[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 15/163

[12] 发明专利申请公开说明书

[21] 申请号 02121784.X

[43] 公开日 2003 年 9 月 24 日

[11] 公开号 CN 1444154A

[22] 申请日 2002.5.31 [21] 申请号 02121784.X

[30] 优先权

[32] 2002. 3. 7 [33] JP [31] 061576/2002

[71] 申请人 株式会社东芝

地址 日本东京都

[72] 发明人 国松敦 藤原崇 雨宫治郎 白川健治

[74] 专利代理机构 中国国际贸易促进委员会专利商标事务所

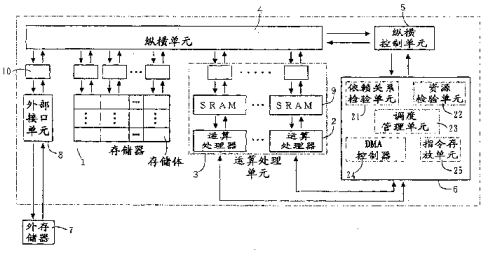
代理人 杜日新

权利要求书 2 页 说明书 11 页 附图 10 页

[54] 发明名称 多处理机系统

[57] 摘要

本发明的构成包括可以以块单位访问的存储器，可利用存放于存储器中的块数据，执行指定的任务的一个以上的运算处理器，以及控制在运算处理器中的任务的执行的控制处理器；控制处理器的构成包括检验执行指定的任务时所使用的块数据之间的依赖关系的依赖关系检验单元，根据检出的依赖关系对访问存储器，从存储器中向运算处理器 2 传送数据，以及运算处理器的数据处理进行调度的调度单元。



ISSN1008-4274

1.一种多处理机系统，其构成包括:

利用存放于存储器中的数据执行任务的一个以上的运算处理器，和

控制上述运算处理器的任务的执行的控制处理器，

上述控制处理器的构成包括:

检验执行任务之际数据之间的依赖关系的检验单元，和

根据上述检验的依赖关系，访问上述存储器，将数据从上述存储器传送到上述运算处理器以及进行上述运算处理器中的运算调度的调度单元。

2.在权利要求1中记载的多处理机系统中，

上述运算处理器以数据的块单位对上述存储器进行访问。

3.在权利要求1中记载的多处理机系统中，

上述依赖关系检验单元检测在执行同一或不同的任务之际共用的数据之间的依赖关系。

4.在权利要求1中记载的多处理机系统中，

包含控制在上述存储器和上述一个以上的运算处理器之间收发数据的数据传送控制单元，

上述调度单元考虑上述数据传送控制单元输出的传送控制信号进行上述的调度。

5.在权利要求1中记载的多处理机系统中，包括:

指令存放单元，存放有包含指定上述运算处理器进行的处理内容的标识符，表示存放上述运算处理器作为输入数据而使用的数据的上述存储器上的第1地址，以及表示存放上述运算处理器的运算结果的上述存储器上的第2地址的宏指令，

上述依赖关系检测单元根据上述第1及第2地址检出数据之间的依赖关系。

6.在权利要求1中记载的多处理机系统中，包括:

利用标识应执行的任务的标识符记录任务之间的依赖关系的条件表，和

根据上述条件表，记录上述应执行的任务的执行条件信息和各任务执行时所使用的资源信息的资源管理表，

上述依赖关系检测单元，根据上述资源管理表中记录的信息，检测在上述应执行的任务中所使用的数据的依赖关系。

7.在权利要求 1 中记载的多处理机系统中，

上述数据是图像数据，

上述依赖关系检测单元判断在生成同一或不同合成图像之际共用的数据之间的依赖关系。

8.在权利要求 1 中记载的多处理机系统中，

上述数据的大小设定为 1 千字节以上的数据大小。

9.在权利要求 8 中记载的多处理机系统中，

执行多个任务的上述运算处理器的数目的越是增加，上述数据的数据大小就可以越大。

10.在权利要求 1 中记载的多处理机系统中，

上述控制处理器，根据在上述存储器和运算处理器之间收发上述数据所需要的时间单位为基准的时钟进行处理动作。

11.在权利要求 1 中记载的多处理机系统中，

上述存储器是分为多个存储体单端口存储器。

12.在权利要求 1 中记载的多处理机系统中，包括:

上述存储器及上述运算处理器之间的数据传送用的缓存和在上述运算处理器中的数据处理用的缓存，以使上述存储器和上述运算处理器之间的数据传送和上述算处理器中的数据处理可并行进行。

多处理机系统

相关申请参考

本申请以 2002 年 3 月 7 日在日本特许厅提出的专利申请 2002-61576 为基本申请提出优先权权利要求，该基本申请的所有内容此处作为参考文献引用。

技术领域

本发明涉及可处理图像数据等的大容量数据的由多个处理机组成的多处理机系统。

背景技术

通常的处理机，因为是以处理容量比较小的数据为前提，寄存器一般是使用昂贵的小容量的多端口存储器。因此，如使用多个这种通常的处理机构筑多处理机系统，则各处理机之间必需进行频繁的数据交换，各处理机的控制变得很复杂。

现有的多处理机系统的代表性系统有共享存储型并行处理机系统和向量处理机系统。

共享存储型并行处理机系统，因为运算器用的处理机是自发地取得数据，程序很难对各处理机的处理进行最优调度。比如，在重叠描绘图形的场合，因为重复小处理单位，结果数据容量很大，因此在同一系统中，各处理机变成自发的多次取得数据进行重复地处理，事实上不可能优化各处理机的处理。

另外，在向量处理机系统中，主计算机控制向量处理机的处理。但是，在现有的系统中，主计算机不对向量处理机的网络访问及存储器访问进行调度，而是由编译器进行调度。比如，在同一系统中进行图形的重叠描绘的场合，数据的依赖关系全部都必须在编译器上检验

图。图 1 的多处理机系统的构成包括：由多个存储体构成的可以以存储体为单位进行访问的存储器 1；具有利用以存储体为单位读出的块数据进行规定的运算处理的多个运算处理器 2 的运算处理单元(LDALU)3;控制在多个运算处理器 2 和存储器 1 之间收发数据的纵横单元(x-bar)4；控制纵横单元 4 的纵横控制单元 5；控制运算处理单元 3 的控制处理器(LDPCU)6；以及和外存储器 7 进行数据收发的外部接口单元 8。

