



(12)发明专利

(10)授权公告号 CN 104067225 B

(45)授权公告日 2017.07.21

(21)申请号 201280067461.X

(22)申请日 2012.12.06

(65)同一申请的已公布的文献号

申请公布号 CN 104067225 A

(43)申请公布日 2014.09.24

(30)优先权数据

61/589,072 2012.01.20 US

13/414,402 2012.03.07 US

(85)PCT国际申请进入国家阶段日

2014.07.18

(86)PCT国际申请的申请数据

PCT/US2012/068203 2012.12.06

(87)PCT国际申请的公布数据

W02013/109353 EN 2013.07.25

(73)专利权人 高通股份有限公司

地址 美国加利福尼亚州

(72)发明人 伟峰·张 张弛红

(74)专利代理机构 北京律盟知识产权代理有限公司 11287

代理人 宋献涛

(51)Int.Cl.

G06F 9/30(2006.01)

(56)对比文件

US 2011/0078424 A1, 2011.03.31, 说明书第[0029-0033]段, 图2-4.

CN 1425982 A, 2003.06.25, 全文.

US 2004/0088689 A1, 2004.05.06, 全文.

审查员 徐菲

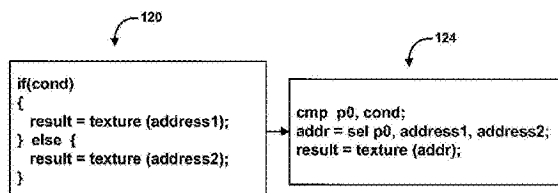
权利要求书3页 说明书13页 附图5页

(54)发明名称

对用于图形处理单元的具有相关联的纹理加载指令的控制流指令的判定

(57)摘要

本发明的各方面涉及一种编译高级软件指令以产生低级软件指令的方法。在一实例中,所述方法包含使用计算装置识别具有一或多个相关联的纹理加载指令的一组高级HL控制流CF指令,其中所述组HL CF指令包括一或多个分支。所述方法还包含使用所述计算装置将所述组所识别的HL CF指令转换为具有判定结构的低级LL指令。所述方法还包含输出具有所述判定结构的所述经转换的LL指令。



1. 一种编译高级软件指令以产生低级软件指令的方法,所述方法包括:

使用计算装置识别具有一或多个相关联的纹理加载指令的一组高级HL控制流CF指令,其中所述一或多个纹理加载指令的执行致使将纹理加载到存储器,其中所述组HL CF指令包括一或多个分支;

判定所述一或多个相关联的纹理加载指令是否关于所述组HL CF指令对称;

在所述一或多个相关联的纹理加载指令对称的情况下,使用所述计算装置在与所述一或多个纹理加载指令相关联的纹理坐标上将所述组所识别的HL CF指令转换为具有判定结构的低级LL指令,其中具有所述判定结构的所述LL指令可在未分支的情况下执行;

在所述一或多个相关联的纹理加载指令不对称的情况下制止转换所述组所识别的HL CF指令以使所述HL CF指令是未经转换的;及

输出具有所述判定结构的经转换的所述LL指令或未经转换的HL CF指令。

2. 根据权利要求1所述的方法,其中所述一或多个纹理加载指令的执行致使将所述纹理传递到图形处理单元的本地存储器。

3. 根据权利要求1所述的方法,其中转换所述HL CF指令包括移除所述一或多个分支以允许所述经转换的低级指令的无条件串行执行。

4. 根据权利要求1所述的方法,其中判定所述一或多个纹理加载指令是否对称包括识别与所述组HL CF指令的每一路径相关联的一或多个纹理加载指令。

5. 根据权利要求4所述的方法,其中判定所述一或多个纹理加载指令是否对称包括识别与所述组HL CF指令的每一路径相关联的相等数目的纹理加载指令。

6. 根据权利要求1所述的方法,其中转换所述组所识别的HL CF指令包括产生一组LL指令,其中所述组LL指令的执行致使基于选择函数的结果而指派地址且从所述所指派的地址加载纹理。

7. 根据权利要求1所述的方法,其中识别所述组HL CF指令包括识别具有条件的IF-ELSE语句,其中如果所述条件是真,那么所述组HL CF指令致使从所述纹理坐标的第一存储器地址加载纹理,且如果所述条件是假,那么所述组HL CF指令致使从所述纹理坐标的第二存储器地址加载所述纹理。

8. 根据权利要求7所述的方法,其中转换所述组HL CF指令包括产生一组LL指令,其中所述组LL指令致使基于所述条件的结果而保留地址且从所述所保留的地址加载所述纹理。

9. 根据权利要求1所述的方法,其中输出所述经转换的LL指令包括将所述经转换的LL指令存储到存储器。

10. 根据权利要求1所述的方法,其中转换所述HL指令包括对所述HL指令进行运行时间编译以产生所述LL指令,且进一步包括在图形处理单元上执行所述所输出的LL指令。

11. 根据权利要求10所述的方法,其中执行所述所输出的LL指令包括调度所述经编译的指令以在所述HL CF指令的条件和所述所输出的LL指令的坐标不相依的指令之前执行。

12. 一种用于编译高级软件指令以产生低级软件指令的设备,所述设备包括一或多个处理单元,所述一或多个处理单元经配置以:

识别具有一或多个相关联的纹理加载指令的一组高级HL控制流CF指令,其中所述一或多个纹理加载指令的执行致使将纹理加载到存储器,其中所述组HL CF指令包括一或多个分支;

判定所述一或多个相关联的纹理加载指令是否关于所述组HL CF指令对称；

在所述一或多个相关联的纹理加载指令对称的情况下，在与所述一或多个纹理加载指令相关联的纹理坐标上将所述组所识别的HL CF指令转换为具有判定结构的低级LL指令，其中具有所述判定结构的所述LL指令可在未分支的情况下执行；

在所述一或多个相关联的纹理加载指令不对称的情况下制止转换所述组所识别的HL CF指令以使所述HL CF指令是未经转换的；及

输出具有所述判定结构的经转换的所述LL指令或未经转换的HL CF指令。

13. 根据权利要求12所述的设备，其进一步包括本地存储器，其中所述一或多个纹理加载指令的执行致使将所述纹理传递到所述设备的本地存储器。

14. 根据权利要求12所述的设备，其中为了转换所述HL CF指令，所述一或多个处理单元经配置以移除所述一或多个分支以允许所述经转换的低级指令的无条件串行执行。

15. 根据权利要求12所述的设备，其中判定所述一或多个纹理加载指令是否对称包括识别与所述组HL CF指令的每一路径相关联的一或多个纹理加载指令。

16. 根据权利要求12所述的设备，其中为了转换所述组所识别的HL CF指令，所述一或多个处理单元经配置以产生一组LL指令，其中所述组LL指令的执行致使基于选择函数的结果而指派地址且从所述所指派的地址加载纹理。

