



(12) 发明专利

(10) 授权公告号 CN 1957329 B

(45) 授权公告日 2010.05.12

(21) 申请号 200580017063.7

代理人 王波波

(22) 申请日 2005.05.20

(51) Int. Cl.

(30) 优先权数据

G06F 9/46 (2006.01)

04102350.8 2004.05.27 EP

(56) 对比文件

(85) PCT申请进入国家阶段日

CN 1168728 A, 1997.12.24, 全文.

2006.11.27

POPLAVKO P ET AL. "Task-level Timing Models for Guaranteed Performance in Multiprocessor Networks-on-Chip". INTERNATIONAL CONFERENCE ON COMPILERS, ARCHITECTURE AND SYNTHESIS FOR EMBEDDED SYSTEMS, 30 October 2003. 2003, 63-72.

(86) PCT申请的申请数据

PCT/IB2005/051648 2005.05.20

(87) PCT申请的公布数据

W02005/116830 EN 2005.12.08

(73) 专利权人 NXP 股份有限公司

地址 荷兰艾恩德霍芬

审查员 张蕾

(72) 发明人 M·J·G·贝库伊

(74) 专利代理机构 中科专利商标代理有限责任

公司 11021

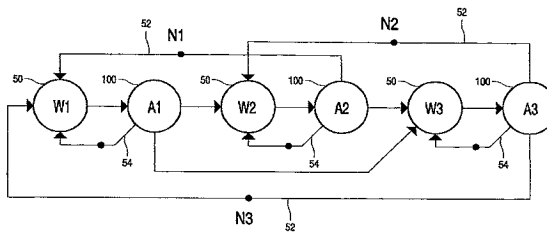
权利要求书 4 页 说明书 13 页 附图 5 页

(54) 发明名称

信号处理装置

(57) 摘要

信号流处理作业包含多个任务 (100), 每个任务 (100) 通过重复执行一个操作而实施, 该操作处理来自流的数据块。每个作业包括多个彼此流通信的任务 (100)。为使信号流通信而相互耦合的多个处理单元 (10) 执行所述多个任务。对每个独立作业实施初始计算, 用以在相应环境中执行各个作业任务时确定该作业要求的执行参数, 来支持所要求的最低流吞吐率, 所述相应环境中至多间隔为该任务规定的循环时间 T 就出现启动任务执行的机会。在运行时, 选择用于执行的作业组合。将选定作业组合的任务组分配给各个处理单元 (10), 为每个特定处理单元 (10) 检查: 分配给该特定处理单元 (10) 的任务的最坏情况执行时间总和没有超出该特定处理单元 (10) 上分配的任何组合 (100) 所规定的循环时间 T。处理单元 (10) 并发执行选定作业组合, 每个处理单元 (10) 时分多路复用分配给该处理单元 (10) 的任务 (100) 组的执行。



1. 一种用于执行信号流处理作业组合的系统,其中作业包含多个任务(100),每个任务(100)通过重复执行一个操作来实施,该操作处理来自任务(100)接收流的数据块和/或从任务(100)产生的流输出一个块,每个作业包括多个彼此流通信的任务(100),该系统被设置执行一种检查,以确定是否满足实时要求,该系统包括:

多个相互耦合的处理单元(10),用于信号流的通信;

初始计算单元(60),它被设置来为每个作业单独地实施初始确定,用以在相应环境中执行了每个作业任务时确定支持需要的最小流吞吐率所要求的作业执行参数,所述相应环境中启动任务执行的机会至多间隔为了该任务而规定的循环时间T而出现;

控制单元(62),用于在运行时间选择应当并行执行的任务组合;将选定作业组合的任务组分配给各个处理单元(10);以及为每个处理单元(10)检查:分配给该处理单元(10)的任务最坏情况执行时间总和没有超出对该处理单元(10)分配的任何任务所规定的循环时间T;

其中各个处理单元(10)并发执行选定的作业组合,每个处理单元(10)时分多路复用对分配给该处理单元(10)的该组任务(100)的执行,

其中至少设置一个处理单元(10)在没有足够可用块实施任务(100)的操作和/或没有足够可用缓冲空间来写入操作结果块时跳过分配给该处理单元(10)的该组的任务的执行。

2. 根据权利要求1的系统,其中设置初始计算单元(60)计算缓冲器的缓冲存储尺寸,用于缓冲各个任务(100)对之间的块,由此缓冲器尺寸足以确保满足吞吐率,保留至少所计算尺寸的缓冲器存储空间,用以在执行期间在任务(100)对之间缓冲。

3. 一种用于处理信号流处理作业的组合的方法,该方法包括实施一种检查来确定是否满足实时要求,该方法包括步骤:

规定多个处理任务(100),每个处理任务(100)通过重复执行一个操作来实施,该操作处理来自任务(100)接收流的数据块和/或从任务(100)产生的流输出一个块;

规定多个作业,每个作业包括多个彼此流通信的处理任务(100);

对每个作业独立地实施初始确定,用以在相应环境中执行了每个作业任务(100)时确定支持需要的最小流吞吐率所要求的作业执行参数,所述相应环境中启动任务执行的机会至多间隔为了该任务而规定的循环时间T而出现;

选择并行执行的作业组合;

将选定作业组合的多组任务(100)分配给各个处理单元(10),为每个处理单元(10)检查:分配给该处理单元(10)的任务的最坏情况执行时间总和没有超出对该处理单元(10)分配的任何任务(100)所规定的循环时间T;

由多个处理单元(10)并发执行选定作业组合,时分多路复用这些组任务的执行,

其中在没有足够的可用块来执行该任务的操作和/或没有足够的可用缓冲空间来写入操作的结果块时在所述组中跳过这一任务(100)的执行。

4. 根据权利要求3的方法,其中初始确定的实施包括计算缓冲器的缓冲存储尺寸,用于缓冲各个任务(100)对之间的块,由此缓冲器尺寸足以确保满足吞吐率,保留至少所计算尺寸的缓冲器存储空间,用以在执行期间在缓冲任务(100)对之间缓冲。

5. 根据权利要求4的方法,其中通过以下步骤计算至少一个在第一和第二任务之间缓

冲数据的缓冲尺寸：

识别作业的连续任务 (100) 的路径,其中在每个路径上,该路径中每个连续任务 (100) 依赖于该路径中前一任务 (100) 的性能来启动操作,每个路径从第一任务 (100) 开始并在第二任务 (100) 结束；

为每个识别出的路径计算与沿该路径的任务的最坏情况执行时间总和有关的信息,加上在相应环境中执行时该任务 (100) 被赋予执行机会之前的最大等待时间,所述相应环境中启动任务 (100) 执行的机会至多间隔在由任务 (100) 规定的循环时间而出现；

根据任何所识别的路径的所述总和的最大值同连续块之间所要求的最大吞吐量时间的比率确定缓冲器尺寸。

6. 根据权利要求 3 的方法,其中所述初始确定的实施包括选择作业的任务 (100) 的一个子组,以便通过多个处理单元中的一个通用单元以时分多路复用来执行,当在相应环境中执行该作业的每个任务 (100) 时确定所要求的多个执行参数是否支持所要求的最低流吞吐量,所述相应环境中启动任务 (100) 的子组执行的机会至多间隔为该子组所规定的循环时间 T 而出现。

7. 根据权利要求 3 的方法,其中所述初始确定的执行包括实施这样一种确定,即确定是否可能在所述相应环境中保证一直满足吞吐量。

8. 根据权利要求 7 的方法,包括当不能保证一直满足吞吐量时减小为至少任务 (100) 之一而规定的循环时间 T,并以这个降低的循环时间重复初始确定的所述实施。

9. 根据权利要求 4 的方法,其中所述初始确定的执行包括基于对同步数据流 SDF 图的分析,来计算任务之间要求的缓冲器尺寸。

10. 一种用于执行信号流处理作业组合设备,其中作业包含多个任务 (100),每个任务 (100) 通过重复执行一个操作来实施,该操作处理来自任务 (100) 接收的流的数据块并且/或从任务 (100) 产生的流输出一个块,每个作业包括多个彼此流通信的处理任务 (100),该设备被设置执行一种检查,以确定是否满足实时要求,该设备包括：

多个耦合的处理单元 (10),用于信号流的通信；

控制单元 (62),用于在运行时间选择应该并行执行的任务组合；将选定作业组合的任务组分配给各个处理单元 (10)；为每个处理单元检查：分配给该处理单元的任务最坏情况执行时间总和没有超出对该处理单元 (10) 分配的任何任务所规定的循环时间 T；

其中各个处理单元 (10) 并发执行选定的作业组合,每个处理单元 (10) 时分多路复用对分配给该处理单元 (10) 的任务组的执行,

其中在没有足够的可用块来执行该任务的操作和/或没有足够的可用缓冲空间来写入操作的结果块时在所述组中跳过这一任务 (100) 的执行。

11. 一种用于计算作业所要求的执行参数的装置,其中作业包含多个任务 (100),每个任务 (100) 通过重复执行一个操作来实施,该操作处理来自任务 (100) 接收的流的数据块和/或从任务 (100) 产生的流输出一个块,每个作业包括多个彼此流通信的处理任务 (100),该装置包括：