存储器 1，比如，是利用有多个存储体组成的单端口存储器构成的。运算处理单元 3 包含利用以存储体单位读出的块数据执行任务的多个运算处理器 2 和对应于各运算处理器 2 而设置的 SRAM 9。

存储器 1，运算处理单元 3 及外部接口单元 8，通过缓存 10 与纵横单元 4 进行数据收发。

控制处理器 6 的构成包括检验各任务使用的块数据之间的依赖关系的依赖关系检验单元 21；掌握运算处理器 2 及纵横单元 4 的处理状况等的资源检验单元 22；从存储器 1 向运算处理器 2 传送数据、访问存储器 1 及进行上述运算处理器 2 中的数据处理的调度的调度管理单元 23；控制在存储器 1 和运算处理器 2 之间的 DMA 传送的 DMA 控制器 24；以及存放由程序给出的指令集的指令存放单元 25。

图 2 为本实施例的处理内容的说明图。如图所示，在本实施例中，比如，假设以两幅图像合成(混合)的任务的多次反复处理作为一个线程，并行执行相互没有依赖关系的多个线程。此处，假设在生成同一或不同的合成图像之际共用的任务之间具有依赖关系，其他的任务没有依赖关系。

在图 2 中，带有符号 0~12 的各块表示图像数据，在各块的上侧记载的 addrXX 表示对应的图像数据的存放地址。比如，addr0a 表示存储器 1 的地址 0a。

图 2 的线程 0，在由标识号 P0 的运算处理器 2 将存放于存储器 1 的地址 0a 中的图像 0 和存放于地址 1a 中的图像 1 合成的图像 8 存放于地址 0c 中的同时，由标识号 P2 的运算处理器 2 将存放于存储器 1

而进行调度，需要编译处理时间。

发明内容

根据本发明的一实施例的多处理机系统的构成包括利用存放于存储器中的数据执行任务的一个以上的运算处理器和控制上述运算处理器的任务的执行的控制处理器，上述控制处理器的构成包括检验执行任务之际数据之间的依赖关系的检验单元和根据上述检验的依赖关系，访问上述存储器，将数据从上述存储器传送到上述运算处理器以及进行上述运算处理器中的运算调度的调度单元。

附图说明

图 1 为示出本发明的多处理机系统的一实施例的概略构成的框图。

图 2 为本实施例的处理内容的说明图。

图 3 为示出 Blend 指令的一例的示图。

图 4 为图 3 的 Blend 指令变换为中间指令的图。

图 5 为控制处理器的动作说明图。

图 6 为示出控制处理器的动作的流程图。

图 7 为示出控制处理器 6 的进行的调度管理的一例的图。

图 8 为示出本实施例的调度方法的一例的流程图。

图 9 为示出调度管理单元的内部构成的一例的框图。

图 10 为示出块数据的有效利用率和传送速度提高率的曲线图。

图 11 为示出图像处理专用的本发明的多处理机系统的一例的框图。

具体实施方式

下面参照附图对本发明的多处理机系统的一优选实施例予以具体说明。

图 1 为示出本发明的多处理机系统的一实施例的概略构成的框

的地址 2a 中的图像 2 和存放于地址 3a 中的图像 3 合成的图像 9 存放于地址 2c 中，之后，由标识号 P0 的运算处理器 2 对图像 8，9 合成的图像 12 进行存放于地址 0d 中的处理。

另外，图 2 的线程 1，在由标识号 P1 的运算处理器 2 将存放于存储器 1 的地址 3c 中的图像 4 和存放于地址 0b 中的图像 5 合成的图像 10 存放于地址 1b 中的同时，由运算处理器 P3 将存放于地址 1d 中的图像 6 和存放于地址 2b 中的图像 7 合成的图像 11 放于地址 3b 中，之后，由标识号 P1 的运算处理器 2 对图像 10，11 合成的图像 13 进行存放于地址 1c 中的处理。

在本实施例中，准备了 Blend 指令作为将两幅图像合成用的专用指令。Blend 指令表述为 Blend(p,x,y,z)。p 是运算处理器 2 的标识号，y 是从存储器 1 读出的第 1 输入块数据的地址，z 是从存储器 1 读出的第 2 输入块数据的地址，x 是写入存储器 1 的输出块数据的地址。就是说，Blend(p,x,y,z)表示由标识号 p 的运算处理器 2 将存储器 1 的地址 y，z 的第 1 及第 2 输入块数据合成的块数据存放于地址 x 中。

图 2 的线程 0，1，如图 3 所示，由 6 个 Blend 指令表述。图 3 的线程 0 的 Blend(P0,0c,0a,1a)对应于生成图 2 的图像 8 的处理，Blend(P2,2c,2a,3a)对应于生成图像 9 的处理，而 Blend(P0,0d,0c,2c)对应于生成图像 12 的处理。

另外，线程 1 的 Blend(P1,1b,3c,0b)对应于生成图 2 的图像 10 的处理，Blend(p3,3b,1d,2b)对应于生成图像 11 的处理，而 Blend(P1,1c,1b,3b)对应于生成图像 13 的处理。

图 3 所示的指令集，存放于图 1 所示的指令存放单元 25 中。控制处理器 6 或图中未示出的比较器及解释器将图 3 所示的指令集变换为如图 4 所示的中间指令。变换后的中间指令，可存放于指令存放单元 25，也可另外设置用于存放中间指令的存放单元。

如图所示，一个 Blend 指令变换为 3 个中间指令，并且由图中未示出的汇编器变换为机械语言而由控制处理器 6 执行。

比如，以 Blend(P0,0c,0a,1a)为例，首先，利用中间指令

DMA(POSPM,0a)将存储器 1 的地址 0a 的块数据 DMA 传送到与识别号 P0 的运算处理器 2 相对应的 SRAM 9。接着，按照中间指令 DMA(POSPM,1a) 将存储器 1 的地址 1a 的块数据经 DMA 传送到与识别号 P0 的运算处理器 2 相对应的 SRAM 9。接着，按照中间指令 kick(P0,0c,POSPM,Blend)，由识别号 P0 的运算处理器 2 将存放于 SRAM 9 中的两个块数据合成，合成的块数据存放于存储器 1 的地址 0c 中。kick 指令的最后自变量 Blend 是表示是否有 Blend 处理的指令集的地址的标志。

