对于在消费类电子应用中的使用，尤其是用在移动通信中，只有在获得有效的加速的情况下才能证明向量处理器的额外成本是适当的。  
5

### 发明内容

10 本发明的目的是提供一种处理器、循环控制电路和执行更好支持高性能处理的循环的方法。

为了达到本发明的目的，本发明提供一种数据处理器，用于执行存储在一指令存储器中的指令，所述指令由一程序计数器指定，所述处理器包括：操作执行单元，用于执行由所述程序计数器指示的指令；  
15 和循环控制电路，可用于为多个指令循环存储各个相关的循环信息；一个指令循环的循环信息至少包括循环结束标识和用于指示循环应执行的次数的循环计数；检测循环之一需要被执行并响应所述检测，为相应的循环加载循环信息，并根据加载的循环信息控制程序计数器执行相应的循环；响应于循环初始化指令初始化循环信息，其中所述初始化指令在由循环信息初始化的循环开始之前发送并且与所述循环开始无关。  
20

根据本发明，可初始化多个循环，其中循环初始化独立于循环的开始。对每个循环至少存储循环计数和循环结束标识（例如，以循环序列中最后指令的地址的形式或以序列中若干指令的形式，从而能够指定相对于序列开始地址的序列结束）。在US4792892的现有系统中，自从循环初始化指令开始已经执行了“D”指令之后，循环被自动开始。使用到一个以上循环时，这样一种方案即使可行也是特别困难，因为可能还不知道在多少个指令之后才需要启动第二循环。还应注意零开销循环实现过程可从允许指定多个循环的飞利浦电子公司的R.E.A.L.DSP获知。这种DSP允许通过使用循环初始化指令指定循环结束地址来预先初始化循环。循环初始化（即，开始）与循环初始化的剩余部分相结合，其中循环计数器被指定。假设循环计数器自动初  
25  
30

始化相应的循环。这意味着循环的开始总是需要一个插入到指令流中的专用循环初始化/初始指令。

在从属权利要求 2 所规定的优选实施例中，所述循环控制电路可用于以嵌套的形式执行多个指令循环，其中在开始执行紧邻的周围循环之前对内循环进行初始化。这大大降低了初始化执行循环中所涉及的系统开销。优选的，所有循环初始化都是在最外循环外面执行的。在该情况下，没有指令周期专用于在嵌套循环内进行循环初始化。本发明者已经意识到特别是数字信号处理涉及短循环的频繁执行。2 或 3 等级深度的循环嵌套经常出现。例如，为了处理图像，最外循环可涉及处理图像帧或字段，其中下一个等级的循环涉及处理帧/字段中的像素块，且第三等级可涉及处理所述块内的像素。传统的，循环初始化处于循环开始之前的相同嵌套等级上。在具有三个嵌套等级的程序中，其中每个循环执行 10 次（因此最内循环被执行 1000 次），最外循环被初始化一次，第二循环被初始化 10 次，而内部循环被初始化 100 次。在根据本发明的系统中，在开始执行第一循环之前，可在最高等级上对所有循环进行初始化。这意味着只需要三次循环初始化，而不是已知系统中的 111 次。这也使所述循环电路高度适用于向量处理器。尽管能够对循环内的指令进行向量化，但循环的初始化难于向量化。使用根据本发明的方案，典型程序中的非向量化指令的数量可被减少。

就其自身而言，可使用各种方式来确定/指示循环的开始。如从属权利要求 3 所述，操作执行单元的每个指令包括一循环开始字段，其能够指出所述指令是形成将由操作执行单元执行的指令循环的指令序列的第一指令。例如，可对规则指令（典型的能在一指令循环中出现的那些指令）添加一位来指示该指令是否为循环的开始。以这种方式，不需要提供循环的开始位置和/或时间的标识。应该意识到这是以在所述指令中使用至少一个附加位为代价的。指令尺寸的这种增加可通过使用指令压缩来减小。

如权利要求 4 所述的措施，响应于检测到所述循环开始字段指出一个指令循环的开始，所述循环控制电路可在与所述循环相关的循环信息中存储循环的开始地址标识。例如，循环控制电路可从程序计数器检索当前指令的地址并将它存储在寄存器中。每次接收到循环的结

束（如通过为循环存储的结束信息所指出的），都能够从寄存器检索开始地址。如果希望，也可以以相对于循环结束的偏移（如循环信息中所指出的）的形式例如通过指出循环中的指令数量来存储开始地址。

5 根据从属权利要求 5 所述的措施，根据连续的循环嵌套等级来存储循环信息，其中对于各个嵌套等级上的循环信息，每个时刻至多只能指定一个循环；所述循环控制电路可用于存储正被执行的指令的当前嵌套等级；和响应于通过检测循环开始字段检测循环的开始来更新嵌套等级；以及通过将程序计数器与为所述循环存储的循环结束标识 10 进行比较来检测循环的结束。仅使用一位循环开始标识符就能够启动嵌套的循环，其中在每个嵌套等级处至多只能有一个循环。那么开始字段中的标识就含蓄的指出将启动哪个循环（即下一个较深等级上的循环）。类似的，退出循环意味着控制返回至下一较高的等级（在最高的等级上，没有循环在执行，但通常随后的处理（可以是流水线式的和/或向量化式的）将发生）。假定更深的循环由更高的数表示，则 15 进入循环将导致嵌套等级（或类似的，循环数量）的增加，而退出循环导致嵌套等级的递减。