初始计算单元,被设置来为各个作业独立地实施初始计算,用以在相应环境中执行每个作业任务时确定该作业所要求的执行参数,来支持所要求的最低流吞吐量,所述相应环境中启动执行的机会至多间隔由该任务规定的循环时间 T 而发生,

所述初始计算单元包括：用于初始计算的所述实施包括计算缓冲器的缓冲存储尺寸，以缓冲各个任务（100）对之间的块的单元，由此缓冲器尺寸足以确保满足吞吐率，保留至少所计算尺寸的缓冲存储空间，用以在执行期间在任务对之间进行缓冲，

其中所述初始计算单元包括：用于实施这样一种确定的单元，即在所述环境中确定是否总能保证满足吞吐率，当不能保证一直满足吞吐率时减小为至少这些任务（100）之一而规定的循环时间，并以这个降低的循环时间重复初始确定的所述实施。

12. 根据权利要求 11 的装置，还包括：用于计算至少一个在第一和第二任务之间缓冲数据的缓冲尺寸的单元，该单元包括：

用于识别作业的连续任务（100）的路径的单元，其中在每个路径上，每个连续任务（100）依赖于该路径上前一任务（100）的性能来启动操作，每个路径从第一任务（100）开始并在第二任务（100）处结束；

用于为每个识别路径计算与沿该路径任务（100）的最坏情况执行时间总和有关的信息的单元，加上在相应环境中执行时该任务（100）被赋予执行机会之前的最大等待时间，所述相应环境中启动任务执行的机会至多间隔为任务（100）所规定的循环时间 T 而出现；

用于从任何识别路径的最大所述总和同连续块之间所要求的最大吞吐量时间的比率确定缓冲器尺寸的单元。

13. 一种用于处理信号流处理作业的组的方法，该方法包括执行一种检查来确定是否满足实时要求，该方法包括步骤：

规定多个处理任务（100），每个处理任务（100）通过重复执行一个操作来实施，该操作处理来自任务（100）接收的流的数据块和 / 或从任务（100）产生的流输出一个块；

规定多个作业，每个作业包括多个彼此流通信的处理任务（100）；

选择并行执行的作业组合；

将选定作业组合的任务（100）组分配给各个处理单元（10），为每个处理单元检查：分配给该处理单元（10）的任务最坏情况执行时间总和没有超出对该处理单元（10）分配的任何任务（100）所规定的预定循环时间 T；

并发执行选定作业组合，时分多路复用任务组的执行，

其中在没有足够的可用块来执行该任务的操作和 / 或没有足够的可用缓冲空间来写入操作的结果块时在所述组中跳过这一任务（100）的执行。

14. 一种计算用来执行信号流处理作业组合的执行参数的方法，该方法包括：

规定多个处理任务（100），每个处理任务（100）通过重复执行一个操作来实施，该操作处理来自任务（100）接收的流的数据块和 / 或从任务（100）产生的流输出一个块；

规定多个作业，每个作业包括多个彼此流通信的处理任务（100）；

为每个作业独立地实施初始计算，用以在相应环境中执行每个作业任务时确定该作业所要求的执行参数，以支持所要求的最低流吞吐率，所述相应环境中启动执行任务（100）的机会至多间隔为该任务规定的循环时间 T 而出现，

其中初始计算的所述实施包括计算缓冲器的缓冲存储尺寸，以缓冲各个任务（100）对之间的块，由此缓冲器尺寸足以确保满足吞吐率，保留至少所计算尺寸的缓冲存储空间，用以在执行期间在任务对之间进行缓冲，

其中初始计算的所述实施包括实施这样一种确定，即在所述环境中确定是否总能保证

满足吞吐率,当不能保证一直满足吞吐率时减小为至少这些任务(100)之一而规定的循环时间,并以这个降低的循环时间重复初始确定的所述实施。

## 信号处理装置

### 技术领域

[0001] 本发明涉及处理信号流的装置,并涉及一种操纵这种装置的方法以及一种制造这种装置的方法。

### 背景技术

[0002] 在诸如电视 / 互联网访问设备、图形处理器、摄像机、音频设备等用于媒体访问的设备中要求信号流处理。现代设备要求执行的流处理计算量越来越大。流处理涉及在信号单元到达的同时处理 (至少原则上) 此类信号单元的无限流的连续信号单元。

[0003] 在这种设备中,流处理计算的实现必须满足几个要求:必须满足实时信号流处理约束条件,必须可以执行灵活的作业组合并且必须每秒能够执行大量计算。需要实时流处理要求是为了,例如,避免音频再现中的迟钝、冻结显示图像或由于缓冲过剩造成的废弃输入音频或视频数据。需要灵活性要求,是因为用户必须能够在运行时间选择应该同时执行哪些信号处理作业组合,总是满足实时约束条件。大量计算的要求通常意味着:应该将所有的这些都实现在一个多处理器的系统中,所述多个处理器并行工作,执行作为信号处理作业一部分的不同任务。

[0004] 在这种灵活的分布式系统中,很难保证能够满足实时约束条件。产生数据需要的时间不仅依赖于实际计算时间,而且依赖于处理器等待输入数据和等待缓冲间隔以便能够写输出数据的等待时间,直到处理器能用的等待时间等等。不可预知的等待会使实时性能变得不可预知。如果进程彼此等待产生数据和或释放资源,等待可能会导致死锁。

[0005] 尽管在通常情况下等待不会妨碍实时性能,但在特殊情况下也会无法满足实时约束条件,所述特殊情况为信号数据导致在异常 (但并非错误) 短或长的时间内完成流组块的某些计算任务时。当然,可以让用户自行尝试设备是否总是支持作业组合。但这会导致用户只能在事后发现视频信号的一部分没有被记录下来或者该系统在无法预知的时刻崩溃。尽管在某些系统中,用户被迫接受这种性能,但这当然不十分令人满意。

[0006] 称为同步数据流图 (SDF) 的理论框架的使用为个别作业提供了这一问题的解决方案。SDF 图背后的理论使得能够事先计算将流处理作业任务分布到多个处理器中实施时是否能够在所有情况下满足实时约束条件或其它吞吐量要求。SDF 图理论的基本方法是为并行执行所有任务的理论处理器组计算执行时间。SDF 图理论证明,在某些条件下,为该组理论处理器计算的吞吐量 (产生流连续部分之间需要的时间) 总是低于任务的实际实施的吞吐量。因此,如果证明任务组合对于理论处理器组而言是实时工作的,则可保证实际实施的实时性能。

[0007] 通过将必须执行的作业分成任务来构造 SDF 图。这些任务对应 SDF 图中的节点。典型地,通过反复执行从其它任务或向其它任务输入和 / 或输出一个或多个输入数据流的操作来实施每个任务。SDF 图节点之间的连线 (edge) 表示任务之间的流通信。在理论处理器组中,由各个处理器执行每个任务的操作。在启动操作执行之前,理论处理器等待足够的时间。在 SDF 模型中,假定每个流都由连续“标记 (token)”组成,每个“标记”对应流数据

的各个块。当处理器的输入端有指定数目的标记可用时便假定该处理器立即开始处理,从其输入端输入(移除)标记,并且在其输出端产生结果标记之前占用预定时间间隔。对于这种理论模型,可以计算输出标记的时间点。

[0008] 为了将这些计算出来的理论时间点转换成实际组处理器的最坏情况时间点,首先必须将理论处理器要求的预定时间间隔持续时间选为等于(或长于)实际处理器所需的最坏情况时间间隔。

[0009] 其次,必须使理论模型“认识到”实际处理器的多种限制。例如,实际中,如果处理器仍在为先前的标记处理操作,那么它无法启动一个操作的执行。这一限制在 SDF 图中通过添加从节点出发回到该节点的“自连线”表示。对应这个节点的处理器被建模为,在开始执行之前要求来自于这一自连线的一个标记并且在执行结束时输出一个标记。当然,每个执行期间也处理来自处理器常规输入的标记。将自连线初始化为包含一个标记。这样,给定理论处理器组以下的实际特性,即在执行一个标记任务之前必须等待先前标记执行完毕。类似地,可使 SDF 图认识到由于缓冲容量导致的实际限制,这在输出缓冲器内没有可用空间时将导致处理器等待。

[0010] 实际处理器的其它限制一般由如下情形导致,即每个处理器通常以时间多路方式执行多个不同任务的操作。这意味着,实际启动操作执行不仅要等待标记的可用性,而且要等待由相同处理器执行的其它任务操作的完成。在某些条件下,这一限制可以在 SDF 图中表现出来。特别地,当存在多路任务将按其执行的预定顺序时,这在 SDF 图上可通过根据该预定顺序附加从一个多路任务到下一个多路任务的循环连线并将初始标记附加到这一循环的第一连线上表现出来。这样,理论处理器组被赋予了以下实际特性,即循环中每个任务执行的开始都要等待先前任务完成。

[0011] 应该注意,这种使 SDF 图模型“认识”实际实施的限制的方法并非对所有可能的限制都适用。例如,如果没有预先确定处理器执行时间多路任务的顺序,则在 SDF 图中无法表示出定时的顺序。因此,例如,如果没有足够标记来启动特定任务时便安排处理器跳过该特定任务(行进至下一个任务),无法在 SDF 图中表示出这种效果。实际上,这意味着在这种情况下不可能保证实时吞吐量。因此,实时保证要付出高昂代价:仅可使用某些实施。通常可以说,为了适应 SDF 图理论,实施必须满足“单调性条件”:更快速的任务执行不能导致任何其它任务的更迟执行。