在图 4 的中间指令集的右侧记述的数字 0A，0B 等是识别中间指令的序号。

图 5 为控制处理器 6 的动作说明图，图 5 的右侧方向表示时间轴。图 5 说明的是图 4 所示的处理线程 0,1 的场合的控制处理器 6 的动作。

首先，控制处理器 6，顺序处理线程 0 的中间指令 0A，0B，0C。此时，控制处理器 6，指示对设置于调度管理单元 23 内的任务队列进行 DMA 传送，立刻进行下一个中间指令的处理。

这样，控制处理器 6，不是对各个中间指令一个个进行 DMA 传送，而只是对 DMA 传送指示进行蓄积于任务队列中的处理。

在线程 0 的中间指令 0C 的处理结束阶段，比如，如果从图中未示出的定时器有线程的切换中断信号输入到调度管理单元 23，则控制处理器 6 就顺序处理线程 0 顺序 1 的中间指令 1A，1B，1C 代替线程 0。并且在此处，控制处理器 6 还指示对设置于调度管理单元 23 内的任务队列进行 DMA 传送，立刻进行下一个中间指令的处理。

在线程 1 的中间指令 1C 的处理结束阶段，比如，如果从图中未示出的定时器有调度中断信号输入到调度管理单元 23，则调度管理单元 23 进行关于执行蓄积于任务队列中的中间指令的处理的任务调度，按照调度的顺序，控制处理器 6，控制 DMA 控制器 24 及运算处理器 2 执行各任务。

线程的切换中断信号及调度中断信号，比如，是从微处理器系统内的图中未示出的定时器及计数器等具有时间计测功能的电路周期地

输入。或者也可以从微处理器系统的外部电路输入这些中断信号。

在图5中示出的是，在将线程0，1的各中间指令每3个为一组执行时，输入调度中断信号，表示每当线程0或1的各中间指令每3个为一组执行时输入线程的切换中断信号的示例，但这些中断信号的输入定时也可根据实际装置形式而进行种种改变。

如果将图5的动作按照时间时序总结，就成为图6的流程图。首先，控制处理器6，选择线程顺序执行各中间指令(步骤S1)，指示对调度管理单元23的任务队列进行DMA传送(步骤S2)。

接着，控制处理器6，判断是否有线程的切换中断信号输入到调度管理单元23(步骤S3)，并且反复执行步骤S1，S2的处理一直到此中断信号输入为止。

如有线程的切换中断信号输入，控制处理器6在可执行的线程间进行仲裁，选择一个线程执行(步骤S4)。在图5中，因为只有两个线程，在线程0之后执行线程1。

其后，如有调度中断信号输入(步骤S5)，调度管理单元23就执行调度处理。如输入调度中断，首先，调度管理单元23，在读出进入任务队列的任务(步骤S6)后，在按照地址检验所读出的任务的数据的依赖关系的同时，检验资源冲突(纵横单元4及存储器1的端口号等)，以效率最高的方式对任务进行调度(步骤S7)。这种调度，因为可以作为控制处理器6的软件安装，可根据安装方式而有种种改变。

接着，控制处理器6，按照调度的顺序，控制DMA控制器24及运算处理器2以执行实际上可执行的任务(步骤S8)。

图7为示出控制处理器6的进行的调度管理的一例的示图。如图7(a)所示，假设在任务队列中蓄积有对标识号P0的运算处理器2的任务E0，E1，E0，E2和对标识号P1的运算处理器2的任务E0，E0，E2，E2。不特别关心这些任务的具体内容，下面以执行上述的Blend指令的任务为例进行说明。

在不进行任何调度管理的场合，控制处理器6，因为是从先进入任务队列的任务顺序执行，无论是标识号P0，P1的运算处理器2中

图中，示出以对应于地址的标识符对在各运算处理器 2 中的处理的开始和结束进行管理的示例。

首先，控制处理器 6，向要开始处理的运算处理器 2 发送对应于地址的标识符(步骤 S21)。接收到此标识符的运算处理器 2 执行指定的处理(步骤 S22)，在处理结束时，将该标识符返回到控制处理器 6(步骤 S23)。

控制处理器 6，将返回的标识符送到控制处理器 6 内的调度管理单元 23。调度管理单元 23 确定下一个应该向其发送标识符的运算处理器 2(步骤 S24)。这样，块数据的依赖关系的检验全部由调度管理单元 23 进行。调度管理单元 23，除了块数据的依赖关系之外，还考虑运算处理器 2 及纵横单元 4 的处理状况等资源信息，确定下一个应该向其发送标识符的运算处理器 2。

于是，控制处理器 6，向适应依赖关系的检验，并且可确保资源的运算处理器 2 发送与地址相对应的标识符(步骤 S25)。

以上的动作反复执行一直到执行任务信息单元中登录的任务全部执行完毕为止(步骤 S26)。

图 9 为示出调度管理单元 23 的内部构成的一例的框图。如图所示，调度管理单元 23 的构成包括记录对应于应该执行的任务的标识符的一览表的执行任务信息单元 31，记录任务的执行条件的执行条件信息单元 32，记录执行任务时可利用的运算处理器 2 的种类和其他的资源信息的资源管理表 33，以及表示标识符与任务的对应关系的标识符表 34。

任务，比如，是上述的 Blend 指令，对各 Blend 指令每一个都赋予一个标识符。比如，图 9 的标识符表 34 示出标识符 T1 对应于 Blend(P0,0c,0a,1a)，标识符 T2 对应于 Blend(P2,2c,2a,3a)，标识符 T3 对应于 Blend(P0,0c,0c,2c)，而标识符 T4 对应于 Blend(P1,1b,3c,0b)。

记录于执行条件信息单元 32 中的条件与记录于执行任务信息单元 31 中的标识符互相对应。比如，在图 9 中，在同时执行与标识符 T2 相对应的 Blend 指令和与标识符 T5 相对应的 Blend 指令时，也执

的任何一个首先都是执行任务 E0。但是，任务 E0 是执行同一 Blend 指令，因为在执行该指令之际是利用存放于存储器 1 中的同一数据，标识号 P0，P1 的运算处理器 2 不能同时进行处理。因此，如图 7(b)所示，标识号 P1 的运算处理器 2 必须等待标识号 P0 的运算处理器 2 结束任务 E0 的处理。所以，标识号 P1 的运算处理器 2 需要花费的时间是一直到所有的处理结束的时间。