为了克服在每个嵌套等级上只能初始化一个循环的限制，权利要求 6 的措施描述了所述循环开始字段能够指出需要启动多个可指定循环中的哪一个。例如，每个循环可与唯一的序号相对应，其中开始字段可包括这样一个数字。如果循环嵌套等级的最大数是 MAX，则总共 20  $\lceil 2\log(MAX) \rceil$  个位需要添加到可应用指令中。

根据如从属权利要求 7 所述的措施，所述循环信息还包括循环开始标识。原则上，所述标识可采取任何适当的形式，例如存储页或相对于已知位置的可寻址范围内的绝对存储器地址或相对存储器地址。尤其是，如果循环开始地址或循环结束地址是以这些方式之一指定的，则另一地址可被规定为相对于特定地址的偏移。这样一个偏移表示循环中的指令的数量。

根据权利要求 8 所述的措施，所述循环控制电路可通过将程序计数器与存储在循环信息中的循环开始标识进行比较来检测循环的开 30

始。在循环初始化指令和已初始化循环的开始之间不存在时间或位置关系的情形下，将当前地址（如在程序计数器中出现的或从其导出的）与存储在循环信息中的循环开始地址进行比较。该比较可通过将程序计数器与每个存储的循环开始地址进行比较直到找出一个匹配项或所有循环开始地址都被比较了来进行。例如通过对开始地址进行分类、简化和/或加速比较处理可以优化这个处理。

根据如从属权利要求 9 中所述的措施，所述循环初始化指令包括多个用于在一项操作中初始化多个循环的循环信息的字段。特别地，如果使用宽存储器，例如用于存储 VLIW 指令的存储器，则仅使用一个指令便可初始化若干个循环。这进一步降低了循环初始化的系统开销。

为了满足本发明的目的，提供了一种在具有用于执行由程序计数器指出的指令的操作执行单元的处理器中使用的循环控制电路，其可用于为多个指令循环存储各个相关的循环信息；一个指令循环的循环信息至少包括循环结束标识和表示循环应该执行的次数的循环计数；检测循环之一需要被执行并响应所述检测，为相应的循环加载循环信息，并根据加载的循环信息控制程序计数器执行相应的循环；响应于循环初始化指令初始化循环信息，其中在由循环信息初始化的循环开始之前发送初始化指令，并且所述指令与所述循环的开始无关。

为了满足本发明的目的，提供一种使处理器执行由一程序计数器指定的指令循环的方法，该方法包括：在循环开始之前并且不依赖于循环的开始为多个指令循环存储各个相关的循环信息；一个指令循环的循环信息至少包括循环结束标识和循环计数；和检测循环之一需要被执行并响应所述检测，为相应的循环加载信息，并根据加载的循环信息控制程序计数器执行相应的循环。

#### 附图说明

本发明的这些和其它方面通过此后所述的实施例是显而易见的，并且将参照这些实施例对其进行说明。

图中：

图 1 表示使用根据本发明的循环初始化的示例程序；

图 2 表示根据本发明的处理器和电路的方框图；

图 3 表示根据本发明的处理器和电路的实施例；  
 图 4 表示适于循环控制电路使用的计数器；以及  
 图 5 表示其中使用了循环控制电路的优选处理器。

5

### 具体实施方式

根据本发明的循环控制电路尤其适用于，但不局限于，数字信号处理器（DSP）。对于数字信号处理应用，循环和嵌套式循环在一个循环和一个循环的通常不间断的处理中常常具有相对较少的指令。这种系统可受益于根据本发明的体系结构，其减少了需要执行循环初始化指令的次数。循环控制电路还特别适用于流水线式处理器，因为它允许自由调度循环初始化指令（只要在开始循环之前初始化循环即可）。同样，可将循环刚开始之前的指令用于任何目的，这样的目的例如最好用于维持流水线的高填充度。

所述循环电路也可有利地用于向量处理器。向量处理器可用于规则的“繁重/职能”处理，尤其是内循环处理。同样，它可对将被执行的代码的向量化部分提供大规模并行性。然而，充分利用该并行性并不总是可行的，因为许多算法并不会呈现充分的正确形式的数据并行性。所谓的“阿姆达尔法则（Amdahl's Law）”规定从具有 P 个处理元素的向量处理器的向量化获得的整体加速（作为能被向量化的一小部分代码（f）的函数）等于  $(1-f+f/P)^{-1}$ 。这意味着当 50% 的代码能被向量化时，实现了小于 2 的总加速（而不是 32 的理论最大加速）。这是因为剩余的 50% 代码不能被向量化，并且因此对于该部分的代码不会获得加速。即使 90% 的代码能被向量化，所述加速仍然小于因数 8。在对代码的直接可向量化部分进行向量化之后，大部分时间花费在剩余代码上。能够将剩余代码分成四种类型：