[0012] 此外,应该注意,很难将 SDF 图理论应用到多个并行作业的灵活组合的执行中。原则上,这要求并行执行的所有不同作业任务必须包含在同一个 SDF 图中。这在表示任务间的相互定时影响时是需要的。但是,如果不同作业的输入和/或输出数据速率不同步,则无法以这种方式提供实时保证。此外,每当附加或从作业组中删除一个作业时便执行一次新的吞吐量时间计算将需要相当可观的开销。

## 发明内容

[0013] 本发明的目的之一是通过能在运行时间以较小开销应用的 SDF 图理论技术来提供实时保证。

[0014] 本发明的目的之一是在必须通过处理器组执行灵活作业组合时降低使用 SDF 图理论技术提供实时保证所需的计算量。

[0015] 本发明的目的之一是在必须通过处理器组执行非同步作业的灵活组合时提供实时保证。

[0016] 本发明的目的之一是能够在多处理器电路中提供实时保证,所述多处理器电路中一个处理器在循环的基础上执行多个任务,若没有充分的输入数据可用于前一任务,则继续行进至循环序列中的下一个任务。

[0017] 本发明的目的之一是使用 SDF 图理论在较小的资源浪费下提供实时保证。

[0018] 本发明提供了一种用于执行信号流处理作业组合的系统,其中作业包含多个任务,每个任务通过重复执行一个操作来实施,该操作处理来自任务接收流的数据块和 / 或从任务产生的流输出一个块,每个作业包括多个彼此流通信的任务,该系统被设置执行一种检查,以确定是否满足实时要求,该系统包括:多个相互耦合的处理单元,用于信号流的通信;初始计算单元,它被设置来为每个作业单独地实施初始确定,用以在相应环境中执行了每个作业任务时确定支持需要的最小流吞吐率所要求的作业执行参数,所述相应环境中启动任务执行的机会至多间隔为了该任务而规定的循环时间  $T$  而出现;控制单元,用于在运行时间选择应当并行执行的任务组合;将选定作业组合的任务组分配给各个处理单元;以及为每个处理单元检查:分配给该处理单元的任务最坏情况执行时间总和没有超出对该处理单元分配的任何任务所规定的循环时间  $T$ 。各个处理单元并发执行选定的作业组合,每个处理单元时分多路复用对分配给该处理单元的该组任务的执行。至少设置一个处理单元在没有足够可用块实施任务的操作和 / 或没有足够可用缓冲空间来写入操作结果块时跳过分配给该处理单元的该组的任务的执行。

[0019] 本发明还提供了一种用于处理信号流处理作业的组合的方法,该方法包括实施一种检查来确定是否满足实时要求,该方法包括步骤:规定多个处理任务,每个处理任务通过重复执行一个操作来实施,该操作处理来自任务接收流的数据块和 / 或从任务产生的流输出一个块;规定多个作业,每个作业包括多个彼此流通信的处理任务;对每个作业独立地实施初始确定,用以在相应环境中执行了每个作业任务时确定支持需要的最小流吞吐率所要求的作业执行参数,所述相应环境中启动任务执行的机会至多间隔为了该任务而规定的循环时间  $T$  而出现;选择并行执行的作业组合;将选定作业组合的多组任务分配给各个处理单元,为每个处理单元检查:分配给该处理单元的任务的最坏情况执行时间总和没有超出对该处理单元分配的任何任务所规定的循环时间  $T$ 。由多个处理单元并发执行选定作业组合,时分多路复用这些组任务的执行。在没有足够的可用块来执行该任务的操作和 / 或没有足够的可用缓冲空间来写入操作的结果块时在所述组中跳过这一任务的执行。

[0020] 本发明还提供了一种用于执行信号流处理作业组合设备,其中作业包含多个任务,每个任务通过重复执行一个操作来实施,该操作处理来自任务接收的流的数据块并且 / 或从任务产生的流输出一个块,每个作业包括多个彼此流通信的处理任务,该设备被设置执行一种检查,以确定是否满足实时要求,该设备包括:多个耦合的处理单元,用于信号流的通信;控制单元,用于在运行时间选择应该并行执行的任务组合;将选定作业组合的任务组分配给各个处理单元;为每个处理单元检查:分配给该处理单元的任务最坏情况执行时间总和没有超出对该处理单元分配的任何任务所规定的循环时间  $T$ 。其中各个处理单元并发执行选定的作业组合,每个处理单元时分多路复用对分配给该处理单元的任务组的执行。在没有足够的可用块来执行该任务的操作和 / 或没有足够的可用缓冲空间来写入操作

的结果块时在所述组中跳过这一任务的执行。

[0021] 本发明还提供了一种用于计算作业所要求的执行参数的装置,其中作业包含多个任务,每个任务通过重复执行一个操作来实施,该操作处理来自任务接收的流的数据块和/或从任务产生的流输出一个块,每个作业包括多个彼此流通信的处理任务,该装置包括:初始计算单元,被设置来为各个作业独立地实施初始计算,用以在相应环境中执行每个作业任务时确定该作业所要求的执行参数,来支持所要求的最低流吞吐率,所述相应环境中启动执行的机会至多间隔由该任务规定的循环时间  $T$  而发生。所述初始计算单元包括:用于初始计算的所述实施包括计算缓冲器的缓冲存储尺寸,以缓冲各个任务对之间的块的单元,由此缓冲器尺寸足以确保满足吞吐率,保留至少所计算尺寸的缓冲存储空间,用以在执行期间在任务对之间进行缓冲。所述初始计算单元包括:用于实施这样一种确定的单元,即在所述环境中确定是否总能保证满足吞吐率,当不能保证一直满足吞吐率时减小为至少这些任务之一而规定的循环时间,并以这个降低的循环时间重复初始确定的所述实施。

[0022] 本发明还提供了一种用于处理信号流处理作业的组合的方法,该方法包括执行一种检查来确定是否满足实时要求,该方法包括步骤:规定多个处理任务,每个处理任务通过重复执行一个操作来实施,该操作处理来自任务接收的流的数据块和/或从任务产生的流输出一个块;规定多个作业,每个作业包括多个彼此流通信的处理任务;选择并行执行的作业组合;将选定作业组合的任务组分配给各个处理单元,为每个处理单元检查:分配给该处理单元的任务最坏情况执行时间总和没有超出对该处理单元分配的任何任务所规定的预定循环时间  $T$ ;并发执行选定作业组合,时分多路复用任务组的执行,其中在没有足够的可用块来执行该任务的操作和/或没有足够的可用缓冲空间来写入操作的结果块时在所述组中跳过这一任务(100)的执行。

[0023] 本发明还提供了一种计算用来执行信号流处理作业组合的执行参数的方法,该方法包括:规定多个处理任务,每个处理任务通过重复执行一个操作来实施,该操作处理来自任务接收的流的数据块和/或从任务产生的流输出一个块;规定多个作业,每个作业包括多个彼此流通信的处理任务;为每个作业独立地实施初始计算,用以在相应环境中执行每个作业任务时确定该作业所要求的执行参数,以支持所要求的最低流吞吐率,所述相应环境中启动执行任务的机会至多间隔为该任务规定的循环时间  $T$  而出现。初始计算的所述实施包括计算缓冲器的缓冲存储尺寸,以缓冲各个任务对之间的块,由此缓冲器尺寸足以确保满足吞吐率,保留至少所计算尺寸的缓冲存储空间,用以在执行期间在任务对之间进行缓冲,其中初始计算的所述实施包括实施这样一种确定,即在所述环境中确定是否总能保证满足吞吐率,当不能保证一直满足吞吐率时减小为至少这些任务之一而规定的循环时间,并以这个降低的循环时间重复初始确定的所述实施。

[0024] 根据本发明,通过一个两阶段过程保证多个同时执行的流处理作业的实时吞吐量。在第一阶段,孤立考虑各个作业,并且将这些作业的执行参数(诸如用于缓冲来自任务之间流的数据的缓冲器尺寸)选择在一个假定范围内,其中至多间隔为该任务规定的周期  $T$  就出现启动任务执行的机会。优选地,还需要检查是否能够根据所需的实时要求执行作业,即是否能以最多一个指定延迟产生连续数据块。在第一阶段,不必知道必须同时执行哪些流处理作业的组合。

[0025] 在第二阶段,考虑并发执行的作业的组合。在这一阶段,多个处理单元中的每

一个都从选定作业组合中分配一组任务。分配期间,为每个特定处理单元检查分配给该特定处理单元的任务的最坏情况执行时间总和不超过分配给该特定处理单元的任何任务所规定的循环时间 T。这个总和反映在处理器单元使用调度算法处理任务(例如,循环调度)的条件下最坏情况执行时间如何影响连续执行机会之间的最大可能延迟。最后,并发执行选定的作业组合,时分多路复用各个处理单元上任务循环的执行。通常,处理单元不必等到任务可以执行,如果使用了保证实时性能的发明过程,对由于缺少输入和/输出缓冲空间而无法行进的任务处理单元可跳至下一个任务。这在提高处理相互不同步的数据流的不同作业的性能方面是非常有利的。