17. 根据权利要求12所述的设备，其中为了识别所述组HL CF指令，所述一或多个处理单元经配置以识别具有条件的IF-ELSE语句，其中如果所述条件是真，那么所述组HL CF指令致使从所述纹理坐标的第一存储器地址加载纹理，且如果所述条件是假，那么所述组HL CF指令致使从所述纹理坐标的第二存储器地址加载所述纹理。

18. 根据权利要求17所述的设备，其中为了转换所述组HL CF指令，所述一或多个处理单元经配置以产生一组LL指令，其中所述组LL指令致使基于所述条件的结果而保留地址且从所述所保留的地址加载所述纹理。

19. 根据权利要求12所述的设备，其中为了输出所述经转换的LL指令，所述一或多个处理单元经配置以将所述经转换的LL指令存储到存储器。

20. 根据权利要求12所述的设备，其中为了转换所述HL指令，所述一或多个处理单元经配置以对所述HL指令进行运行时间编译以产生所述LL指令，且进一步包括执行所述所输出的LL指令。

21. 根据权利要求20所述的设备，其中为了执行所述所输出的LL指令，所述一或多个处理单元经配置以调度所述经编译的指令以在所述HL CF指令的条件和所述所输出的LL指令的坐标不相依的指令之前执行。

22. 一种用于编译高级软件指令以产生低级软件指令的设备，所述设备包括：

用于识别具有一或多个相关联的纹理加载指令的一组高级HL控制流CF指令的装置，其中所述一或多个纹理加载指令的执行致使将纹理加载到存储器，其中所述组HL CF指令包括一或多个分支；

用于判定所述一或多个相关联的纹理加载指令是否关于所述组HL CF指令对称的装置；

用于在所述一或多个相关联的纹理加载指令对称的情况下，在与所述一或多个纹理加载指令相关联的纹理坐标上将所述组所识别的HL CF指令转换为具有判定结构的低级LL指

令的装置,其中具有所述判定结构的所述LL指令可在未分支的情况下执行;

用于在所述一或多个相关联的纹理加载指令不对称的情况下制止转换所述组所识别的HL CF指令以使所述HL CF指令是未经转换的装置;及

用于输出具有所述判定结构的经转换的所述LL指令或未经转换的HL CF指令的装置。

23.根据权利要求22所述的设备,其中所述一或多个纹理加载指令的执行致使将所述纹理传递到图形处理单元的本地存储器。

24.根据权利要求22所述的设备,其中转换所述HL CF指令包括移除所述一或多个分支以允许所述经转换的低级指令的无条件串行执行。

25.根据权利要求22所述的设备,其中判定所述一或多个纹理加载指令是否对称包括识别与所述组HL CF指令的每一路径相关联的一或多个纹理加载指令。

26.根据权利要求22所述的设备,其中转换所述组所识别的HL CF指令包括产生一组LL指令,其中所述组LL指令的执行致使基于选择函数的结果而指派地址且从所述所指派的地址加载纹理。

27.根据权利要求22所述的设备,其中识别所述组HL CF指令包括识别具有条件的IF-ELSE语句,其中如果所述条件是真,那么所述组HL CF指令致使从所述纹理坐标的第一存储器地址加载纹理,且如果所述条件是假,那么所述组HL CF指令致使从所述纹理坐标的第二存储器地址加载所述纹理。

28.根据权利要求27所述的设备,其中转换所述组HL CF指令包括产生一组LL指令,其中所述组LL指令致使基于所述条件的结果而保留地址且从所述所保留的地址加载所述纹理。

29.根据权利要求22所述的设备,其中输出所述经转换的LL指令包括将所述经转换的LL指令存储到存储器。

30.根据权利要求22所述的设备,其中转换所述HL指令包括对所述HL指令进行运行时间编译以产生所述LL指令,且进一步包括在图形处理单元上执行所述所输出的LL指令。

31.根据权利要求30所述的设备,其中执行所述所输出的LL指令包括调度所述经编译的指令以在所述HL CF指令的条件和所述所输出的LL指令的坐标不相依的指令之前执行。

对用于图形处理单元的具有相关联的纹理加载指令的控制流指令的判定

[0001] 本申请案主张2012年1月20日申请的第61/589,072号美国临时专利申请案的优先权,所述申请案的全部内容以全文引用的方式并入本文中。

技术领域

[0002] 本发明涉及计算装置,且更特定来说,涉及图形处理装置。

背景技术

[0003] 编译器是从例如根据各种所谓的高级计算机编程语言(例如,C、C++、Java、Basic等)界定的高级软件指令产生例如由各种机器或汇编计算机编程语言界定的低级软件指令的计算机程序。计算机程序员通常使用高级软件指令界定计算机程序且调用编译器来产生低级软件指令。所述低级软件指令对应于可由支持低级软件指令的执行的任何给定计算装置执行的高级软件指令。以此方式,所述编译器编译所述高级软件指令以产生低级软件指令,使得任何给定计算装置可执行由计算机程序员使用根据高级编程语言界定的软件指令而界定的计算机程序。

发明内容

[0004] 一股来说,本发明描述用于有效地编译包含一或多个纹理加载指令的一组控制流指令的技术。短语“控制流”一股是指根据高级编程语言而界定的一组指令,所述高级编程语言针对于控制基于一些有条件语句而形成计算机程序的高级软件指令的执行流。也就是说,例如,一组控制流指令的执行可导致关于应执行高级软件指令的两个或更多分支中的何者作出选择。纹理加载指令一股是指在被执行时致使将纹理加载到图形处理单元(GPU)的本地存储器的指令。举例来说,执行纹理加载指令可致使将纹理(例如,图像数据)从在GPU外部的存储器传递到GPU的本地存储器。

[0005] 本发明的技术包含将包含纹理加载指令的一组控制流(CF)指令转换为具有判定结构的一或多个指令。具有判定结构的指令可被串行地执行(例如,不具有分支)。通常,将CF语句转换为具有判定结构的指令可包含移除分支及执行每一分支的所有指令。根据本发明的技术,然而,可以一种方式将具有一或多个相关联的纹理加载指令的一组CF指令转换为判定结构,使得不需要执行与每一分支相关联的所有纹理加载指令。以此方式,所述技术可提供对包含纹理加载指令的CF语句的有效转换。

[0006] 在一实例中,本发明的各方面针对于一种编译高级软件指令以产生低级软件指令的方法。所述方法包含使用计算装置识别具有一或多个相关联的纹理加载指令的一组高级(HL)控制流(CF)指令,其中所述组HL CF指令包括一或多个分支;使用所述计算装置将所述组所识别的HL CF指令转换为具有判定结构的低级(LL)指令;及输出具有所述判定结构的所述经转换的(LL)指令。