- 地址相关指令（例如，使用模量寻址在循环缓冲器中对指针递增）
- 规则标量操作（即，与向量处理器的主循环相应的标量操作）
- 循环
- 不规则标量操作

循环控制电路减少了在循环上花费的时间，并且同样有助于使向量处理更适合于消费类电子应用，尤其是移动通信，只有在能够获得

相当大的加速的情况下才能证实向量处理器的额外成本是正当的。

图 1 表示使用根据本发明的循环初始化的示例程序。该示例程序包括四个循环，表示为 N1 至 N4，它们是以三个嵌套的等级组织的。循环 N1 是最高等级。N2 是更深的等级，而 N3 和 N4 是在更深一层的等级上的两个连续循环。该程序以任意数量的指令开始，表示为 101 到 109。其后是所有四个循环的初始化过程，表示为 110 到 113。根据本发明，循环的初始化可在程序的任意点处执行，假定在相应的循环的开始地址（在图中为：开始\_地址（start\_address））之前。同样在初始化内部循环之前初始化更高等级的循环也没有严格的理由。在初始化步骤中，至少循环计数和循环的结束标识（此后称作循环结束地址）被指定。根据执行情况，也可以指定循环的开始标识，此后称作循环开始地址。这三个参数就完整地指定了每个循环，从而在程序执行期间到达所述开始地址时，可自动开始循环，而不需要任何初始化指令，即触发开始执行循环的单独指令。能够如此做的一个详细实施例将参照图 3 进行说明。从图 1 可以看出，该原理可应用于嵌套循环，并且也可以用于在一个嵌套等级出现一个以上循环的情况。如果在初始化指令中没有（明确的或含蓄的）给出循环开始地址，则启动循环的触发被并入循环的第一个指令中，下面将更详细地说明。

在图 1 给出的例子中，所有初始化都是在最外的循环 N0 外部执行的。因为在嵌套的循环内部没有专用于循环初始化的指令周期，所以循环系统开销被充分降低。也能够在外循环内部对内循环执行某些初始化，但这样就降低了本发明的优点。对于嵌套循环，如果至少一个内循环是在开始执行紧临的周围循环之前进行初始化的，就可获得一个优点。如所述的，所有循环都优选地在任何一个循环外的主执行等级上被初始化。

图 2 表示根据本发明的数据处理器 200 的基本框图。数据处理器 200 能够执行存储在指令存储器 210 中的指令。将要执行的指令由程序计数器 220 来指定。所述指令存储器可完全或部分（例如以指令高速缓冲存储器的形式）结合在处理器中。如果期望，指令存储器也可以与处理器分开。所述处理器包括一操作执行单元 225，用于执行由程序计数器指示的规范式指令。特殊指令，例如处理器配置指令可分别进行处理。这不是本发明的一部分并且不作进一步的说明。循环控制电

路 230 能够存储用于多个指令循环的各个相关循环信息。一个指令循环的循环信息至少包括循环结束标识和用于表示循环将被执行的次数的循环计数。所述循环信息还可以包括循环开始标识。实际存储器 232 (例如以一个或多个寄存器单元的形式) 可处在循环控制单元 230 中或与它相连接。图 2 示出了布置存储器 232 的一种典型方式。所述存储器被分成三个寄存器组 235、236 和 237，分别用于存储开始地址、结束地址和循环计数。在图中，每个组能够存储四个值。所示为 241、242、243 和 244 用于存储开始地址，251、252、253 和 254 用于存储结束地址，261、262、263 和 264 用于存储循环计数。同样，在本实施例中四个循环的最大值可在每个时刻进行初始化。循环控制单元也能够识别一个循环的各个值 (例如用于初始化各个值和用于执行循环的值)。各个循环中的一个循环的值可例如由循环号表示。例如，循环 no. 0 包括值 241、251 和 261；循环 no. 2 包括值 242、252 和 262，等等。所述循环控制电路能够检测到循环之一需要被执行。下面，将更详细的说明检测这一过程的若干种方式。响应于检测到需要开始一个循环，循环控制电路能够为相应的循环加载循环信息，并根据加载的循环信息控制程序计数器执行相应的循环。在这方面，所述循环控制电路与已知的循环控制电路的作用相同，并且将不对这个方面做更详细的说明。根据本发明，操作控制单元 230 能够响应循环初始化指令初始化循环信息，如 240 所示。循环控制单元确保将提供的信息存储在存储器 232 的适当存储位置供稍后使用。必须在循环信息初始化的循环的启动之前发送初始化指令，并且所述初始化指令与该循环开始无关。可在程序计数器 220 的控制下从指令存储器 210 加载循环初始化指令。一指令解码单元 (未示) 可将指令中的信息提供给循环控制单元而不是将指令提供给执行单元 230。

为了进一步说明本发明，在下表 (表 1) 的左栏中示出了用于传统的零系统开销循环处理器例如 Philips R. E. A. L DSP 的指令序列，而根据本发明的指令序列表示在右栏中：

loop 1 init loop 1 body {         instr 1-1 : loop 2 init loop 2 body {             inst 2-1 : loop 3 init loop 3 body {                 inst 3-1 : }         }     :	loop 1 init loop 2 init loop 3 init loop 1 body {         instr 1-1 : loop 2 body {             inst 2-1 : loop 3 body {                 inst 3-1 : }         }     :
--	--

表 1

如上所述，循环初始化指令至少提供了循环计数，和循环结束地址。为了循环控制电路确定循环应该开始，用于操作执行单元的每个 5 指令包括一循环开始字段，其能够表示所述指令是形成将由操作执行单元执行的指令循环的指令序列的第一条指令。实际上所有指令都可具有这样一个循环开始字段以保持所有指令一致的指令结构。然而，应该意识到这不是必需的。例如，某些指令只能用于配置处理器而不适于在循环内使用。原则上，这种指令并不需要所述字段。以简单的 10 形式，循环开始字段可以是所述指令中的一位字段。一预定值（例如，二进制 1）可用于表示所述指令是一个循环的第一指令，而另一个二进制值（例如 0）用于非所述循环第一指令的序列中的所有指令。在下表左侧示出了每条指令的一个典型开始字段值。

```

0 loop 1 init
0 loop 2 init
0 loop 3 init
loop 1 body {
    1     instr 1-1
    0     :
    loop 2 body {
        1         inst 2-1
        0         :
        loop 3 body {
            1             inst 3-1
            0             :
            }
            0         :
        }
        0     :
    }
}