[0026] 优选将所有任务的循环时间 T 选为相同。这简化第二阶段的操作。但是,根据第二实施例,当实时要求不能得到满足时便对选定任务的循环时间进行调整。通过降低特定任务的循环时间,我们能够有效允许相同的处理单元处理更少的特定任务,以便提高性能。循环时间的调整使得能够在第一阶段搜索可能的实时实施,即在不知道那些必须并行执行的任务组合时。

[0027] 可通过 SDF 图技术计算假定情况下所需的最小缓冲器尺寸。一个实施例中,通过在过程的 SDF 图中将虚拟节点附加在真实任务节点之前来计算缓冲器尺寸。将这些虚拟节点的最坏情况执行时间设置为表示最坏情况延迟,这种最坏情况延迟由执行一个任务循环时等待处理单元到达任务引起。然后,通过考虑从产生数据流的节点到消耗该数据流的另一个节点穿过 SDF 图的所有路径,并确定每条路径上的节点的最坏情况执行时间的总和,来确定缓冲器尺寸。正如实时吞吐量要求确定的那样,由连续标记间最大允许时间分割后,这些总和中的最大者可用来确定缓冲器尺寸。

### 附图说明

[0028] 通过下面的附图对本发明的这些和其它目的和优势进行更详细的阐述,这些附图对实施例的非限制性示例进行说明。

[0029] 图 1 示出一个多处理器电路的示例

[0030] 图 1a-c 示出一个简单作业的 SDF 图

[0031] 图 2 示出一个保证实时性能的过程的流程图

[0032] 图 3 示出一个保证实时性能的两阶段过程的流程图

[0033] 图 4 示出保证实时性能的两阶段过程中一个步骤的流程图

[0034] 图 5 示出一个简单作业的精细 SDF 图

[0035] 图 6 示出一个实现本发明的典型系统

### 具体实施方式

[0036] 图 1 示出一个多处理器电路的示例。该电路包含多个处理单元 10,它们通过互连电路 12 互相连接。尽管只示出了 3 个处理单元 10,但应该理解可提供更多或更少数目的处理单元。每个处理单元包含一个处理器 14、一个指令存储器 15、一个缓冲存储器 16 和一个互连接口 17。应该理解,尽管没有示出,但处理单元 10 可包含其它元件,诸如数据存储器、高速缓冲存储器等。每个处理单元中,处理器 14 耦合到指令存储器 15,并经由缓冲存储器 16 和互连接口 17 耦合到互连电路 12。例如,互连电路 12 包括一个总线或一个网络等,用

于在处理单元 10 之间传输数据。

[0037] 操作中,多处理器电路能够并行执行多个信号处理作业。一个信号处理作业涉及各自的多个任务,一个作业的不同任务可由不同处理单元 10 执行。信号处理应用的一个示例是这样一种应用,它涉及两个 MPEG 流的 MPEG 解码以及所述流中视频部分数据的混合。这样一个应用可被划分为多个作业,例如两个 MPEG 解码作业、一个音频解码作业、一个视频混合作业以及一个对比度校正作业。反过来,每个作业涉及一个或多个重复执行的任务。例如,一个 MPEG 作业包括一个可变长度解码任务、一个余弦块变换任务等。

[0038] 一个作业的不同任务由不同处理单元 10 并行执行。例如,这样做是为了实现足够的吞吐量。由不同处理单元执行任务的另一个原因是某些处理单元 10 专门高效处理某些任务,而其它处理单元专门高效执行其它任务。每个任务输入和 / 或输出一个或多个信号数据流。这些信号数据流被分组到预定最大尺寸的块中(通常表示预定时间间隔的信号数据或图像的预定部分,并优选为预定尺寸),例如这包括一个传输数据包、单个像素或单个像素线的数据、一个  $8 \times 8$  像素块、一个像素帧、一个音频采样、一定时间间隔的音频采样集合等等。

[0039] 在作业的执行期间,对每个任务而言,重复执行对应该任务的操作,每次将流的预定数目块(例如一个块)用作输入和 / 或产生预定数目块作为输出。一个任务的输入数据块通常由其它任务产生,并且输出数据块通常由其它任务使用。当第一任务输出被第二任务使用的流块时,将在输出之后和使用之前将该流块缓冲在缓冲存储器 16 中。如果第一和第二任务由不同处理单元 10 执行,则流块经由互连电路 12 传送到将该流块用作输入的处理单元 10 的缓冲存储器 16 中。

#### [0040] SDF 图理论

[0041] 在 SDF(同步数据流)图理论的基础上对多处理器电路的性能进行管理。从现有技术可基本了解 SDF 图理论本身。

[0042] 图 1a 示出一个 SDF 图示例。SDF 图理论概念性地将应用描绘为带有“节点”100 的图,所述节点 100 对应不同的任务。这些节点由直接“连线”102 连接,直接连线 102 连接节点对,并表示流块由对应该节点对中第一节点的任务输出并由对应该节点对中第二节点的任务使用。流块由“标记(token)”符号表示。对每个节点规定在执行相应任务之前应该在其输入连接处出现多少标记,并规定执行时任务将输出多少标记。在产生流块之后且在其被使用之前,在一条连线上呈现一个标记。这对应于流块在缓冲存储器 16 中的存储。连线上标记的出现与否规定了 SDF 图的状态。当节点“消耗”一个或多个标记和 / 或产生一个或多个标记时,这个状态发生改变。

[0043] SDF 图基本上描绘了数据流和一个作业执行期间的处理操作、同可在一个操作中处理的数据流块相对应的标记。但是,也可在 SDF 图中表示诸如总线访问判优、并行性执行量限制、缓冲器尺寸限制等多个方面。

[0044] 例如,可通过附加一个表示传输任务的节点来建模经由总线或网络的传输(假定使用了总线或网络访问机制,确保在预定时间内的访问)。作为另一个示例,原则上假定一旦存在足够的可用输入标记,图形中的任何节点就会启动任务的执行。这隐含这样一个假定,即在先任务的执行不妨碍执行的启动。这可通过不限制数目地为同一个任务提供并行处理器来保证。实际上,处理器的数目当然是有限的,通常限制为不多于一个,这意味着在

前一执行完成之前不能开始下一个任务的执行。图 1b 示出这是如何通过将“自连线”104 附加到 SDF 图上来建模的,每个自连线 104 都从节点出发并回到该节点,自连线上初始附带多个标记 106,对应可并行实施的执行数量,例如一个标记 106。这表示可通过消耗该标记来初始启动任务,但其在该任务完成并由此替换该标记之前不能再次启动。实际上,由于一个节点上任务的有限启动可能性自动隐含对将被启动的连接节点上的任务的启动次数限制,因此将此类自连线附加到选定节点就足够了。

[0045] 图 1c 示出一个示例,其中通过附加返回连线 108 并通过在这个返回连线 108 上安置多个标记 110 来表达对缓冲器的尺寸限制,该缓冲器用于从第一任务向第二任务通信,所述返回连线 108 从第二任务的节点返回到第一任务的节点,标记 110 的数目对应可在缓冲器中存储的流块数目。这表示第一任务最初可执行同初始标记相对应的次数,并且只有第二任务完成执行并由此替代标记时才可能实施后续执行。

[0046] SDF 图表示从任何特定实施中提取出来的任务之间的数据通信。为了清楚地呈现,可认为每个节点对应一个专门执行相应任务的处理器,并且可认为每个连线对应一个通信连接,在一对处理器之间包括一个 FIFO 缓冲器。但是,从下述提取 SDF 图:它同样表示这样的情况,其中由相同的处理器执行不同任务,并且不同任务的流块经由一个诸如总线或网络的共享连接进行通信。

[0047] SDF 图理论的主要吸引力之一是它通过实施 SDF 图的处理器支持最坏情况吞吐量的预测。这一预测的起始点是带有自定时处理单元的 SDF 图理论实施,每一个都专用于特定任务,并且每一个都被安排为在接收到足够的输入标记来执行任务时便立即启动执行任务。这一理论实施中,假定每个处理单元对其相应任务的每次执行都要求一个预定执行时间。

[0048] 对于这一实施,可以很容易地计算任务(由标注“v”识别)的各个执行(由不同标注值  $k = 0, 1, 2, \dots$  识别)的启动时间  $s(v, k)$ 。通过有限的计算量可为无限数目的  $k$  值确定启动时间  $s(v, k)$ ,因为现有技术已通过 SDF 图理论证明,这一实施导致启动时间  $s(v, k)$  的重复模式:

$$[0049] \quad s(v, k+N) = s(v, k) + \lambda N$$

[0050] 在此,  $N$  表示其后模式重复的执行数目,并且  $\lambda$  是在该周期内两个连续执行之间的平均延迟,即  $1/\lambda$  是平均吞吐率,每个单位时间内产生的平均流块数目。