另一方面，本实施例中的调度管理单元 23，为了使标识号 P0，P1 的运算处理器 2 可以高效率地执行任务，对蓄积于任务队列中的任务进行调度。图 7(c)示出使标识号 P1 的运算处理器 2 先执行任务 E2 的调度示例。因为任务 E0，E2 分别利用另外的数据执行 Blend 指令，不同的运算处理器 2 可同时执行各任务。

这样，在本实施例中，因为是由控制各运算处理器 2 的任务的控制处理器 6 进行调度以使多个运算处理器 2 并行执行任务，所以各运算处理器 2 可高效率地处理任务。就是说，根据本实施例，可以对各运算处理器 2 中的处理进行有效的调度。

在上述实施例中，是针对执行 Blend 指令的任务予以说明的，执行的指令不限定于 Blend 指令。作为构成任务的要素，只要是具备以下 3 种的指令就可以。

1)表示任务所需要的数据的标识符。此处，所谓的标识符是表示指示存储器 1 的块数据用的，标识符也可以是多个。

2)表示执行任务的运算器的标识符。

3)表示作为执行任务的结果的数据的标识符。

1)~3)的标识符不一定必须是用来访问存储器 1 的地址本身，也可以是与地址相对应的标识符(权标)。调度管理单元 23，将任务的顺序依赖关系表现为标识符之间的依赖关系而执行任务的调度。

下面对调度管理单元 23 的调度方法的一例予以详述。调度管理单元 23 的处理，可以以软件或硬件任何一种来实现，也可以使软件和硬件协调动作而实现。

图 8 为示出本实施例的调度方法的一例的流程图。在图 8 的流程

行与执行任务信息单元 31 的标识符 T4 相对应的 Blend 指令。另外，在执行与标识符 T2 相对应的 Blend 指令，或与标识符 T3 相对应的 Blend 指令时，执行与执行任务信息单元 31 的标识符 T1 相对应的 Blend 指令。

另外，如执行与执行任务信息单元 31 的标识符 T4 相对应的 Blend 指令结束，执行条件信息单元 32，就当作记录的标识符 T4 全部结束。在不能为标识符分配多个位字段的场合，在执行任务信息单元中有时会出现多个 T4。在此场合，已经结束的 T4 作为执行任务信息单元中从该 T4 到下一个出现的 T4 之间的时隙的任务。

执行任务信息单元 31，在执行与标识符 T4 相对应的 Blend 指令之际，参照资源管理表 33，确定执行对应的 Blend 指令的运算处理器 2。调度管理单元 23，参照资源管理表 33 的信息，确定执行 Blend 指令的运算处理器 2 的种类和 Blend 指令的执行时期。

如确定的运算处理器 2 结束处理，该运算处理器 2 释放资源，并将这一点记录于资源管理表 33 中。另外，在多个运算处理器 2 要求同一资源的场合，原则上优先处理先发出的 Blend 指令。

在本实施例中，是以块数据为单位从存储器 1 中读出，为了提高数据的传送速度，块数据的数据大小最好是设定为 1 千字节以上。这一点从一般的帧缓存的组块大小为 2 千字节来考虑可认为是合适的。但是，根据实际安装形式的不同最佳块数据的大小可以改变。

图 10 为示出表示在块数据中进行运算处理时可有效利用的数据的比例的有效利用率和从存储器 1 向运算处理器 2 发送的块数据的传送速度提高率的曲线图。有效利用率是数据大小越小越高，传送速度提高率是数据大小越大越高。

这样，块数据为 1 千字节以上的数据大小，对于块数据的传送及处理需要通常的处理器的系统时钟的数个周期。存储器 1 和运算处理器 2，因为是以块数据为单位进行处理，所以可以利用以块数据的处理时间为单位的时钟使控制处理器 6 动作。由此，可以使用比通常的处理器的系统时钟慢的时钟使控制处理器 6 动作，不需要使用昂贵的

高速部件及高速处理过程，并且硬件的定时设计也容易。

另外，对运算处理器 2 的数目没有特别限制，希望随着运算处理器 2 的数目的增加，可以使运算处理器 2 一次处理的块数据的数据大小加大。因此，由于在一个运算处理器 2 中的处理时间变长而使得控制处理器 6 切换运算处理器 2 的频繁程度相应地减小，可以减轻控制处理器 6 的处理负担。

此外，为了使多处理机系统整体的性能提高，可以考虑提高相同系统整体的动作频率的方法及增加运算处理器 2 的数目的方法，希望增加运算处理器 2 的数目而使各运算块应处理的块数据的大小加大。

(第 2 实施例)

第 2 实施例是将本发明应用于图像处理专用系统的示例。

图 11 为示出图像处理专用的本发明的多处理机系统的第 2 实施例的框图。如图所示，与纵横单元 4 相连接的有分别另外进行图像处理的多个运算处理单元(LDALU)3，控制处理器(LDPCU)6，以及存储器 1。

在运算处理单元 3 的内部，设置有多个像素管道 31，与各像素管道 31 相连接的 SRAM(SPM)9，以及进行前处理的初始化/DDA 单元 32。

各运算处理单元 3 内的像素管道 31 相当于图 1 的运算处理器 2，进行多边形的再现和模板匹配处理等图像处理。

图 11 的控制处理器 6，检验在图像处理用的任务中使用的块数据的依赖关系，根据该检验结果对运算处理单元 3 内的像素管道 31 的动作进行调度。由此，可使各像素管道 31 并行动作，可以以极高速度进行各种图像处理。