[0007] 在另一实例中,本发明的各方面针对于一种用于编译高级软件指令以产生低级软

件指令的设备,所述设备包括一或多个处理单元,所述一或多个处理单元经配置以:识别具有一或多个相关联的纹理加载指令的一组高级(HL)控制流(CF)指令,其中所述组HL CF指令包括一或多个分支;将所述组所识别的HL CF指令转换为具有判定结构的低级(LL)指令;及输出具有所述判定结构的所述经转换的(LL)指令。

[0008] 在另一实例中,本发明的各方面针对于一种用于编译高级软件指令以产生低级软件指令的计算机程序产品,所述计算机程序产品存储指令,所述指令在被执行时致使一或多个处理器:识别具有一或多个相关联的纹理加载指令的一组高级(HL)控制流(CF)指令,其中所述组HL CF指令包括一或多个分支;将所述组所识别的HL CF指令转换为具有判定结构的低级(LL)指令;及输出具有所述判定结构的所述经转换的(LL)指令。

[0009] 在另一实例中,本发明的各方面针对于一种用于编译高级软件指令以产生低级软件指令的设备,所述设备包括:用于识别具有一或多个相关联的纹理加载指令的一组高级(HL)控制流(CF)指令的装置,其中所述组HL CF指令包括一或多个分支;用于将所述组所识别的HL CF指令转换为具有判定结构的低级(LL)指令的装置;及用于输出具有所述判定结构的所述经转换的(LL)指令的装置。

[0010] 一或多个实例的细节陈述于附图及以下描述中。其它特征、目标及优势将从描述及附图和从权利要求书中显而易见。

附图说明

[0011] 图1是说明可实施本发明中所描述的技术的开发系统的框图。

[0012] 图2是说明可实施本发明中所描述的技术的计算装置的框图。

[0013] 图3说明实例性伪码,其包含已在没有判定转换的情况下以及在具有判定转换的情况下编译的控制流指令的实例性伪码。

[0014] 图4说明实例性伪码,其包含具有纹理加载的控制流指令,所述指令以传统的判定转换进行编译。

[0015] 图5说明实例性伪码,其包含已根据本发明的技术进行转换的具有纹理加载的控制流指令。

[0016] 图6是说明在实施本发明中所描述的技术的各方面中的编译器的实例性操作的流程图。

具体实施方式

[0017] 一般来说,本发明描述用于有效地编译包含一或多个纹理加载指令的控制流语句的技术。短语“控制流”一股是指根据高级(HL)编程语言而界定的一组指令,所述高级编程语言针对于控制基于一些有条件声明而形成计算机程序的HL软件指令的执行流。也就是说,控制流(CF)语句的执行可导致关于应执行HL软件指令的两个或更多分支中的何者作出选择。在一些例子中,CF语句还可被称作“流控制”语句,或“有条件流控制”语句。

[0018] 举例来说,一组HL CF语句一股涉及“if”指令继之以有条件语句的使用。此有条件语句通常使用布尔操作符被界定为布尔语句。一个实例性有条件语句可涉及布尔比较以确定变量的当前值是否大于给定值,其可表达为“ $x > 10$ ”,其中在此语句中,所述变量被表示为 x ,其中大于操作符被界定为符号“ $>$ ”。此语句是布尔值,原因在于其返回“真”(其通常被

界定为一)或“假”(其通常被界定为零)的布尔值。在此“if”指令之后的是一或多个额外指令。如果有条件语句是真,那么执行额外指令。如果有条件语句是假,那么跳过或不执行额外指令,且执行流在额外指令之后重新开始。

[0019] 其它类型的HL CF指令集包含使用“if”指令继之以“else”指令而界定的指令集(通常被称为“if-else”CF指令)、使用操作符“:?”而界定的指令集,以及使用多个“if”语句而界定的指令集(通常被称作“if-if”CFC指令)。

[0020] 纹理加载指令一股是指在被执行时致使将纹理加载到本地存储器(例如,计算装置(例如图形处理单元(GPU))的本地存储器)的指令。“纹理”一股是指含有全部具有相同格式的一或多个图像的对象(例如,具有特定大小和特定格式的某一维度的像素的阵列)。也就是说,例如,纹理可为给予对象纹理外观的像素色彩的位图。在GPU中,执行纹理加载指令可致使将纹理(例如,图像数据)从外部存储器传递到GPU的本地存储器。所述外部存储器可相对于GPU是芯片外的,且可经由系统总线进行存取。在一些例子中,如下文更详细地描述,相对长的等待时间可与此数据传递相关联。举例来说,将数据从外部存储器传递到本地存储器可消耗相对大量的时钟循环。

[0021] 本发明的技术包含将包含纹理加载指令的控制流(CF)语句转换为具有判定结构的一或多个指令。举例来说,本发明的技术可包含将包含纹理加载指令的“IF”CF语句转换为具有判定结构的一或多个指令。判定执行支持提供从指令流消除分支的方式。举例来说,判定执行可指代基于布尔源操作数的值(被称为判定)对指令的有条件执行。判定执行支持允许编译器将有条件分支转换为判定的界定指令,且沿着每一分支的替代性路径将指令转换为判定指令。也就是说,可获取判定指令,而不管其判定值如何。正常地执行其判定为真的指令。相反,取消其判定为假的指令,且因此防止修改GPU的状态。因此,具有判定结构的指令可被串行地执行(例如,不具有分支)。

[0022] 将程序的CF指令转换为具有判定结构的指令可增加最终由处理装置执行的指令的数目。举例来说,通过预测,处理装置有效地评估分支的两侧且丢弃结果中的一者。在一些例子中,评估分支的两侧可相对成本较高,尤其在路径包含较大和/或复杂语句的情况下。

[0023] 然而,一些处理架构可比CF指令更有效地执行判定指令。举例来说,归因于GPU(有时被称为“流处理器”)的并行性质,某些GPU架构可比CF指令更有效地执行判定指令。举例来说,CF指令的分支可强加控制相依性,其约束在每一时钟循环可用于执行的独立指令的数目。零位,虽然某些处理单元可执行既定减少与所述相依性相关联的等待时间的推测性执行,但分支的错误预测也可导致性能损失。

[0024] 在其中处理单元(例如,GPU)包含判定支持的例子中,编译器可执行全判定转换或部分判定转换。全判定支持允许将所有CF指令转换为判定结构。相比而言,部分判定支持限制可被转换为判定结构的指令的类型。举例来说,一些GPU仅允许部分判定结构,且在此些例子中,对应的GPU编译器可能限制被转换为有条件移动指令(例如,MOVC)和/或选择指令(例如,SELECT)的所述组指令。