```

表 2

应该意识到字段的其它编码法也是可能的，只要循环控制电路能够确定一条指令是一个循环中的第一条指令。优选地，响应于检测到 5 循环开始字段表示一指令循环的开始，循环控制电路 230 在与循环相关的循环信息 232 中存储循环开始地址标识。在其自身中可存储任何适当的标识，例如使用完全绝对地址，使用可寻址范围内的相对地址（因此相对于所述范围的开始），或者使用相对于循环结束地址的地址（例如，使用循环中的指令数的计数）。

仅使用一位开始字段能够支持多个嵌套循环，如表 2 中所示。一个限制是在循环的每个嵌套等级处只能指定一个循环。参照图 1，在相同的嵌套等级上将不可能具有两个连续的循环 N2 和 N3，因为一位指示符不能在相同等级上的两个循环之间做出鉴别。由于该限制，另外要求循环控制电路知道循环的嵌套等级。这能够以简单的方式实现，例如通过使循环数表示嵌套等级（连续更高的循环数表示更深层的循环）。循环控制电路例如在寄存器中存储将被执行的当前循环号/嵌套等级。假定循环/嵌套等级的指示连续排列顺序，循环控制电路响应于 15

检测循环的开始增加当前循环号/嵌套等级。如上所述，可通过检查接着将由处理器执行的指令的循环开始字段检测循环的开始。响应于检测循环的退出，循环控制电路缩减当前循环号/嵌套等级。循环控制电路通过将程序计数器与存储的当前循环标识的结束地址相比较而能够检测一个循环的结束。如果检测到循环的结束并且已经根据存储的循环计数执行了所述循环，则退出循环。

在根据本发明的另一个实施例中，循环开始字段能够指出需要开始多个可指定循环中的哪一个。例如，通过在每条指令中指定一循环数，循环控制电路通过确定两个连续指令之间的循环数的变化而能够确定进入或退出一个新的循环。主执行等级（不是任何循环的一部分）可例如使用等级0（零）来指示。所有其它循环可以它们在程序中出现的顺序进行编号，但这并不是必需的；原则上任何序列都是允许的。对于具有三个循环的程序，必须产生三个循环和主等级之间的区别，这需要两位。在表3的左侧中示出了每个指令的一典型2位开始字段值。左栏示出了三个嵌套等级的工作情况，而右栏示出了两个嵌套等级的情况，其在等级2上具有两个连续的循环。

00	loop 1 init	00	loop 1 init
00	loop 2 init	00	loop 2 init
00	loop 3 init	00	loop 3 init
	loop 1 body{		loop 1 body{
01	instr 1-1	01	instr 1-1
01	:	01	:
10	loop 2 body{	10	loop 2 body{
10	inst 2-1	10	inst 2-1
10	:	10	:
11	loop 3 body{	10	:
11	inst 3-1		}
11	:	11	loop 3 body{
	}	11	inst 3-1
10	:	11	:

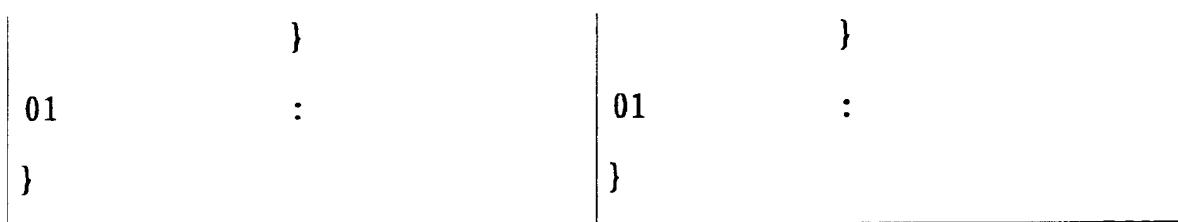


表 3

图 3 表示根据参照图 1 解释的原理在程序控制器内部的零系统开销循环 (0 OHL) 的一优选实施例的方框图。0 OHL 单元的唯一初级输入为循环指令 300。这个指令由完整指令流的循环相关部分构成，并且  
 5 当没有出现循环指令时，信号循环\_指令 (loop\_instruction) 等于无操作 (NOP)。当发送循环初始化指令时，输入信号循环\_指令指定循环计数、开始地址和结束地址。优选的零开销循环硬件包括两个地址寄存器单元 (在图中，表示为开始地址单元 (START ADDRESS UNIT)  
 10 310 和结束地址单元 (END ADDRESS UNIT ) 320)、一个循环计数器单元 330、一个循环控制单元 340 和三个比较器单元 350、360 和 370。  
 所述硬件支持 M 个循环，也就是当每个嵌套等级只包含一个循环时，  
 最大嵌套等级是 M。因此，开始和结束地址单元 310、320 具有 M 个寄存器，用于存储每个循环的循环开始和结束地址。此外，在循环计数器单元 330 中包括 M 个循环计数器。当发生循环初始化时，循环参数  
 15 (开始地址、结束地址和循环计数) 被写入相匹配的寄存器中。循环  
 指令包括正被初始化的循环的标识，优选地以直接可转换成寄存器\_选择 (register-select) 信号 (和用于循环计数器单元的计数器\_选择 (counter-select) 信号) 的形式。循环控制单元 340 通过寄存器\_选择信号和计数器\_选择信号使用该信息选择相匹配的寄存器。通过各个  
 20 输入信号提供各个寄存器值和计数器值。使用各个写\_允许 (writer\_enable) 信号和设置\_计数器 (set\_counter) 信号来控制将  
 寄存器/计数器值写入指示的寄存器/计数器字段。