在上述实施例中，说明的是在运算处理单元 3 内设置多个运算处理器 2 的示例，但在运算处理器 2 仅仅为一个时也可应用本发明。

在上述实施例中，说明的是进行图像数据的合成的处理的示例，但本发明也可应用于图像数据合成处理以外的各种运算处理。

另外，图 1，图 5 及图 9 中示出的框图的至少一部分也可以利用软件代替硬件来实现。

图1

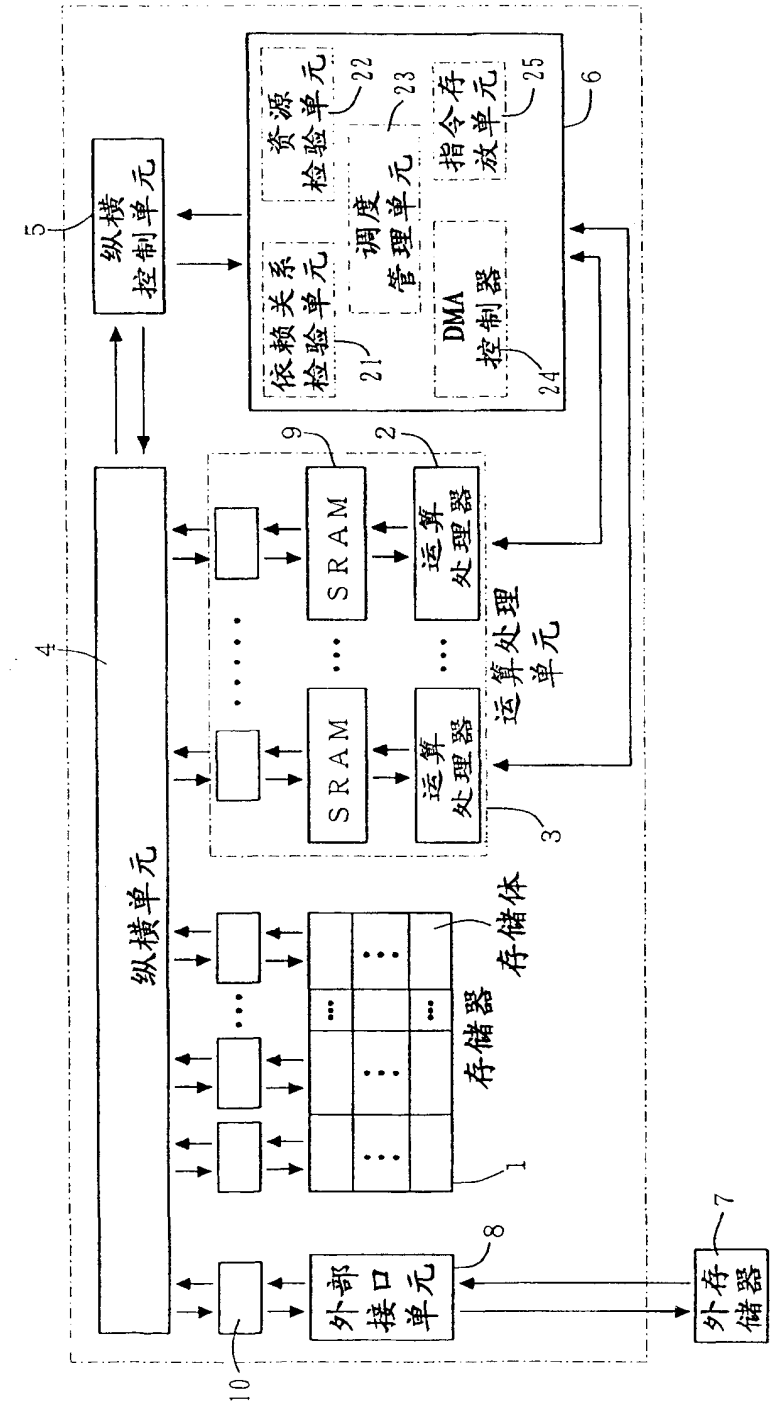

纵横控制单元
5
资源检验单元
22
调度管理单元
23
指令存放单元
25
6
依赖关系检验单元
21
DMA控制器
24
9
2
SRAM
运算处理器
运算处理单元
4
纵横单元
SRAM
运算处理器
3
存储体
存储器
1
外部接口单元
8
7
外存储器
10

图2

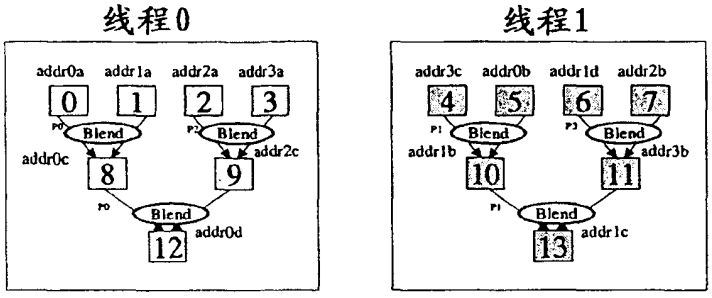

线程0
addr0a
addr1a
addr2a
addr3a
0
1
2
3
P0
Blend
P2
Blend
addr0c
8
9
addr2c
P0
Blend
12
addr0d
线程1
addr3c
addr0b
addr1d
addr2b
4
5
6
7
P1
Blend
P3
Blend
addr1b
10
11
addr3b
P1
Blend
13
addr1c

图3

线程0

Blend（P0，0c，0a，1a）
Blend（P2，2c，2a，3a）
Blend（P0，0d，0c，2c）

线程1

Blend（P1，1b，3c，0b）
Blend（P3，3b，1d，2b）
Blend（P1，1c，1b，3b）

图4

线程0

Blend(P0,0c,0a,1a)
- DMA(P0SPM , 0a) … 0 A
- DMA(P0SPM , 1a) … 0 B
- kick(P0 , 0c , P0SPM, Blend) … 0 C

Blend(P2,2c,2a,3a)
- DMA(P2SPM , 2a) … 0 D
- DMA(P2SPM , 3a) … 0 E
- kick(P2 , 2c , P2SPM, Blend) … 0 F

Blend(P0,0d,0c,2c)
- DMA(P0SPM , 0c) … 0 G
- DMA(P0SPM , 2c) … 0 H
- kick(P0 , 0d , P0SPM, Blend) … 0 I

线程1

Blend(P1,1b,3c,0b)
- DMA(P0SPM , 3c) … 1 A
- DMA(P0SPM , 0b) … 1 B
- kick(P1 , 1b , P0SPM, Blend) … 1 C

Blend(P3,3b,1d,2b)
- DMA(P2SPM , 1d) … 1 D
- DMA(P2SPM , 2b) … 1 E
- kick(P3 , 3b , P2SPM, Blend) … 1 F

Blend(P1,1c,1b,3b)
- DMA(P0SPM , 1b) … 1 G
- DMA(P0SPM , 3b) … 1 H
- kick(P1 , 1c , P0SPM, Blend) … 1 I