[0025] 一些编译器(例如,一些GPU编译器)可选择不执行部分判定转换,因为转换某些指令的成本可能超过判定结构的益处。举例来说,如上所述,纹理加载指令可致使将纹理(例如,图像数据)从外部存储器传递到GPU的本地存储器。在一些例子中,相对长的等待时间可

与此数据传递相关联。归因于与纹理加载指令相关联的相对长的等待时间,编译器可不将包含纹理加载指令的一组CF指令转换为判定结构。举例来说,执行每一分支的纹理加载指令可通常超过转换的益处。

[0026] 本发明的技术包含以一种方式将具有一或多个相关联的纹理加载指令的一组CF指令转换为判定结构,使得避免执行与所述组CF指令的每一分支相关联的每个纹理加载指令。以此方式,实施本发明的技术的GPU编译器可实施适合于GPU架构的判定结构,同时使纹理加载指令中固有的等待时间最小化。

[0027] 图1是说明可实施本发明所描述的技术的开发系统10的框图,本发明的技术针对于将具有一或多个相关联的纹理加载指令的控制流(CF)指令转换为具有判定结构的指令(判定指令)。在图1的实例中,开发系统10包含计算装置12。计算装置12可包括桌上型计算机、膝上型计算机(包含所谓的“笔记本”计算机)、工作站、板或平板计算机、个人数字助理(PDA)、移动或蜂窝式电话(包含所谓的“智能电话”)、数字媒体播放器、游戏装置,或用户(例如,软件开发者13)可与其交互以界定高级(HL)代码且随后编译HL代码以产生低级(LL)代码的任何其它装置。在本发明中,术语“代码”一般是指界定计算机程序、软件或其它可执行文件的一组一或多个软件指令。

[0028] 计算装置12包含控制单元14。控制单元14可包括例如可编程微处理器等一或多个处理器(图1的实例中未展示),所述一或多个处理器执行软件指令,例如用于界定软件或计算机程序的软件指令,存储到计算机可读存储媒体(再次,图1的实例中未展示)的软件指令,所述计算机可读存储媒体例如为存储装置(例如,磁性硬盘驱动器、固态驱动器或光学驱动器),或存储器(例如,快闪存储器、随机存取存储器或RAM)或存储用以致使可编程处理器执行本文中所描述的技术的指令的任何其它类型的易失性或非易失性存储器。或者,控制单元14可包括用于执行本文中所描述的技术的专用硬件,例如一或多个集成电路、一或多个专用集成电路(ASIC)、一或多个专用特殊处理器(ASSP)、一或多个现场可编程门阵列(FPGA),或可编程处理器、专用硬件等的前述实例中的一或多者的任何组合。

[0029] 控制单元14执行或以其它方式实施用户接口(UI)模块16、软件开发模块18和编译器20。也就是说,在一些实例中,UI模块16、软件开发模块18和/或编译器20可包括由控制单元14执行的软件。UI模块16表示呈现例如开发者13等用户可与其介接以与软件开发模块18和编译器20进行交互的用户接口的模块。UI模块16可表示开发者13可与其交互以与模块18和20进行介接的任何类型的用户接口,例如命令线接口(CLI)和/或图形用户接口(GUI)。

[0030] 软件开发模块18表示在HL编程语言方面促进软件的开发模块。通常,软件开发模块18表示经由UI模块16到开发者13的一或多个用户接口,借此开发者13与这些用户接口交互,从而以高级(HL)代码22的形式界定软件。如下文更详细地描述,HL代码22可包含控制流(CF)指令。再次地,如本发明中所使用的术语“代码”是指界定计算机程序、软件或其它可执行文件的一组一或多个软件指令。HL代码22通常表示在通常被称为HL编程语言的东西中界定的指令。HL编程语言一般是指具有来自计算机的基础细节的某一抽象的编程语言,例如处理器的存储器存取模型以及处理器内的范围的管理。

[0031] HL编程语言一般实现比低级(LL)编程语言更高级的抽象,所述低级编程语言是一股指代机器编程语言和汇编编程语言的术语。HL编程语言的实例包含C编程语言、所谓的C++编程语言、Java编程语言、视觉基础(VB)编程语言,及基础编程语言。在一些实例中,可出

于任何特定目的对HL编程语言进行裁剪,例如产生HL代码以用于执行图形处理单元(GPU)。此类HL编程语言的实例包含开放图形库(GL)编程语言、开放GL嵌入式系统(ES)编程语言,或高级着色器语言(HLSL)。许多HL编程语言是面向对象的,原因在于其实现对能够存储数据且进行开放以由算法操纵的对象(其可包括(例如)指令、接口及数据结构)的界定,以便抽象地解决多种问题,而不必考虑计算装置的基础架构。

[0032] 编译器20表示将根据HL编程语言而界定的HL指令精简为LL编程语言的LL指令的模块。在编译之后,这些LL指令能够由特定类型的处理器或其它类型的硬件(例如,FPGA、ASIC等)执行。LL编程语言在以下意义上被视为低级:它们从处理器或其它类型的硬件的指令集架构提供极少的抽象,或较低级的抽象。LL语言一股是指汇编和/或机器语言。汇编语言是比机器语言略微更高的LL语言,但一股汇编语言可在不使用编译器或其它转译模块的情况下被转换为机器语言。机器语言表示界定与基础硬件(例如处理器)在本地执行的指令(例如,x86机器代码(其中x86指代由英特尔公司开发的x86处理器的指令集架构))类似(如果不相同的话)的指令的任何语言。

[0033] 编译器20实际上将根据HL编程语言而界定的HL指令转译为由基础硬件支持的LL指令,且移除与HL编程语言相关联的抽象,使得根据这些HL编程语言而界定的软件能够由实际的基础硬件更直接地执行。通常,例如编译器20等编译器能够将与单一HL编程语言相关联的HL指令精简为例如LL代码24等LL代码,包括根据一或多种LL编程语言而界定的指令,但一些编译器可将与一种以上HL编程语言相关联的HL指令精简为根据一或多种LL编程语言而界定的LL指令。

[0034] 虽然出于阐释的目的而将软件开发模块18和编译器20展示为图1的实例中的单独模块,但在一些实例中,软件开发模块18和编译器20可组合在单一模块中,所述单一模块通常被称为集成开发环境(IDE)。本发明的技术不应在此方面受限于图1的实例中所展示的单模块18和20,而是可适用于在其中组合这些模块的例子,例如在IDE中。通过IDE,开发者可使用HL指令界定软件且通过采用编译器将HL指令转译为LL指令而产生包括能够由处理器(例如,GPU)或其它类型的硬件执行的LL指令的可执行文件两者。通常,IDE提供综合GUI,开发者可与其交互以界定且调试使用HL指令而界定的软件、将HL指令编译为LL指令且对LL指令的执行模型化,以便在LL指令由存在于装置内或存在于另一装置(例如,蜂窝式电话)内的硬件执行时观察所述LL指令的执行将如何表现。