当前循环被定义为进入程序后最近的循环。循环控制单元 340 使用当前循环指针 342 来产生寄存器\_选择信号，其选择用于当前循环的  
 25 循环参数。在开始和结束地址单元处的各个比较器 310 和 320 负责将程序计数器 380 的值与已经存储在这些单元中的值进行比较。所述比较器可同时将其寄存器单元的所有 M 个寄存器值与程序计数器的当前值进行比较。如果它检测到一个相匹配的值，则比较器就指出相等。

当一个以上的开始地址值与程序计数器相匹配时，通过将对应于最小结束地址的循环作为当前循环来确定当前循环。当一个以上的结束地址值与程序计数器相匹配时，按照从当前循环开始的一个顺序来处理各个循环。在一优选实施例中，循环控制单元 340 还执行对开始地址的排序并产生一个用于选择目前程序计数器值之后所期望出现的下一个开始地址（在图中表示为开始地址单元的输出“下一个（next）”）的信号（在图中表示为下一个\_选择（next-select））。相应的，当两个或多个循环开始于相同的地址时，信号下一个\_选择自动选择具有最小结束地址的循环。以这种方式，不需要额外的系统开销也可以处理开始于相同地址的多个循环。

在程序的任一点处（同样当程序计数器对应于最外侧循环之外的一个地址时），选择一个开始地址（在图中表示为下一个）并将其与程序计数器值相比较。另外，当程序计数器在至少一个循环内时，所述程序计数器与对应于当前循环的配置的一个结束地址（在图中表示为 END ADDRESS UNIT（结束地址单元）的输出）相比较。当在开始地址比较器 310 处检测到相等时，循环控制单元 340 更新当前循环指针 342，当前循环由新的开始地址来指定，结束地址存在于相应的结束地址寄存器中，而迭代计数存在于相应的计数器的影像寄存器中。

当在结束地址比较器 320 处检测到相等时，循环控制单元 330 启动相应的循环计数器（在图中表示为计数\_允许（count-enable））。已经借助信号计数\_选择而被选择的循环计数器随后被递减并与 0 进行比较。如果计数器值为 0，则循环控制单元更新当前循环指针（程序运行至当前循环之外），程序计数器被递增，并且如上所述的以当前循环的新值继续执行程序。在这一点上，如果与刚刚退出的循环相应的最外循环具有更多的迭代要执行，则必须将循环计数器值重新初始化为初始值，以便在外循环的下一次迭代期间能够再次启动所述循环。由于此原因，在循环控制单元中必须包括一个用于确定是否为这种情况的检测。如果所述检测是肯定的（即，相应的最外侧循环仍然是有效的），则所述循环控制单元产生一个复位\_计数器（reset-counter）信号，其将所述循环的循环原始迭代数量从影像寄存器（重新）拷贝至循环寄存器中。影像寄存器的这样一种应用可从 US6064712 获知。图 4 表示具有影像寄存器 400 的循环计数器电路。存储在计数器 410

中的值可通过块 420 递减。可控制多路复用器将递减值、存储在影像寄存器中的值或输入值 440 加载到计数器 410 中。使用信号设置\_计数器、复位\_计数器和计数\_允许（图 2 中所示）来产生信号选择 450，并且该信号用于控制所述多路复用器。当接收到循环配置指令时（设置\_计数器），可通过输入值 440 加载为新的循环配置所指定的迭代数量。另两个选项是从影像寄存器更新循环（复位\_计数器）并递减循环计数器（计数\_允许），如图 2 所示。如果通过结束地址检测到相等但递减的计数值不是零，则相应的循环的开始地址（通过开始地址单元 310 的寄存器\_选择输入进行选择）被拷贝至程序计数器 380 中，从而使循环重复。

所述循环控制电路优选地在优化用于信号处理的处理器中使用。这样一个处理器可以是 DSP 或任何其它适当的处理器/微控制器。本说明的剩余部分将说明在高度有效标量/向量处理器中使用所述电路。标量/向量处理器主要用于规则、“繁重/职能”处理，尤其是用于内循环处理。所有信号处理的相当大的部分将是由标量/向量处理器的向量部分执行的。规则标量操作的运作通过在一个处理器中集成标量和向量处理而得以最佳化。可使用单独的微控制器或 DSP 130 来执行不规则的任务，并且优选地，还用于控制标量/向量处理器。

图 5 表示其中可以使用根据本发明的循环控制电路的处理器的主要结构。所述处理器包括一流水线式向量处理部分 510。为了支持向量部分的操作，所述标量/向量处理器包括一标量处理部分 520，其被布置的与向量部分并行操作。优选地，所述标量处理部分也是流水线式的。为了支持向量部分的操作，所述向量部分的至少一个功能单元也提供标量部分的相应部分的功能。例如，移位功能单元的向量部分在功能上可以使向量移位，其中标量分量由移位功能单元的标量部分提供（或发送给它）。这样，所述移位功能单元包含向量和标量部分。因此，至少一些功能单元不但具有向量部分，而且具有标量部分，其中向量部分和标量部分可通过交换标量数据协同操作。一个功能单元的向量部分提供原始（raw）处理能力，其中相应的标量部分（即，同一功能单元的标量部分）通过提供和/或使用标量数据而支持向量部分的操作。用于向量部分的向量数据通过向量流水线来提供。