图5

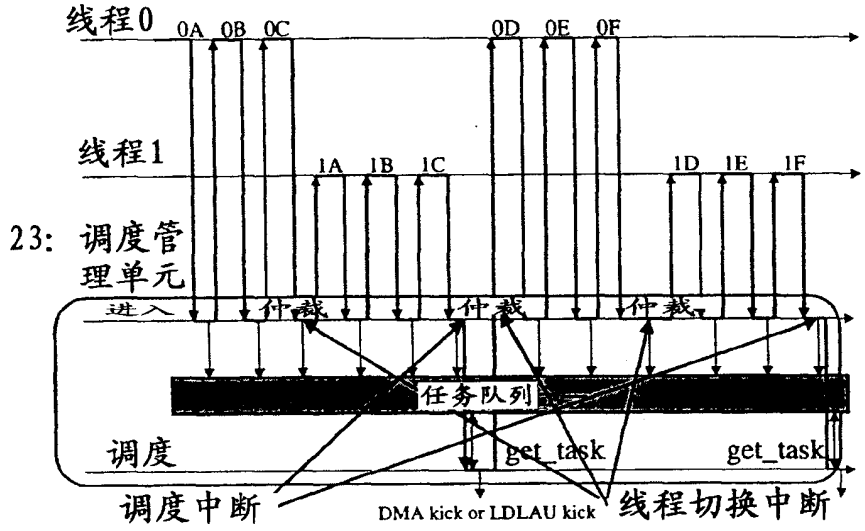

线程0
0A 0B 0C
0D 0E 0F
线程1
1A 1B 1C
1D 1E 1F
23: 调度管理单元
进入
仲裁
仲裁
仲裁
任务队列
调度
get_task
get_task
调度中断
DMA kick or LDLAU kick
线程切换中断

图6

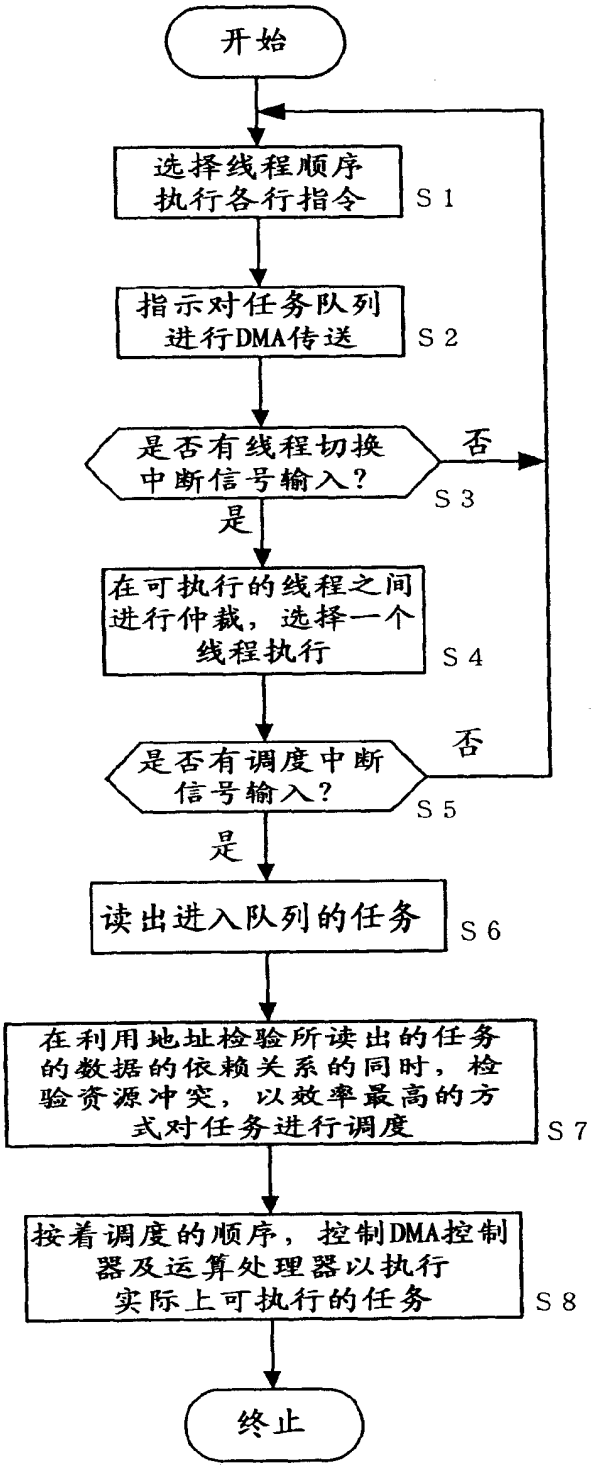
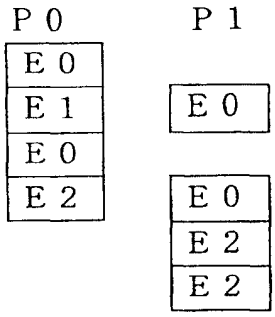

开始
选择线程顺序执行各行指令
S1
指示对任务队列进行DMA传送
S2
是否有线程切换中断信号输入?
S3
否
是
在可执行的线程之间进行仲裁，选择一个线程执行
S4
是否有调度中断信号输入?
S5
否
是
读出进入队列的任务
S6
在利用地址检验所读出的任务的数据的依赖关系的同时，检验资源冲突，以效率最高的方式对任务进行调度
S7
按着调度的顺序，控制DMA控制器及运算处理器以执行实际上可执行的任务
S8
终止

图7

(a)

线程0	线程1
E 0 → P 0	E 0 → P 1
E 1 → P 0	E 0 → P 1
E 0 → P 0	E 2 → P 1
E 2 → P 0	E 2 → P 1

(b)



(c)