[0035] 举例来说,开放GL ES编程语言是开放GL(其经开发以供桌上型和膝上型计算机执行)的一版本,其经调适以用于不在例如桌上型和膝上型计算机等个人计算机上而是在移动装置(例如,蜂窝式电话(包含所谓的智能电话))、笔记本计算机、平板计算机、板式计算机、数字媒体播放器、游戏装置和其它便携式装置上执行。开发GL以及因此开发GLES提供综合架构,借此界定、操纵且再现二维(2D)和三维(3D)图形两者。在IDE内将这些移动装置(其可具有拥有与个人计算机中常见的指令集架构极大不同的指令集架构的处理器)模型化的能力已进一步增加了IDE作为开发者寻求开发移动装置的软件所选择的开发环境的合意性。虽然图1的实例中未展示,但控制单元14还可执行或实施模型化器模块,所述模型化器模块能够通过常常未在本地包含在计算装置12内的硬件(例如,移动处理器等)来将LL软件指令的执行模型化。

[0036] 在任何情况下,例如编译器20等编译器的一个功能可涉及将HL代码22(例如,根据

HL编程语言而界定)的控制流(CF)指令转译为具有拥有判定结构的指令的LL代码24(例如,根据LL编程语言而界定)。CF指令是指可借以控制处理器对指令的执行流的任何指令。举例来说,许多HL编程语言指定“if”指令,其语法通常需要在此“if”指令的调用之后界定有条件语句。此有条件语句通常使用布尔操作符被界定为布尔语句。

[0037] 一个实例性有条件语句可涉及布尔比较以确定变量的当前值是否大于给定值,其可表达为“ $x > 10$ ”,其中在此语句中,所述变量被表示为‘x’,其中大于布尔操作符被界定为符号‘>’。此语句是布尔值,原因在于其返回“真”(其通常被界定为一)或“假”(其通常被界定为零)的布尔值。在此“if”指令之后的是一或多个额外指令,且如果有条件语句是真,那么执行额外指令如果有条件语句是假,那么跳过或不执行额外指令,且执行流在额外指令之后重新开始。在此意义上,在评估有条件(常常是布尔)语句之后,“if”指令即刻调节且进而控制额外指令的执行。出于此原因,“if”指令常常被称为CF指令。

[0038] 其它类型的HL CF指令集包含使用“if”指令继之以“else”指令而界定的指令集(通常被称为“if-else”或“if-then-else”CF指令)、使用操作符“:?”而界定的指令集,以及使用多个“if”语句而界定的指令集(通常被称作“if-if”CF指令)。如果“if-else”指令集、“if”指令与上文论述的相同,但是执行的流或控制被“else”语句修改,使得当在“if”之后的有条件语句是假时,执行在“else”指令之后的第二组额外指令。仅在“if”指令之后的有条件语句是假的情况下才执行此第二组额外指令,进而提供对指令的执行的进一步层级的控制。

[0039] “:?,”指令一股是指模仿“if-else”指令的三元操作符。此指令还通常被称为“?:”指令。通常,“?”指令或操作符的前面是有条件且常常是布尔语句,且紧跟着将在所述有条件语句是真的情况下指派给变量的值。此“真”值于是后面是“:”指令或操作符,其后面又是将在所述有条件语句是假的情况下指派给变量的值。“if-if”指令一股是指在形式上与上文界定的“if”语句相同或至少类似的“if”语句的序列。可以类似于“if-else”指令集的方式采用“if-if”指令集,例如在第一“if”指令的后面是某一有条件语句且在所述第一指令之后的一组“if”指令具有针对第一“if”指令而界定的有条件语句的相反情况时。

[0040] 在一些实例中,编译器20可包含部分判定支持。也就是说,例如,编译器20可将具有某些类型的CF指令的HL代码22转换为具有判定结构的指令,同时制止转换其它类型的CF指令。在一实例中,编译器20可将HL代码22的移动指令(例如,MOVC)和/或选择指令(例如,SELECT)转换为具有判定结构的指令,同时制止转换其它有条件、循环或CF指令。一股来说,将HL代码22的一组CF指令转换为具有判定结构的指令可包含界定评估所述组CF指令的每一分支且丢弃结果中的一者的一组指令。

[0041] 在其中编译器20被配置为所谓的部分判定编译器的实例中,编译器20可归因于与纹理加载相关联的等待时间而不将具有纹理加载指令的一组CF指令转换为具有判定结构的指令。举例来说,当执行纹理加载指令时,GPU可将纹理(例如,图像数据)从外部存储器加载到GPU的本地存储器。在一些例子中,相对长的等待时间可与此数据传递相关联。也就是说,将数据从外部存储器传递到本地存储器可消耗相对大量的时钟循环。转换具有纹理加载指令的一组此类CF指令可涉及执行与所述组CF指令相关联的所有纹理加载指令且选择适当的经加载纹理。执行与一组CF指令相关联的所有纹理加载指令所需的时间可超过通过转换为具有判定结构的指令而获得的效率(例如,硬件效率)。

[0042] 本发明的技术包含将包含具有一或多个相关联的纹理加载指令的一组CF指令的HL代码22转译为根据LL编程语言而界定的LL代码24。根据本发明的技术的转换移除控制流且可以一种方式执行,使得避免执行与所述组CF指令的每一分支相关联的所有纹理加载指令。

[0043] 根据本发明的一些方面,可通过在纹理坐标上产生具有判定结构的指令而将具有纹理加载指令的一组CF指令转换为具有判定结构的指令。坐标可包含(例如)存储器中的特定地址。因此,具有判定结构的指令的结果可指向特定存储器地址。可随后使用指令从特定存储器地址加载纹理。以此方式,可从根据判定指令而界定的地址加载单一纹理。

[0044] 在用于说明的目的的非限制性实例中,HL代码22可包含具有与if-else语句的每一路径相关联的纹理加载指令的if-else CF语句。也就是说,例如,如果if-else语句的“if”路径是真,那么从第一存储器地址加载纹理。相反,如果“if”路径是假,那么执行if-else语句的“else”路径,且从第二存储器地址加载纹理。此if-else语句可表示为:if (condition) {x=texture (coordinate1);} else {x=texture (coordinate2);}。