在图 5 的优选实施例中，标量/向量处理器包括下述七个专用功能

单元：

-指令分配单元 (idu 550)。所述 idu 包括程序存储器 552，读取连续的 vliw 指令并对 7 个功能单元分配每个指令的 7 个段。优选地，它包含支持根据本发明的零系统开销循环的循环单元。

5 -向量存储器单元 (vmu 560)。所述 vmu 包含向量存储器（在图 5 中未示出）。

-代码产生单元 (cgu 562)。所述 cgu 专用于有限字段计算，例如用于产生 cdma 代码芯片的向量以及相关功能，例如通道编码和 CRC。

10 -ALU-MAC 单元 (amu 564)。该 amu 专用于规则整数和定点计算。

-ShuFfle 单元 (sfu 566)。该 sfu 可根据专门的混洗 (shuffle) 模式重新布置向量元素。

-左移单元 (slu 568)。该 slu 可将向量元素向左移位一个单位，例如一个字、两个字或四个字。所产生的标量被提供给其标量部分。

15 -右移单元 (sru570)。该 Sru 类似于 slu，但是向右移位。另外其能够组合 amu 的向量间操作获得的连续结果。

如上所述，可使用许多不同的方式来指出循环的开始和结束。在一优选实施例中，可使用各自的 16 位地址来指定开始地址和结束地址。也可使用 16 位来指定循环计数器。因此，需要 48 位来指定循环 20 初始化指令的参数。假定最大能够指定三个循环，则需要另外两个位来指示循环，共计给出 50 位。另外，需要在可能的指令中识别循环初始化指令的位。如果指令宽度允许，则循环初始化指令有利地包括用于初始化一个操作中的多个循环的循环信息的多个字段。特别的如果在 VLIW (超长指令字) 处理器中使用所述循环控制电路，类似例如图 5 中所示，则可在在一个指令中配置一个以上的循环。对于图 5 的 VLIW 处理器，优选的使用 128 位宽的指令。这样构成指令，使用一位来区分规则 VLIW 指令（将由执行单元执行）和 IDU 指令。一个 IDU 可使用两位来区分四个 IDU 指令（为调用、返回、循环或程序结束）。如上所述，使用具有 16 位地址宽度、11 位的循环计数器、2 位用于识别一个循环的指令存储器，能够在在一个指令中配置两个循环。这时指令字段可如表 4 中所示。第二列表示字段宽度。

<IDU 指令, VLIW 指令>	1 位
<IDU 命令>	2 位
<循环 1>	2 位
<循环计数 1>	16 位
<开始_地址 1>	16 位
<结束_地址 1>	16 位
<循环 2>	2 位
<循环计数 2>	16 位
<开始_地址 2>	16 位
<结束_地址 2>	16 位

应该意识到所示出的用于初始化循环的各种方法都可与用于紧凑编码（例如通过压缩指令）的技术结合使用。为了阐明本发明的原理，而示出这种紧凑法。

应该注意上述实施例仅仅是说明，而非限制本发明，在不脱离后  
5 附权利要求的范围的情况下，本领域技术人员将能够设计出许多可选择实施例。在权利要求中，置于括号中间的任何参考标记都不应视为对权利要求的限制。单词“包括”并不排除权利要求中所列举的那些之外的其它元件或步骤。