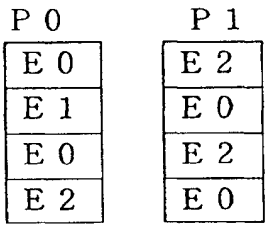


图8

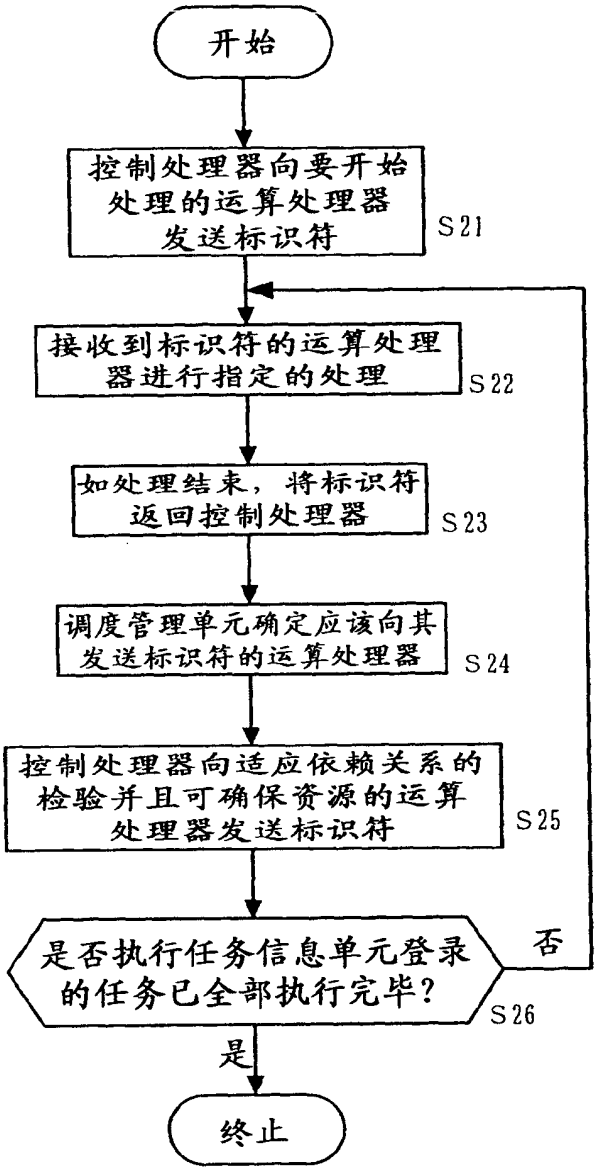

开始
控制处理器向要开始处理的运算处理器发送标识符
S21
接收到标识符的运算处理器进行指定的处理
S22
如处理结束，将标识符返回控制处理器
S23
调度管理单元确定应该向其发送标识符的运算处理器
S24
控制处理器向适应依赖关系的检验并且可确保资源的运算处理器发送标识符
S25
是否执行任务信息单元登录的任务已全部执行完毕?
S26
否
是
终止

图9

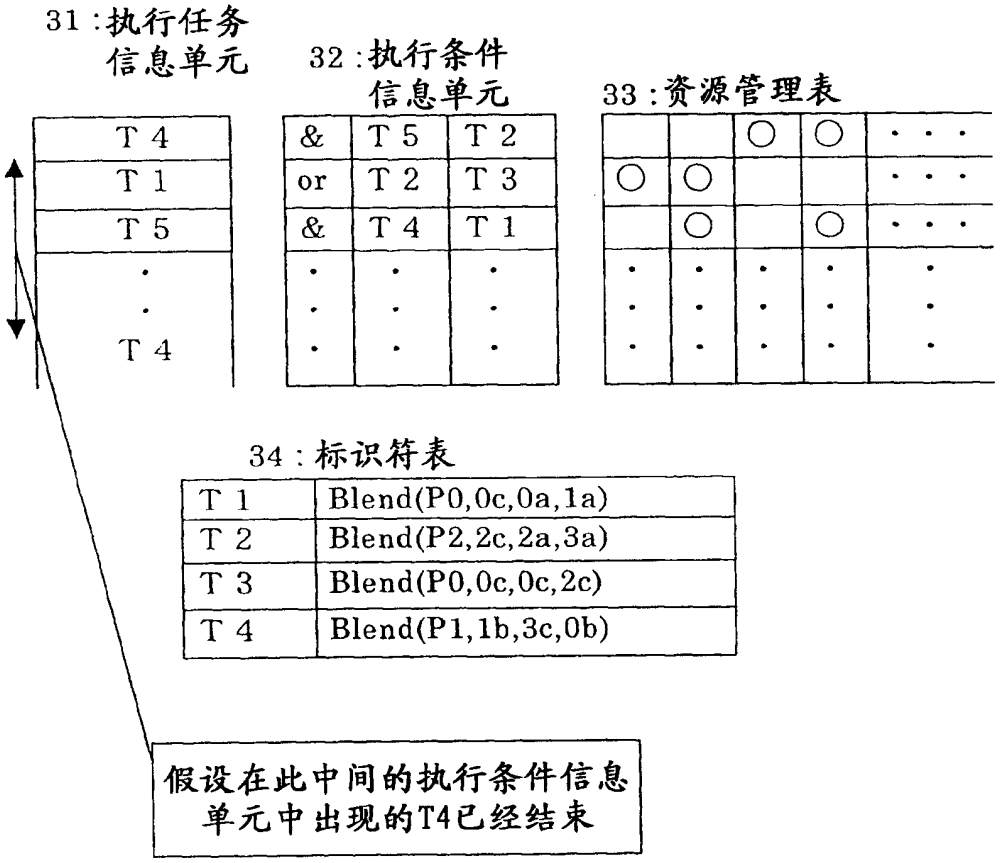

31：执行任务信息单元
32：执行条件信息单元
33：资源管理表
T 4
T 1
T 5
T 4
& T 5 T 2
or T 2 T 3
& T 4 T 1
34：标识符表
T 1 Blend(P0,0c,0a,1a)
T 2 Blend(P2,2c,2a,3a)
T 3 Blend(P0,0c,0c,2c)
T 4 Blend(P1,1b,3c,0b)
假设在此中间的执行条件信息单元中出现的T4已经结束