[0045] 在此实例中,根据本发明的各方面,编译器20可将if-else CF语句转换为具有拥有判定结构的指令的LL代码24。也就是说,编译器20可在纹理坐标上产生判定结构。举例来说,相对于上文所描述的if-else语句,经转换的if-else语句可表示为:{new_coordinate =condition?coordinate1:coordinate2;x=texture (new_coordinate);}。通过用在纹理坐标上执行的三元操作符来取代if-else语句,可在不加载纹理的情况下执行所述语句。另外,if-else控制流被移除且可在无分支的情况下执行指令。虽然关于单一if-else语句来描述所述实例,但应理解,本发明的技术可关于其它组CF指令而一般化,包含嵌套的控制流结构。

[0046] 以此方式,由控制单元14执行的编译器20可实施本发明的技术以将具有纹理加载指令的一组CF指令转换为具有判定结构的指令。转换具有纹理加载指令的所述组CF指令可增加一些处理架构(例如,GPU)的效率。举例来说,具有判定结构的经转换的指令可被串行地执行(不具有反馈环路)。另外,通过限制被执行的纹理加载指令的数目,可最小化与将数据从外部存储器传递到GPU的芯片上存储器相关联的等待时间。

[0047] 在一些实例中,编译器20可将本发明的技术应用于具有对称纹理加载指令的一组CF指令。对称纹理加载指令一股是指具有相同的或近似相同的数目的与一组CF指令的每一路径相关联的纹理加载指令的所述组CF指令。在为了说明的目的的实例中,if-else语句具有与所述语句的“if”部分相关联的第一路径和与所述语句的“else”部分相关联的第二路径。根据本发明的各方面,编译器20可确定与第一路径相关联的纹理加载指令的数目是否近似等于与第二路径相关联的纹理加载指令的数目。

[0048] 如果纹理加载指令近似对称,那么编译器20可执行本发明的技术以将所述组CF指令转换为具有判定结构的指令。如果所述纹理加载指令不对称(例如,仅一个路径包含纹理加载指令),那么编译器20可制止转换CF指令。编译器20可不转换具有非对称纹理加载指令的所述组CF指令,因为潜在存在较少的转换所述指令的动机(例如,所述CF指令可保持完整)。举例来说,如果一组CF指令具有包含纹理加载指令的一个路径,和不包含纹理加载指令的另一路径,那么存在以下可能性:将不执行所述纹理加载指令且将不引发与所述纹理加载指令相关联的等待时间(例如,在不具有所述纹理加载指令的路径是真的情况下)。归

因于此可能性,编译器20可制止花费与转换所述组CF指令相关联的计算和/或时间成本。

[0049] 虽然关于与一组CF指令的每一路径相关联的单一纹理加载指令来描述所述实例,但本发明的技术不以此方式受限。也就是说,编译器20可针对一组CF指令的一个以上分支确定纹理加载指令是否近似对称。另外,纹理加载指令不需要在本质上是明确对称的。举例来说,尽管一个路径比另一路径具有更多或更少的相关联的纹理加载指令(例如,一个路径具有两个相关联的纹理加载指令,且另一路径具有一个或三个纹理加载指令),但只要这些加载的纹理坐标可适当地判定,编译器20便可执行本发明的技术来用于转换CF指令。

[0050] 图2是说明可实施本发明中所描述的技术的另一计算装置70的框图。在图2的实例中,计算装置70表示移动装置,例如以下各者的任何组合:蜂窝式电话(包含所谓的“智能电话”)、膝上型计算机和所谓的“笔记本”,或个人数字助理(PDA)、数字媒体播放器、游戏装置、地理定位系统(GPS)单元、嵌入式系统、便携式媒体系统,或通常根据OpenGL ES规格来实施或支持OpenGL ES的任何其它类型的计算装置。然而,应理解,本发明的技术不限于移动装置。

[0051] 在图2的实例中,计算装置70包含中央处理单元(CPU)72、图形处理单元(GPU)74、存储单元76、显示单元78、显示缓冲器单元80,及用户接口单元84。在一个实例中,图1的实例中所示的控制单元14可包括单元72到76及80。虽然CPU72和GPU74被说明为图2的实例中的单独单元,但CPU72和GPU74可被集成为单一单元,例如在GPU被集成到CPU中时的情况下。CPU72表示能够执行机器或LL指令的一或多个处理器。在一些实例中,CPU72可包含存储器75A。

[0052] GPU74表示用于执行图形操作的一或多个专用处理器。也就是说,例如,GPU74可为具有用于再现图形并执行GPU应用的固定功能和可编程组件的专业硬件单元。GPU74还可包含DSP、通用微处理器、ASIC、FPGA,或其它等效的集成的或离散的逻辑电路。GPU74还可包含其它组件,例如专用GPU存储器75B。

[0053] 如上所述,CPU72和GPU74中的每一者可包含存储器75A、75B(“存储器75”)。存储器75可表示用于执行机器或对象代码的芯片上存储装置或存储器。存储器75可各自包括能够存储固定数目的数字位的硬件存储器寄存器。CPU72和GPU74可分别能够比从存储单元76读取值或将值写入到所述存储单元更快速地从本地存储器75A、75B读取值或将值写入到所述存储器,可例如在系统总线上存取所述存储单元。在一些实例中,存储器75A可为具有CPU72的芯片上存储器且存储器75B可为具有GPU74的芯片上存储器。存储单元76可包括一或多个计算机可读存储媒体。存储单元76的实例包含(但不限于)随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、CD-ROM或其它光盘存储装置、磁盘存储装置,或其它磁性存储装置、快闪存储器,或可用于存储呈指令或数据结构的形式所要的程序代码且可由计算机或处理器存取的任何其它媒体。

[0054] 在一些实例性实施方案中,存储单元76可包含致使CPU72和/或GPU74执行归于本发明中的CPU72和GPU74的功能的指令。在一些实例中,存储单元76可被视为非暂时性存储媒体。术语“非暂时性”可指示所述存储媒体未体现于载波或所传播信号中。然而,术语“非暂时性”应被解释为是指存储单元76是非可移动的。作为一个实例,存储单元76可被从计算装置70移除,且移动到另一装置。作为另一实例,大体上类似于存储单元76的存储单元可插入到计算装置70中。在某些实例中,非暂时性存储媒体可存储可随着时间改变的数据(例

如,在RAM中)。

[0055] 显示单元78表示能够显示视频数据、图像、文本或任何其它类型的数据以供观看者消耗的单元。显示单元78可包含液晶显示器(LCD)、发光二极管(LED)显示器、有机LED(OLED)、有源矩阵OLED(AMOLED)显示器等。显示缓冲器单元80表示专用于为显示单元78存储用于例如相片或视频帧等画像的呈现的数据的存储器或存储装置。用户接口单元84表示用户可与其交互或以其它方式介接以与计算装置70的其它单元(例如,CPU72)通信的单元。用户接口单元84的实例包含(但不限于)跟踪球、鼠标、键盘,以及其它类型的输入装置。用户接口单元84还可为触摸屏且可被集成为显示单元78的一部分。