[0051] 现有技术 SDF 图理论已经示出,可通过识别 SDF 图中的简单循环来确定  $\lambda$  (简单循环是沿着至多包含节点一次的连线的闭合环路)。对于每个此类循环“c”,可以计算一个标称平均执行时间  $CM(c)$ ,它是该循环内节点的执行时间总和被循环连线上的初始标记数目相除得到的。 $\lambda$  是具有最长平均执行时间的循环  $c_{\max}$  的平均执行时间  $CM(c_{\max})$ 。类似地,现有技术 SDF 图理论提出了一种计算一个周期内执行次数  $N$  的方法。可以注意到,实际情况中的图至少包含一个循环,因为否则的话该图将对应无限数目的处理器,它们能够并行执行任务无限次,而这将导致无限的吞吐率。

[0052] 理论实施所得到的结果可用来确定 SDF 图实际实施的最小吞吐率。基本思想是确定实际实施中每个任务的最坏情况执行时间。随后,将这个最坏情况执行时间作为执行时间分配给对应理论实施中该任务的节点。SDF 图理论通过这个最坏情况执行时间为理论实施计算启动时间  $s_{th}(v, k)$ 。在一定条件下,可以保证这些最坏情况启动时间总是至少同实

实际实施中的执行启动  $s_{imp}(v, k)$  一样迟：

$$[0053] \quad s_{imp}(v, k) \leq s_{th}(v, k)$$

[0054] 这使得能够保证最坏情况吞吐率和得到数据之前的最大延迟。但是,只有在 SDF 图中对可能延迟任务执行的所有实施细节进行建模时才能提供这一保证。这将实施限制为其中的未建模方面具有单调影响的实施:一个任务的执行时间降低不会导致任何任务启动时间的延迟。

#### [0055] 预定任务组合的调度

[0056] 图 2 示出在图 1 处理电路上通过 SDF 图理论调度任务组合过程的流程图。第一步 21 中,该过程接收任务组合以及任务之间的通信的说明。第二步 22 中,该过程将特定任务的执行分配给不同的处理单元 10。由于实际电路中的处理单元数通常远小于任务数,因此至少多个处理单元 10 之一被分配多个任务。

[0057] 第三步 23 中,该过程调度一个顺序和一个相对频率,将按其执行任务(将在运行时间无限重复顺序的执行)。这个顺序必须保证不会出现死锁:处理单元 10 中的任何特定任务都直接或间接地从该处理单元 10 执行的另一个任务要求流块,应该在所述特定任务之前如此频繁地调度其它任务使其产生足够的流块来启动该特定任务。这需要被所有处理器保持。

[0058] 第四步 24 中,该过程选择存储流块的缓冲器尺寸。对于相同处理单元 10 中执行的任务,根据调度得到缓冲器尺寸的最小值,它必须能够在其它任务使用该数据或在重复调度之前将任务产生的数据存储在缓冲器中。正如下面讨论的那样,根据第六和第七步 26、27 的结果,可任意选择在不同处理单元上执行的任务之间的缓冲器尺寸。

[0059] 第五步 25 中,该过程通过特定任务及其产生节点和连线的从属关系有效形成 SDF 图表示。尽管通俗地说该过程形成 SDF 图并以一定方式修正这个图,但应该理解这意味着产生了表征至少同 SDF 图等价的信息的数据,即可由此清楚地导出这个 SDF 图的相关特性。

[0060] 该过程在调度于不同处理单元 10 上的任务节点之间的连线上以及表示缓冲器尺寸限制的连线上附加“通信处理器”节点,并附加能够并行实施的任务执行数。该过程还将各个执行时间 ET 同每个特定节点相关联,它对应在相同处理单元 10 上以相同顺序调度的任务的最坏执行时间总和 WCET,所述处理单元 10 具有对应该特定节点的特定任务。这对应从可能的输入数据直到执行完成之间的最坏情况等待时间。

[0061] 第六步 26 中,该过程分析 SDF 图,以便计算该 SDF 图的最坏情况启动时间  $s_{th}(v, k)$ ,通常包括计算前述的平均吞吐量延迟  $\lambda$  和重复频率  $N$ 。第七步 27 中,该过程测试求得的最坏情况启动时间  $S_{th}(vk)$  是否满足任务组合指定的实时要求(即,这些启动时间位于必须得到流块的指定时间点之前或之上,它们通常是周期性重复的时间点,诸如输出视频帧的时间点)。如果是这样,该过程执行第八步 28,加载任务的程序代码和信息,以便迫使调度进入调度这些任务的处理单元 10,或者至少输出随后用于这一加载的信息。如果第七步显示调度不满足实时要求,则该过程通过处理单元 10 的不同任务分配和/或在不同处理单元 10 上执行的任务之间的不同缓冲器尺寸从第二步 22 开始重复。

[0062] 执行调度任务期间,当轮到调度中的一个任务时,相关处理单元 10 一直等待,直到有足够的输入数据和输出缓冲器空间来执行该任务为止(或者等价地,该任务本身被启动之后一直等待)。也就是说,即使清楚地看到一个任务无法执行而调度中的后续任务可以

执行也不允许背离调度。这样作的原因是,背离调度将会导致违背实时约束条件。

#### [0063] 灵活的任务运行时间组合

[0064] 图 3 示出另一个过程的流程图,它将多个作业的任务动态分配给处理单元 10。这一过程包括第一步 31,其中该过程接收多个作业的说明。不必要在第一步 31 中指定哪些作业必须组合执行。每个作业都可能包含将被组合执行的多个通信任务。第二步 32 中,该过程单独为每个作业实施初始缓冲器尺寸选择。可在实际运行时间操作之前脱机实施第一步和第二步。

[0065] 在运行时间,该过程动态调度作业组合。通常,作业被逐个添加,并且如果多处理器电路执行了任何一个作业,该过程便执行第三步 33,其中该过程接收一个向所述多个作业添加一个作业的请求。第四步 34 中,在运行时间,该过程将任务分配给处理单元 10。第五步 35 中,附加作业的任务被加载至处理单元 10 并被启动(或仅当它们被提前加载时才启动)。

[0066] 优选地,第四步 34 中选定的分配为各个处理单元 10 指定各个任务顺序。指定任务的执行期间使用无阻塞执行。也就是说,尽管处理单元 10 测试其中一个处理单元 10 选定顺序的任务是否存在足够的可用标记,但如果如果没有足够的标记该处理单元 10 可跳过这个任务的执行并执行该选定顺序中有足够可用标记的下一个任务。这样,执行顺序不必同用来测试标记可用性的选定顺序对应。这使得能够执行信号流不同步的作业。

[0067] 初始缓冲器尺寸选择步骤 32 为每个任务计算输入缓冲器尺寸。这一计算基于各个作业的 SDF 图理论计算,假设在执行相同处理单元 10 上执行其它作业的最坏情况时间。

[0068] 图 4 示出图 3 中初始缓冲器尺寸选择步骤 32 的详细流程图。第一步 41 中,该过程选择一个作业。第二步 42 中,构造该作业的初始 SDF 表示,包括该作业中涉及的任务。第三步 43 中,在处理单元 10 之一如同未知其它任务一样以时间多路技术方式执行每个任务的假定下,该过程添加节点和连线来表示实际实施特性,所述未知其它任务的组合的最坏情况执行时间没有超过预定值。

[0069] 第四步 44 中,该过程对 SDF 图实施分析,用以计算任务之间要求的缓冲器尺寸。非强制地,该过程还可计算 SDF 图的最坏情况启动时间  $s_{th}(v, k)$ ,通常包括计算前述的平均吞吐量延迟  $\lambda$  和重复频率  $N$ 。第五步 45 中,该过程测试所计算的最坏情况启动时间  $s_{th}(v, k)$  是否满足任务组合指定的实时要求(也就是说,这些启动时间位于必须得到流块的指定时间点之前或之上,所述指定时间点为例如输出视频帧的时间点)。如果是这样,该过程执行第六步 46,输出信息,该信息包括用于随后加载的选定缓冲器尺寸和保留时间。然后,该过程从第一步 41 开始重复另一个作业。

[0070] 图 5 示出可用于这一目的的虚拟 SDF 图的一个示例。通过在每个特定任务 100 之前附加虚拟任务 50 的节点获得该虚拟 SDF 图。虚拟任务 50 不对应执行期间的任何真实任务,但表示延迟,这种延迟由将被分配互相同处理单元的(仍未知的)其它任务导致,这就如同特定任务 100 跟在虚拟任务 50 后面一样。此外,从每个原始节点 100 返回它的前一个虚拟任务 50 节点来添加第一附加连线 54。在图的初始状态,这些第一附加连线中的每一个都包括一个标记。这些第一附加连线 54 表示对应特定节点 100 的任务的完成会启动由虚拟任务 50 的节点表示的延迟时间间隔。

[0071] 此外,从每个特定原始节点 100 向虚拟任务的节点 50 添加第二附加连线 52,该虚

拟任务节点 50 位于具有至所述特定原始节点 100 的连线的供给节点 100 的前面。可认为将每条第二附加连线 52 初始化为具有各自的待定标记数目  $N_1$ 、 $N_2$ 、 $N_3$ 。第二附加连线 52 表示所涉及任务之间缓冲量的影响。第二附加连线 52 上的标记数  $N_1$ 、 $N_2$ 、 $N_3$  表示这些缓冲器至少能够存储的信号流块数目。第二附加连线 52 被耦合回到虚拟任务节点 50,用以表征这样一个事实,即,当向后续任务供给信号数据的缓冲存储器满了,使得需要跳过任务时,可能出现处理单元 10 上整个任务循环的等待时间。