图10

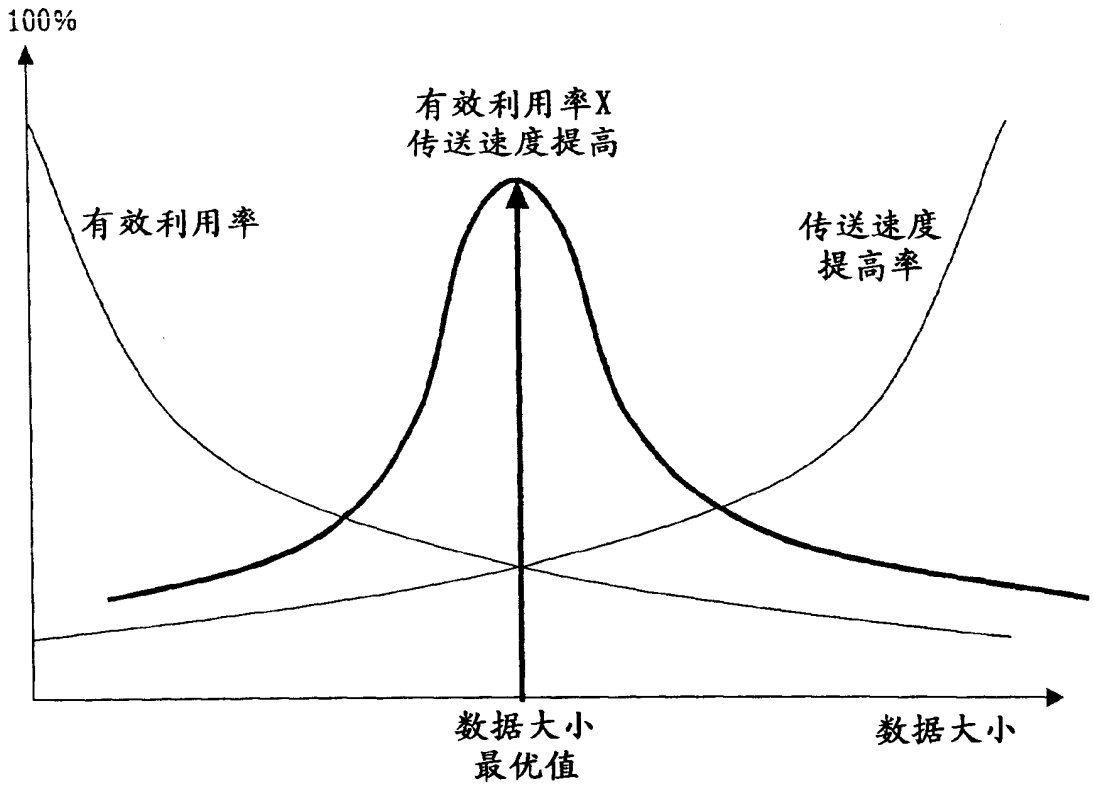
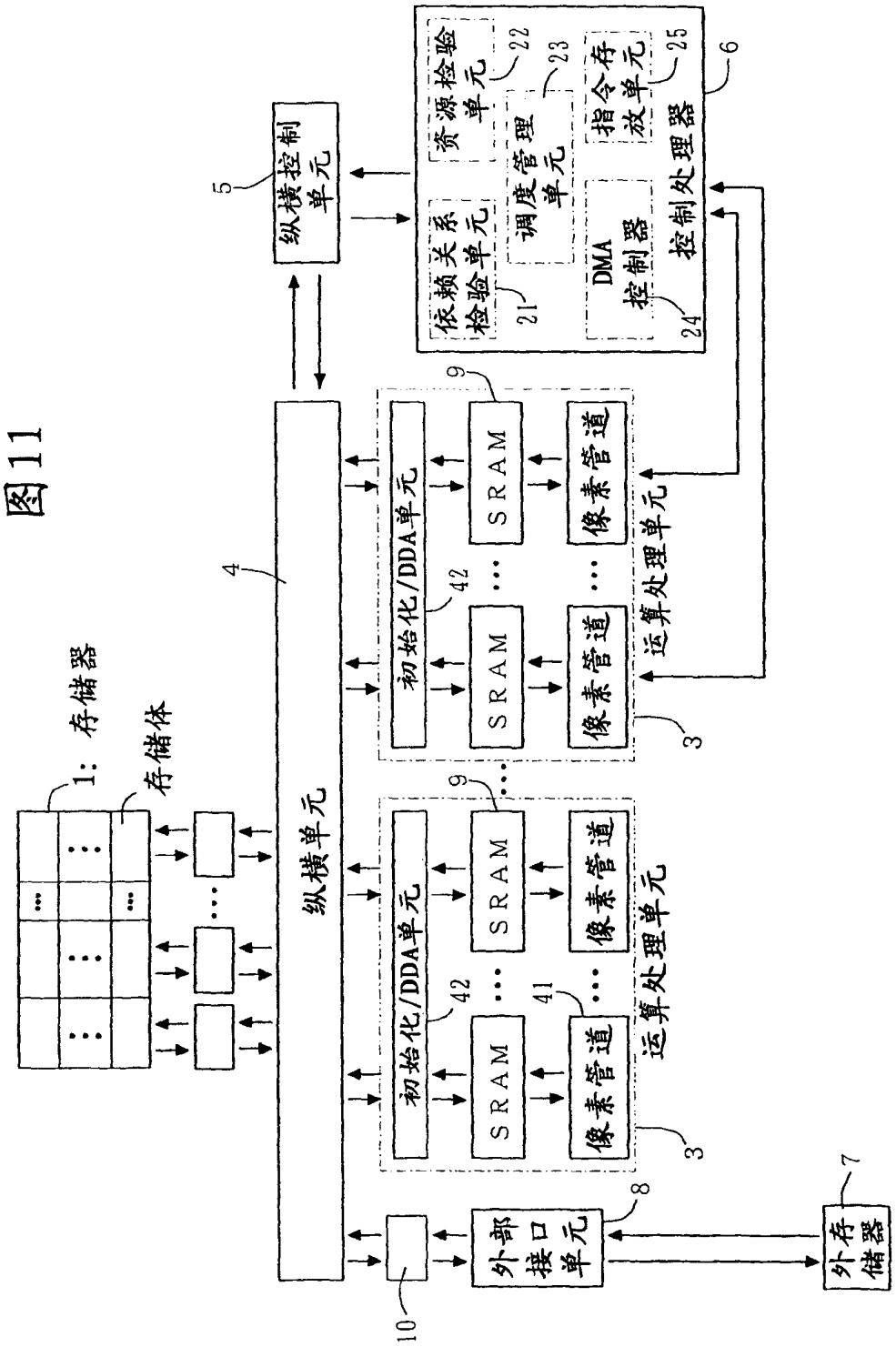


图11


1: 存储器
存储体
纵横单元
4
纵横控制单元
5
资源检验单元
22
依赖关系检验单元
21
调度管理单元
23
指令存放单元
25
DMA控制器
24
控制处理器
6
初始化/DDA单元
42
SRAM
像素管道
41
运算处理单元
3
9
外部接口单元
8
外存储器
7
10