[0056] 计算装置70可包含出于清楚起见而在图2中未展示的额外模块或单元。举例来说,计算装置70可包含扬声器和麦克风(其任一者都未在图2中展示),从而在其中计算装置70是移动无线电话的实例中实现电话通信,或在计算装置70是媒体播放器的情况下实现扬声器。在一些例子中,在其中计算装置78是经装备以与外部用户接口或显示器介接的桌上型计算机或其它装置的实例中,用户接口单元84和显示单元78可在计算装置78的外部。

[0057] 如图2的实例中所说明,存储单元76存储GPU驱动器86、GPU程序88,及编译器92。如下文更详细地描述,存储单元76还可存储在本地产编译的GPU程序94和纹理数据96。GPU驱动器86表示提供用以存取GPU74的接口的计算机程序或可执行代码。CPU72执行GPU驱动器86或其部分以与GPU74介接,且出于此原因,GPU驱动器86在图2的实例中展示为CPU72内的标记为“GPU驱动器86”的虚线框。GPU驱动器86是可由CPU72执行的程序或其它可执行体存取的,包含GPU程序88。

[0058] GPU程序88可包括以HL编程语言编写的程序,所述HL编程语言例如为开发计算语言(其通常被称为“OpenCL”)、OpenGL ES、HLSL或利用由GPU88提供的专用CPU专有操作的另一HL编程语言。使用OpenGL规格而开发的GPU程序可被称作着色器程序。或者,使用OpenCL规格而开发的GPU程序可被称作程序内核。然而,本发明的技术不限于特定HL编程语言。举例来说,GPU程序88可嵌入或以其它方式包含于在CPU72上执行的另一程序内。

[0059] GPU程序88可调用或以其它方式包含由GPU驱动器86提供的一或多个功能。CPU72一股执行GPU程序88嵌入于其中且在遇到GPU程序88后即刻就将GPU程序88传递到GPU驱动器86的程序。CPU72在此背景下执行GPU驱动器86以处理GPU程序88。也就是说,例如,GPU驱动器86可通过将GPU程序88编译为可由GPU74执行的对象或机器代码来处理GPU程序88。此对象代码在图3的实例中被展示为在本地产编译的GPU程序94。

[0060] 为了编译此GPU程序88,GPU驱动器86包含编译GPU程序88的编译器92。也就是说,在一些实例中,编译器92可大体上类似于上文关于图1所描述的编译器20,不同之处在于编译器92在GPU程序88嵌入于其中的程序的执行期间实时地或准实时地操作以编译GPU程序88。编译器92可利用本发明的技术将具有纹理加载指令的CF指令转换为具有判定结构的指令。

[0061] 举例来说,编译器92可在执行包含GPU程序88的HL代码时从CPU72接收GPU程序88。编译器92可编译GPU程序88以产生符合LL编程语言(例如,机器语言或汇编语言)的在本地产编译的GPU程序94。在一些实例中,可根据OpenGL ES着色语言来界定GPU程序94。GPU程序88可包含HL CF或编译器92根据本发明的技术进行编译的包含纹理加载指令的其它循环指令。编译器92随后输出包含LL指令的在本地产编译的GPU程序94。

[0062] GPU74一般接收在本地编译的GPU程序94(如GPU74内的标记为“在本地编译的GPU程序94”的虚线框所展示),于是,在一些例子中,GPU74再现图像且将所述图像的经再现的部分输出到显示缓冲器单元80。显示缓冲器单元80可暂时地存储经再现图像的经再现像素,直到整个图像被再现为止。在此背景下,显示缓冲器单元80可被视为图像帧缓冲器。显示缓冲器单元80可随后传输经再现的图像以在显示单元48上显示。在一些替代实例中,GPU74可将图像的经再现部分直接输出到显示单元78以供显示,而不是暂时地将所述图像存储在显示缓冲器单元80中。显示单元78可随后显示存储于显示缓冲器单元78中的图像。

[0063] 在经编译的GPU程序94的执行期间,可在存储单元76与GPU存储器75B之间传递数据。举例来说,GPU存储器75B(其可为芯片上存储器)的容量与存储单元76的容量相比可相对小。因此,当执行程序94时,GPU74可将数据从存储单元76传递到存储器75B、处理所述数据,且将经处理的数据存储到存储单元76和/或显示缓冲器80。在一些例子中,限制在存储单元76与存储器75B之间传递的数据的量和/或传递数据的次数可提高性能。举例来说,在存储单元76与存储器75B之间传递数据可引入与数据传递相关联的等待时间。此类等待时间可呈现瓶颈,尤其是在运行时间期间(例如,在其期间,编译GPU程序88以形成经编译的GPU程序94且执行)从存储单元76将数据获取到存储器75B的情况下。另外,从存储单元76获取数据会消耗电力。对于使用有限电源进行供电的例如计算装置70等计算装置(例如,通过电池供电的移动计算装置),减少电力消耗会增加计算装置70在两次充电之间可操作的使用时间。

[0064] 在一些情况下,GPU74可在经编译的GPU程序94的执行期间产生且/或存取纹理数据96,且可将纹理数据96存储到存储单元76和/或显示缓冲器80。纹理数据96可包含(例如)含有具有相同格式的一或多个图像的一或多个对象(例如,具有特定大小和特定格式的某一维度的像素的阵列)。GPU74可将纹理数据96存储到GPU存储器75B,使得所述纹理数据可在再现期间使用。在一实例中,纹理数据96可包含与特定色彩相关联的数据。当再现包含特定色彩的像素数据时,GPU74可将适当的纹理数据从存储单元76加载到存储器75B。也就是说,GPU74可执行经编译的GPU程序94的纹理加载指令以将特定纹理从存储单元76传递到存储器75B。GPU可随后使用所述纹理再现像素数据且将经再现的像素数据存储到显示缓冲器单元80。

[0065] 根据本发明的各方面,编译器92可将GPU程序88的包含纹理加载指令(例如,致使GPU74将纹理数据96从存储单元76加载到本地GPU存储器75B的指令)的HL CF指令转换为具有判定结构的LL指令。举例来说,编译器92可在转换HL指令之后产生在本地编译的GPU程序94以供GPU74执行,其移除CF指令。编译器92可以一种方式转换GPU程序88的HL CF指令,使得避免执行与所述组CF指令的每一分支相关联的所有纹理加载指令,且进而可减少对存储单元78的存取的数目。