[0072] 现已发现,可以证明,能够通过等于或高于下列表达式值的最近整数从图 5 所示类型的虚拟图计算缓冲器容量

$$[0073] \quad (\sum WCET_i) / MCM$$

[0074] 其中,  $MCM$  是要求的实时吞吐量时间(产生连续流块之间的最长时间),且  $WCET_i$  是任务(由  $i$  标注)的最坏情况执行时间。总和  $\sum$  中涉及的任务依赖于为其计算容量的缓冲器,或者在 SDF 图方面,依赖于第二附加连线 52 上的起始节点同表示缓冲器的末端节点之间出现的节点 100、50。从选定数目的任务  $i$  得到总和,所述选定数目任务出现在从末端节点到起始节点的整个 SDF 图上的最坏情况路径上。应该只考虑“简单”路径:如果图包含循环,则应该只考虑穿过任何节点都不多于一次的路径。

[0075] 例如,在图 5 所示示例中,考虑从任务 A3 返回到虚拟任务 W1 的第二附加连线 52。最初有  $N_3$ (仍然未知的一个数目)个标记出现在这条连线上,表示  $N_3$  流块尺寸的缓冲器尺寸用于从任务 A1 向任务 A3 传输数据流。现通过寻找从 W1(具有  $N_3$  个标记的连线的末端点)到 A3(这条连线的起点)穿过图的路径来计算缓冲器尺寸  $N_3$ 。存在两条这样的路径:W1-A1-W2-A2-W3-A3、W1-A1-W3-A3。由于循环,还存在其它路径,例如 W1-A1-W2-A2-W1-A2(等)-W3-A3 或者 W1-A1-W2-A2-W1-A2-W3-A2,但不应该考虑这些路径,因为这些路径通过某些节点两次。但是,在更复杂的路径中,可提供通过后向连线的路径(只要它们是简单路径)。对于两个简单路径 W1-A1-W2-A2-W3-A3、W1-A1-W3-A3 中的每一个,必须计算沿路径节点 100、50 所表示任务的最坏情况执行时间总和,并且这些总和中的最大者用来计算标记数  $N_3$ 。

[0076] 在此,最坏情况执行时间同虚拟任务 50 相关联。这些最坏情况执行时间被设为  $T-T_i$ 。其中  $T$  为循环时间。特定任务的循环时间  $T$  对应将同该特定任务一起分配给相同处理单元 10 的任务的最坏执行时间的最大允许总和(所述特定任务的执行时间包含在这个总和当中)。优选地,为每个任务分配相同的预定循环时间  $T$ 。

[0077] 能够重新执行特定任务之前的最坏情况等待时间是  $T-T_i$ ,其中  $T_i$  是该特定任务的最坏情况执行时间。

[0078] 对其它缓冲器尺寸实施类似计算,计算示例图中的数目  $N_1$  和  $N_2$ ,通过路径 W1-A1-W2-A2 和 W1-A1-W3-A3-W2-A2 计算  $N_1$ ,通过路径 W2-A2-W3-A3 和 W2-A2-W1-A1-W3-A3 计算  $N_2$ 。

[0079] 这样,在给定以循环方式执行任务的前提下,如果可得到足够的数据和输出缓冲量,那么可在每个任务都同未知的其它任务一起被处理单元 10 执行的情况下确定任务之间缓冲的最小缓冲量。

[0080] 在图 3 的第四步 34 中,在运行时间,当过程将任务分配给处理单元 10 时,该过程会为每个处理单元检验分配给同一个处理器的任务的最坏情况执行时间总和是否超过循

环时间  $T$ , 这个循环时间  $T$  是在缓冲器尺寸的脱机计算期间为任何分配任务所假定的。如果分配任务超出这个循环时间, 则为处理单元选择不同的任务分配, 直到发现分配不再超出所假定的循环时间  $T$  为止。如果没有找到这样的分配, 则该过程报告无法给出实时保证。

[0081] 如果图 4 第五步 45 显示已经脱机, 无法满足实时要求, 则可非强制性降低某些节点 100 的假定循环时间  $T$ 。一方面, 这样使得由相应虚拟任务节点 50 引入的延迟降低, 使得更容易满足实时要求。另一方面, 这样做, 在图 3 中第四步 34 期间, 对连同这样一个假定循环时间  $T$  降低的任务一起的多个任务进行调度的空间减小。

[0082] 图 6 示出实现本发明的一个典型系统。提供计算机 60 来执行图 3 中的最初步骤 32。计算机 60 具有一个输入端, 用于接收有关作业任务结构和最坏情况执行时间的信息。提供运行时间控制计算机 62, 用于组合作业。提供用户接口 64, 以使用户附加或删除作业 (这常常通过启用和停用诸如家庭视频系统的设备的功能来隐含完成)。用户接口 64 耦接到运行时间控制计算机 62, 该运行时间控制计算机 62 具有一个耦接到计算机 60 的输入端, 用于接收计算机 60 选定作业的执行参数。运行时间控制计算机 62 耦合到处理单元 10, 用以控制在哪些处理单元 10 中启用哪些任务, 以及哪些执行参数 (诸如缓冲器尺寸) 将用于处理单元 10 上。

[0083] 计算机 60 和运行时间控制计算机 62 可以是同一台计算机。或者, 计算机 60 可以是一台独立计算机, 它只在名义上耦合到运行时间控制计算机 62, 因为计算机 60 所计算的参数被存储或程序化在运行时间控制计算机 62 中, 而不要求计算机 60 和 62 之间存在永久链接。可将运行时间控制计算机 62 同处理单元 10 集成在同一个集成电路中, 或者可为运行时间控制计算机 62 和处理单元 10 提供独立电路。作为备选, 一个处理单元 10 可以作为运行时间控制计算机 62 运行。

#### [0084] 进一步实施例

[0085] 至此, 应该认识到, 本发明能够为潜在处理无限信号数据流的作业组合的同步执行提供实时保证。这通过一个两阶段过程实现。第一阶段为独立作业计算诸如缓冲器尺寸的执行参数并验证实时能力。这在这样的假定下完成, 即如果处理单元执行的任务的总循环时间不超过假定循环时间  $T$ , 作业任务由多个处理单元 10 通过时间多路技术执行, 该多个处理单元 10 执行同所述作业任务相连的其它未指定的任务。第二阶段组合这些作业, 并检查分配给相同处理单元 10 的任务的最坏情况执行时间没有超出其中任何一个任务的假定循环时间  $T$ 。

[0086] 与传统 SDF 图技术相比, 存在许多不同点: (a) 使用了两阶段过程; (b) 首先为各个作业计算实时保证; (c) 对于所执行的作业组合, 不需要完整的实时保证计算: 足以计算分配给处理单元 10 的任务序列的最坏情况执行时间总和没有超出所分配任务的任一假定循环时间; 以及 (d) 在分配任务循环中, 处理单元 10 可跳过一个任务的执行, 而不是像传统 SDF 图技术要求的那样等待足够的输入数据和输出缓冲空间。

[0087] 这具有多种优势: 可保证不相关的作业组合的实时性, 这种组合的调度要求更少的开销, 并且作业的数据供给和产生不必同步。

[0088] 应该理解, 本发明不限于公开的实施例。首先, 尽管通过 SDF 图对本发明进行了阐述, 但当过程由机器执行时不必产生明确的图形。只要产生并处理表示这些图主要特性的数据就足够了。许多其它表示也可用于这一目的。本文中, 能够理解, 也仅仅将等待任务到

图的附加阐述为方便的比喻。并没有添加实际任务,并且有许多实用方法来考虑同此类概念上等待的任务的影响相等价的影响。

[0089] 其次,尽管初始阶段优选脱机为各个作业选定缓冲器尺寸,但当然也可以联机执行,即,只在作业附加到被执行的作业中之前。缓冲器尺寸计算只是可计算的执行参数计算的一个示例。正如解释的那样,任务自身使用的循环时间是能够在第一阶段计算的另一个参数。作为另一个示例,能够为连续流块处理相同任务的处理单元个数是能在第一阶段确定的另一个执行参数,用以确保实时能力。例如,这可以通过下述实现:对 SDF 图附加任务以便使流块周期地分布到连续处理器上,附加该任务的复本用以处理不同的分布流块,并且附加一个组合任务用以将复本的结果组合到一个组合输出流中。依赖复本的数量可在假定环境中确保与实时吞吐量条件的兼容。

[0090] 此外,可使用到处理单元 10 更精细的分配形式。例如,一个实施例中,初始阶段还可涉及这样的强加约束,即一个作业的任务组应该由同一个处理单元 10 执行。这种情况下,需要更少的虚拟任务 50 来表示等待时间(如果该组中的任务被连续调度),或者表示等待时间的虚拟任务 50 可具有更短的等待时间,表示可能在该组任务之间稍后调度的部分(已知)其它任务的最坏情况执行时间。实际上,在该组内任务前面的虚拟任务 50 的组合等待时间只需要对应一个循环时间  $T$ ,而不是  $n$  个循环时间  $T$ ,在不限定为由同一个处理单元 10 执行的情况下考虑  $n$  个任务时便才需要  $n$  个循环时间  $T$ 。这使得更容易保证实时约束的满足。此外,这种方式可以降低所需的某些缓冲器的尺寸。