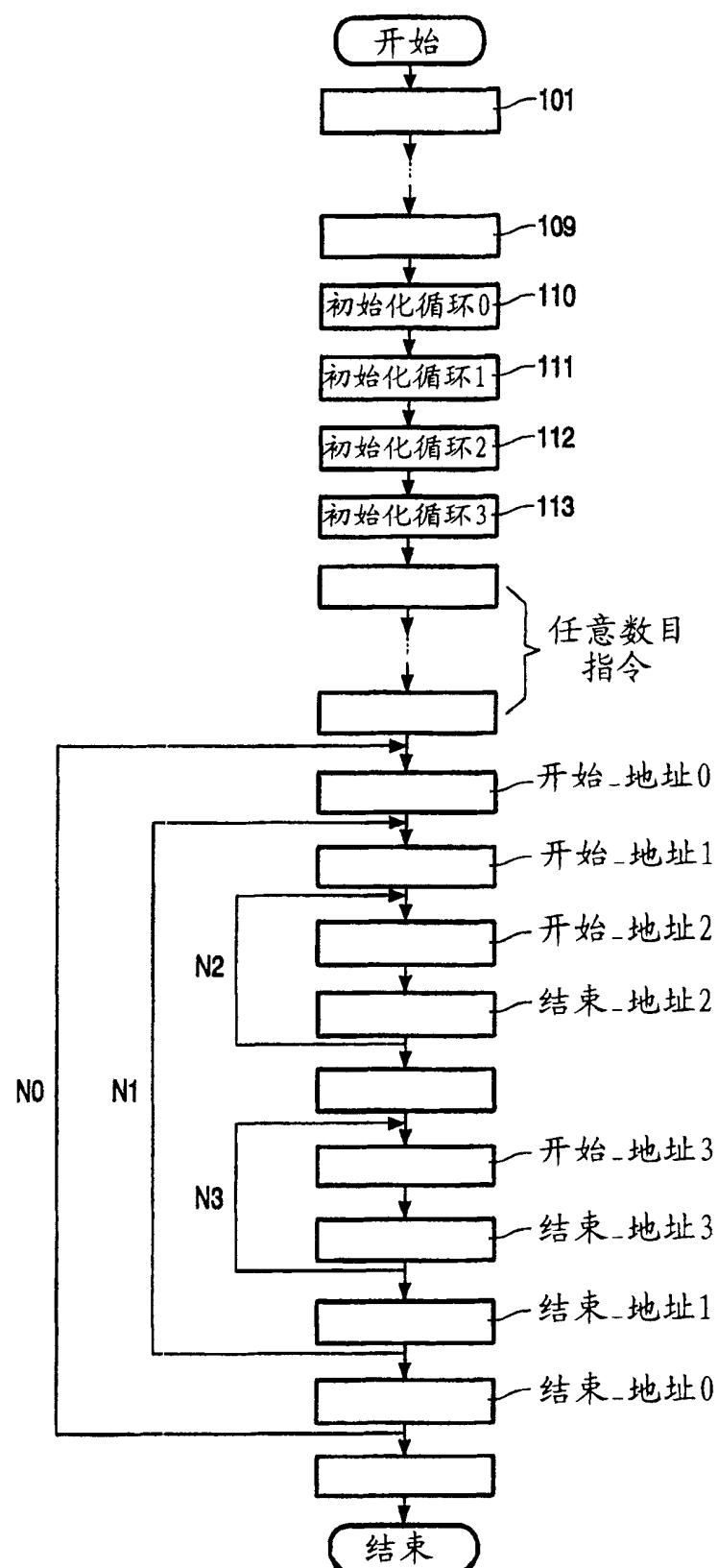


图 1

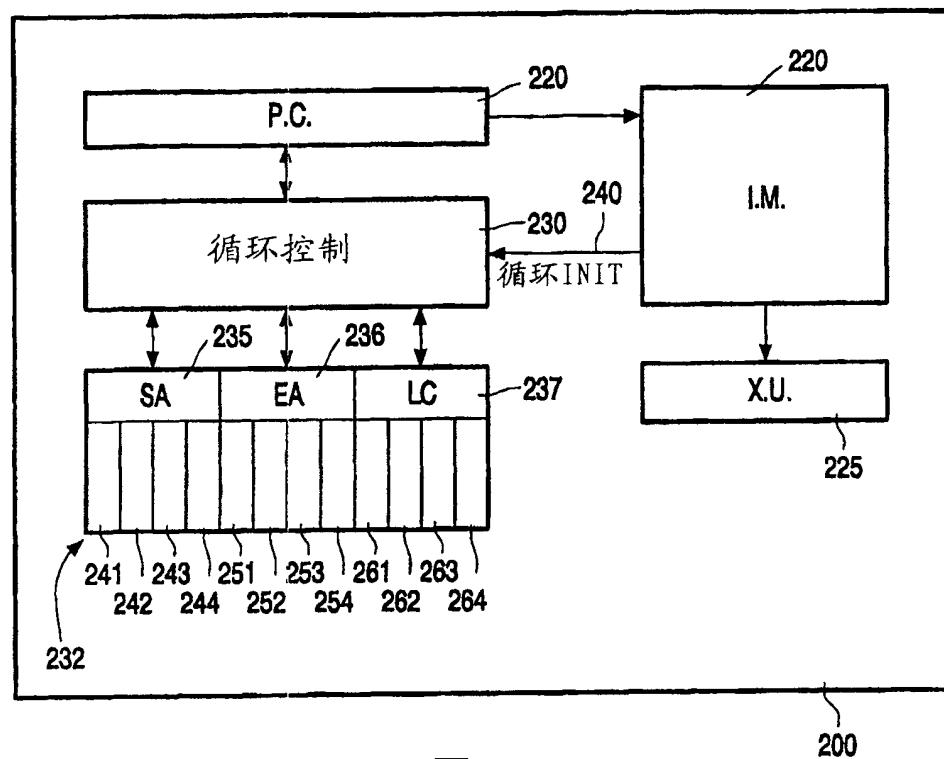


图 2

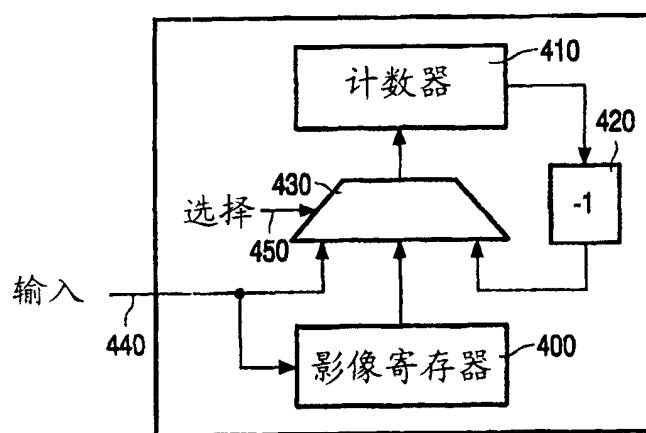