[0091] 此外,如果不同作业数据流存在某种形式的同步,则在执行期间不必使用任务跳过。可在 SDF 图中表示出这种同步。

[0092] 此外,尽管就能够处理任何任务的通用处理单元 10 阐述了本发明,但是可替代地,某些处理单元可以是只能执行选定任务的专用单元。正如理解的那样,这并不影响本发明原理,而只是强加一个限制条件,限定任务到处理单元的分配的最终可能性。同样正如理解的那样,尽管为了说明清楚图中省略了通信任务(或者认为合并到任务当中),但实际上可以增加通信任务,以及相应定时和等待关系。

[0093] 此外,尽管就其中每个处理单元 10 都使用一个循环调度方案的实施例(其中任务以固定的顺序执行)对本发明进行阐述,但应该理解只要能够为该方案计算出任务执行之前的最大等待时间就可以使用任何调度方案,所述方案对处理单元 10 所执行任务的最坏执行时间(非指定的)给定一个预定义约束。显然,用来确定任务是否获得足够执行机会的最坏情况执行时间总和的类型依赖于调度的类型。

[0094] 优选地,通过能够在运行时间灵活添加和/或删除作业的处理系统执行作业。这种情况下,可结合关于所需要的缓冲器尺寸和假定的循环时间  $T$  的计算信息来提供作业任务的程序代码。可从另一个处理系统提供所述信息,或者可在执行这些作业的处理系统中本地产生所述信息。然后,可在运行时间采用这个信息添加作业。或者,调度作业执行所要求的信息可永久存储在一个信号处理集成电路中,该集成电路具有执行作业的多个处理单元。甚至可以将它应用到一种集成电路上,所述集成电路被编程为静态执行预定作业组合。后一种情况下,不必在运行时间动态实施处理器的任务分配。

[0095] 因此,依赖于实施,可为执行作业组合的实际设备提供在运行时间确定缓冲器尺寸和将任务分配给处理单元的完整能力,或者只为实际设备提供在运行时间将任务分配给

处理单元的能力,甚或只为其提供预定分配。可利用适当程序对该设备进行编程来实现这些能力,该程序可以是驻留的或者从诸如磁盘的计算机程序产品提供或从表示该程序的互联网信号提供。或者,一种专用硬连线电路可用来支持这些能力。