图 3

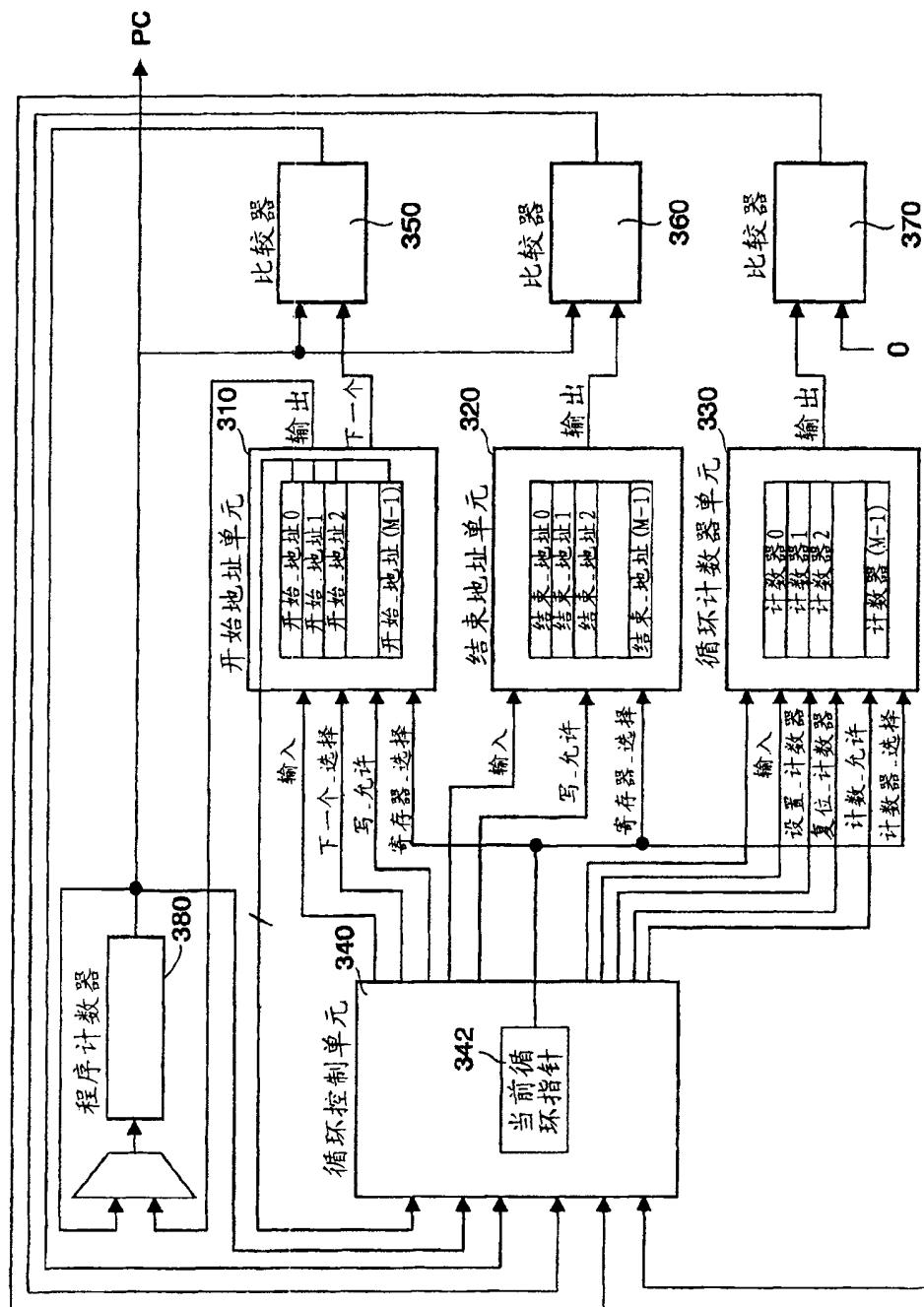


图 4

— 循环-指令 300

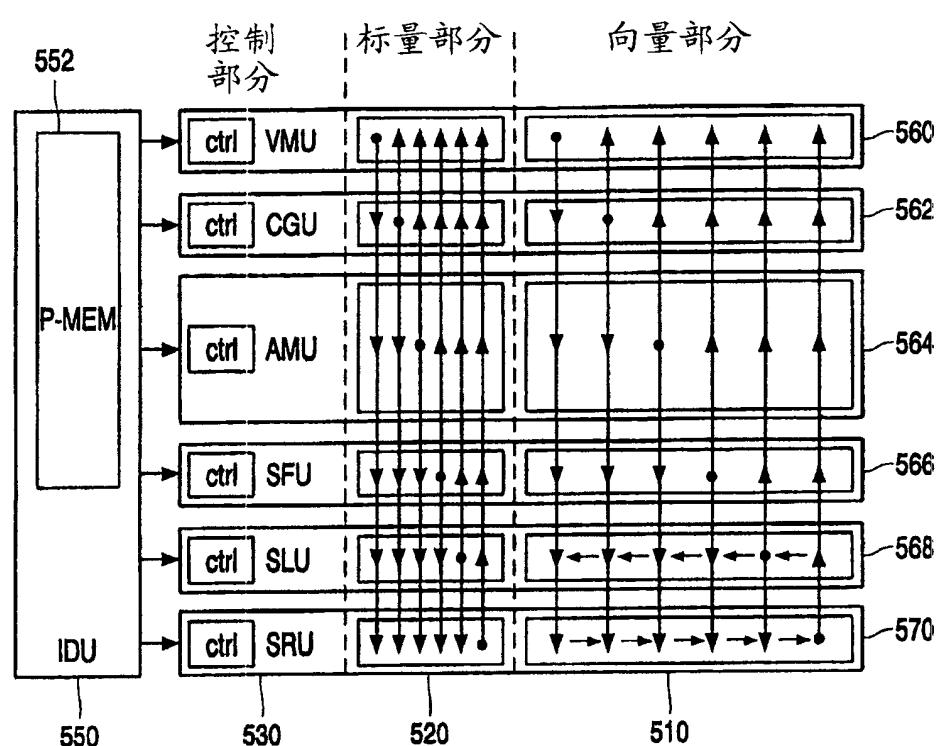


图 5