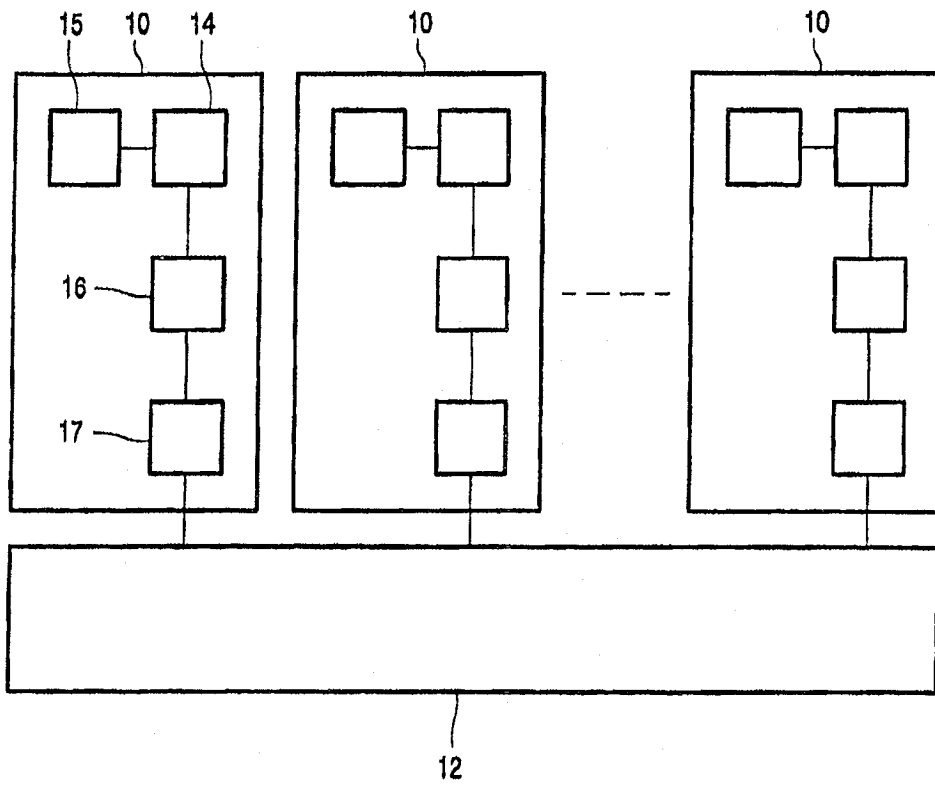


图 1

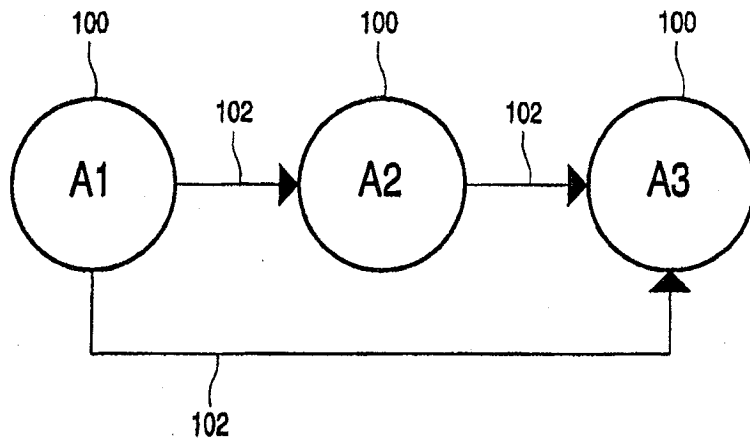


图 1a

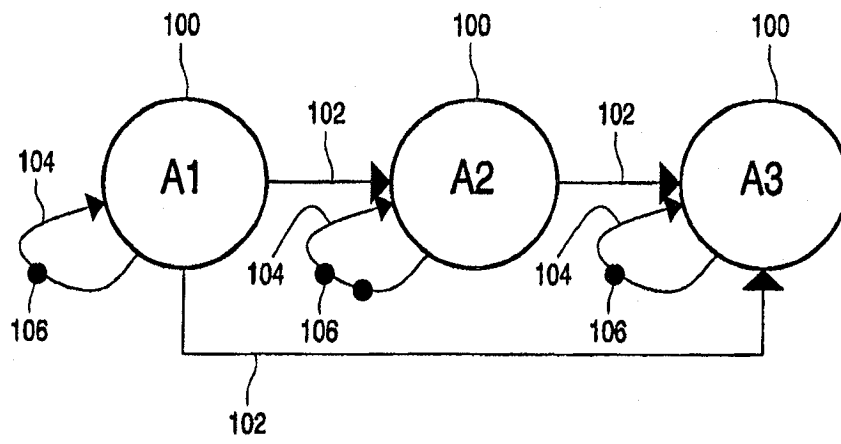


图 1b

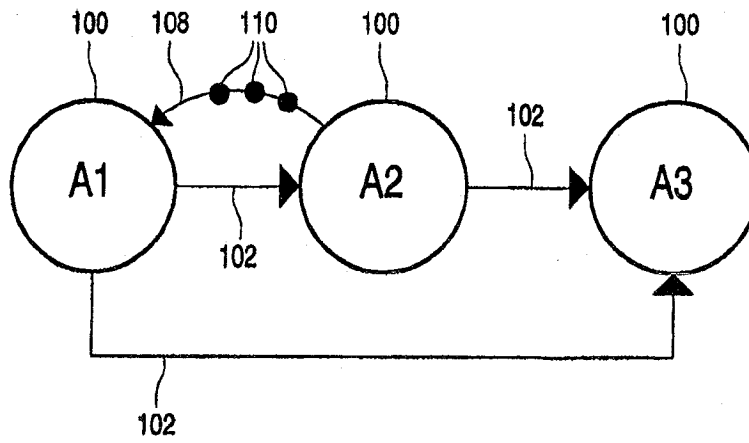


图 1c

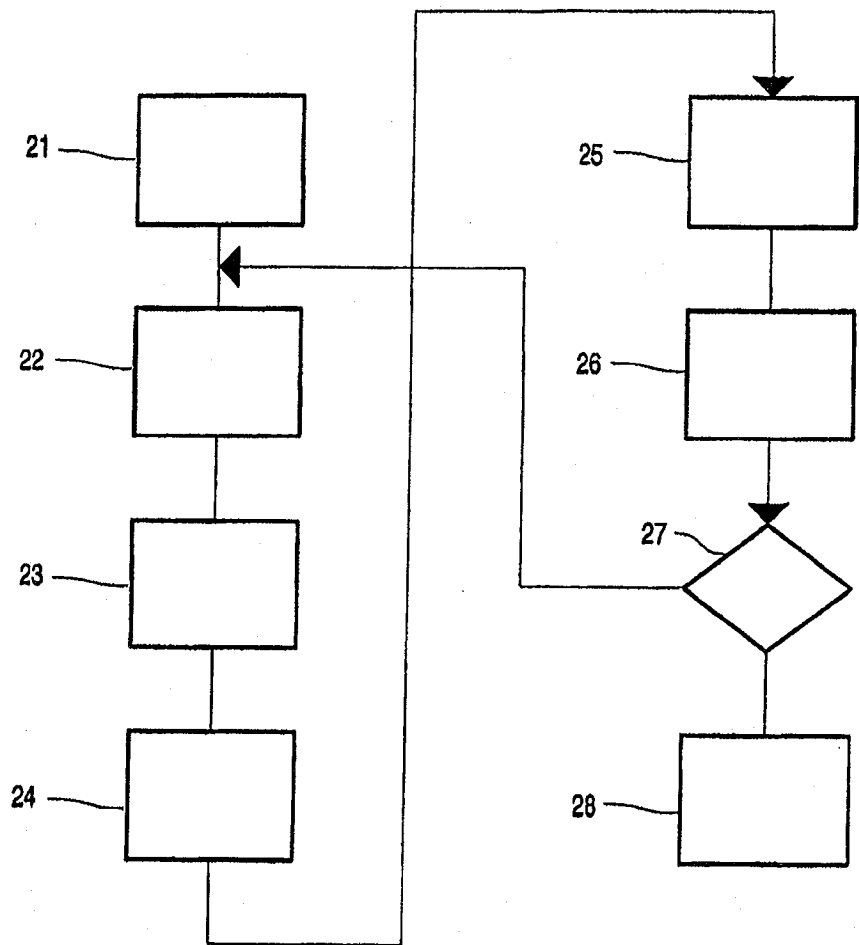


图 2

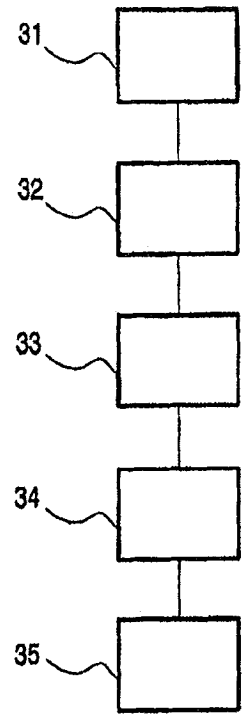


图 3

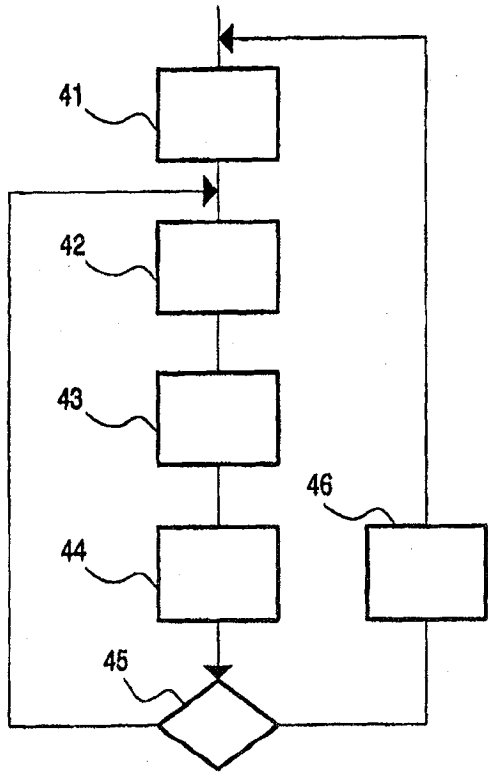


图 4

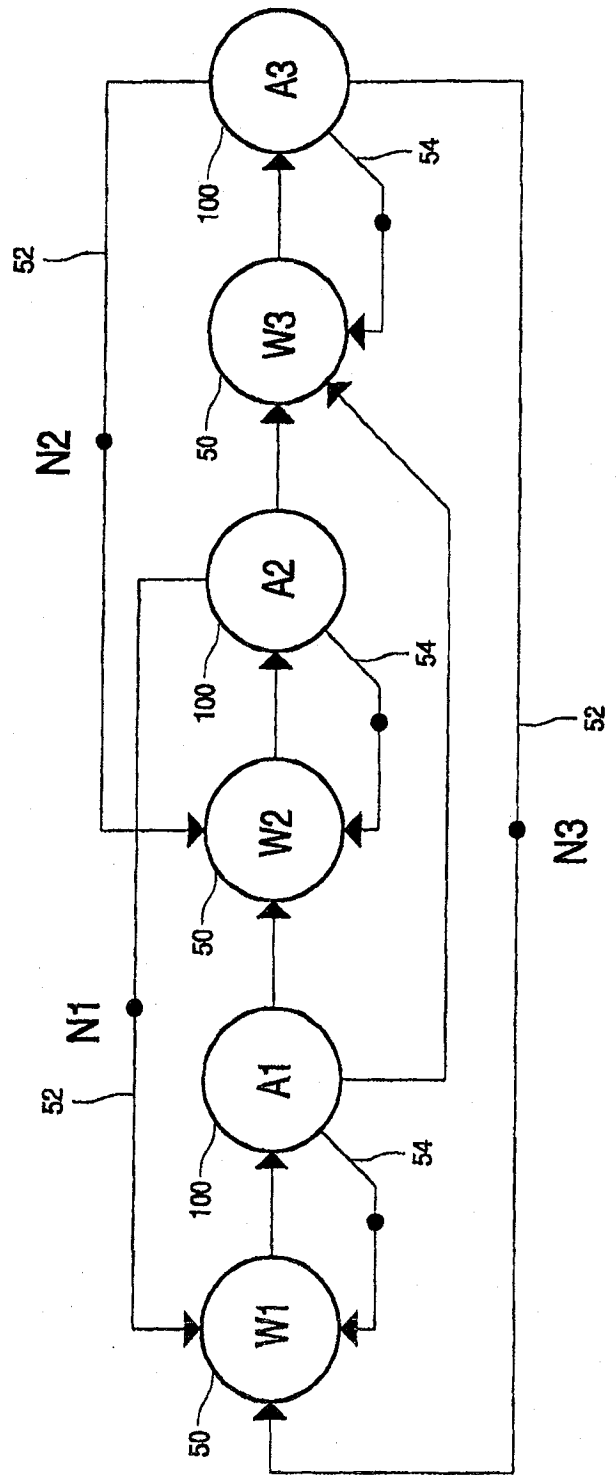


图 5

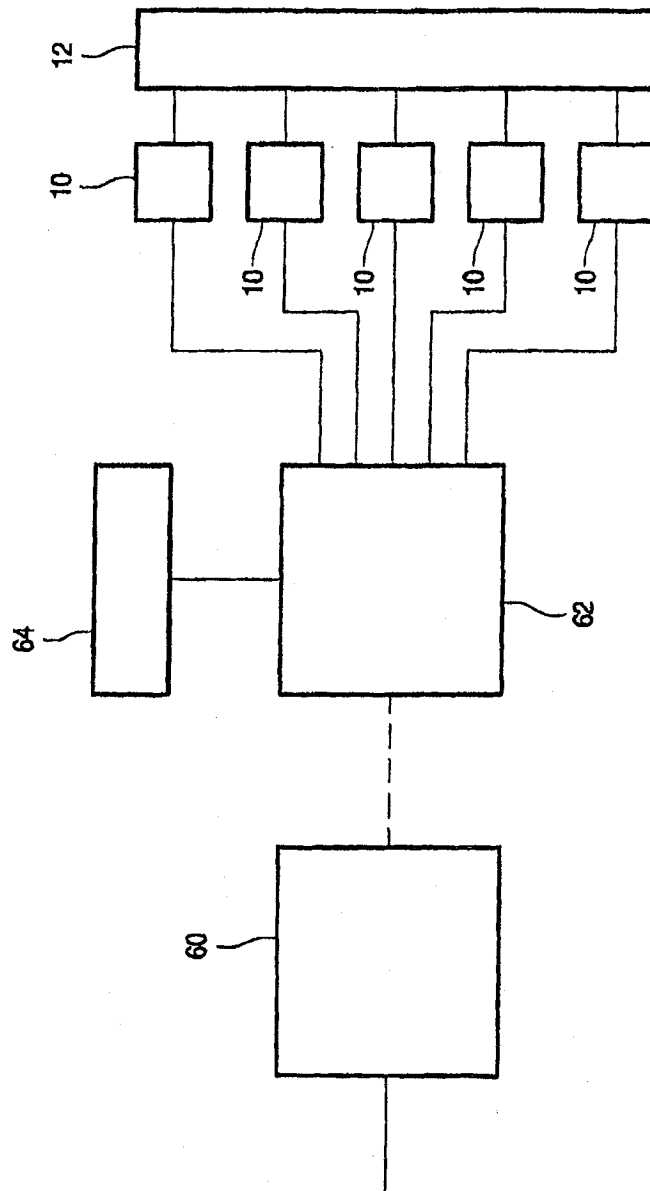


图 6