[0066] 在一些实例中,编译器92可通过在纹理坐标上产生具有判定结构的LL指令而将具有纹理加载的HL CF指令转换为具有判定结构的指令。举例来说,如关于图5更详细地描述,编译器92可产生具有判定结构的指令,其结果可指向特定存储器地址,例如存储器75B中的存储器地址。编译器92还可产生例如基于具有判定结构的指令而将纹理加载到存储器75B中的特定存储器地址的指令。

[0067] 以此方式,编译器92可增加GPU74的效率。举例来说,将具有纹理加载指令的CF指

令转换为具有判定结构的指令可允许GPU74串行地执行经编译的GPU程序94(不具有反馈环路)。另外,通过限制被执行的纹理加载指令的数目,可最小化与将数据从存储单元76传递到存储器75B相关联的等待时间。另外,限制被执行的纹理加载指令的数目还可例如通过限制在存储单元76与存储器75B之间传递的数据的量而有助于节省电力。

[0068] 在一些实例中,如上文关于图1所描述,编译器92可将本发明的技术应用于GPU程序88的具有对称纹理加载指令的CF指令。也就是说,如果纹理加载指令关于正被转换的一组CF指令是近似对称的(例如,具有相对成比例的数目的纹理加载指令的分支),那么编译器92可执行本发明的技术将所述组CF指令转换为具有判定结构的指令。然而,如果所述纹理加载指令不对称(例如,仅一个路径包含纹理加载指令),那么编译器92可制止转换CF指令。

[0069] 根据本发明的一些方面,通过从GPU程序88移除具有相关联的纹理加载指令的CF指令,编译器92可在调度经编译的GPU程序94的执行的方面具有增加的灵活性。举例来说,如果在具有纹理加载的CF指令之前存在包含于HL GPU程序88中的(CF指令的)CF条件和纹理加载坐标与其不具有相依性的一或多个其它指令,那么编译器20可调度纹理加载指令在所述一或多个其它指令之前执行。

[0070] 图3说明例如通过编译器20或92将包含一组CF指令100的HL伪码转译为不具有判定结构的LL指令104(“不具有转换”)和具有判定结构的LL指令108(“具有转换”)。如图3中所示,不具有转换的LL指令104包含例如分支(BR)和跳跃(JMP)命令等CF指令。因此,LL指令104可不串行地执行。也就是说,例如,可执行跳跃命令JMPL2,进而改变指令指针寄存器,使得执行L2(而非L1)。

[0071] 然而,LL指令108已被转译,使得分支(BR)和跳跃(JMP)命令已被移除。因此,LL指令108可串行地执行。虽然执行LL指令108可由于移除CF指令而导致相对于LL指令104执行更多指令,但例如GPU74等某些处理单元可比LL指令104更快地且/或以更高程度的准确度(例如,导致适当的值被存储到适当的寄存器)执行LL指令108。举例来说,可包含于GPU74(ALU)中的算术逻辑单元(未展示)的并行性质可能未有效地处置分支、跳跃或其它CF指令。

[0072] 在一些实例中,编译器(例如,编译器20或编译器92)可被约束在可被转换为具有判定结构的指令的指令中。此类编译器可被称作“部分判定”编译器,因为所述编译器仅可转换有限组的指令,例如,“move”(例如,MOVC)和“select”(例如,SELECT)指令。

[0073] 图4说明将包含具有纹理加载指令(“纹理(地址1)”和“纹理(地址2)”)的一组CF指令112的HL伪码转译为具有判定结构的LL指令116。可由例如编译器20(图1)或编译器92(图2)等编译器执行此转译。然而,可通常不执行此转译,因为必须执行CF指令112的两个纹理加载指令。也就是说,通过以图4中所示的方式转换CF指令112,执行所述纹理加载指令中的每一者且选择适当的纹理。

[0074] 如上所述,执行纹理加载指令导致将纹理数据从外部存储器传递到本地存储器。因此,相对高的等待时间可与纹理加载指令相关联。归因于此高等待时间,编译器可不执行图4的实例中所示的转换。

[0075] 图5说明根据本发明的技术将包含具有纹理加载指令(“纹理(地址1)”和“纹理(地址2)”)的一组CF指令120的HL伪码转译为具有判定结构的LL指令124。可由例如编译器20(图1)或编译器92(图2)等编译器执行此转译。应理解,仅出于实例的目的而提供图5的代

码。也就是说,虽然图5说明if-else指令,但编译器可以类似方式将其它CF和/或循环指令(包含更复杂的指令)转译为判定指令。另外,仅出于实例的目的而提供框124的LL汇编语言指令,且可将CF指令120编译为一或多种其它LL语言。

[0076] 如图5中所示,根据本发明的各方面,可将具有纹理加载指令的所述组CF指令120转换为具有判定结构的LL指令124。例如通过在纹理坐标上产生具有判定结构的LL指令124来执行所述转换。举例来说,LL指令124包含变量p0与条件(cond)的比较(cmp)。基于所述条件的结果来保留地址(addr)。地址调用与选择函数(sel)组合使用(addr)。也就是说,取决于选择的结果(例如,如果sel p0=真,那么选择地址1处的纹理,如果否,那么选择地址2处的纹理),addr保持地址1或地址2的值。随后将结果命令用于加载适当的纹理(addr)。以此方式,可串行地执行命令(例如,不存在分支或跳跃命令),且仅加载一个纹理。因此,本发明的技术允许将具有纹理加载的控制流指令转换为判定结构,而不具有上文所描述的缺陷(例如,多个纹理加载及存储器存取违规的可能性)。

[0077] 在一些实例中,编译器(例如编译器92)可检测来自THEN和ELSE块两者的对称纹理加载,且仅对对称的纹理加载执行优化。举例来说,如果特定指令仅在一个分支中包含纹理(例如,if“x” then“y”,else“加载纹理”,其中“y”不是纹理加载),那么可不执行所述转换,因为可不需要此转换。也就是说,在此实例中,仅存在用于一个纹理加载的可能性,因此转换未由于消除纹理加载而增加效率。

[0078] 本发明的技术可用于使用判定的纹理坐标来“提升”控制流之外的纹理加载。也就是说,可与其它指令(被称作“提升”)独立地执行纹理加载,因为不存在跳跃、分支或可能影响纹理加载的结果的其它指令。因此,在与其它指令没有相依性的情况下,编译器可调度纹理加载以在再现过程中相对早地执行。以此方式,纹理在再现期间需要它们时可就绪,而不是必须等待从存储器加载纹理(其可消耗相对大量的时钟循环)。

[0079] 图6说明根据本发明的技术将具有相关联的纹理加载指令的一组CF指令转换为具有判定结构的指令的方法。虽然将图6的方法描述为由计算装置70(图2)的编译器92实行,但应理解,可通过多种其它计算装置和编译器来实行本发明的技术。

[0080] 在图6的实例中,编译器92识别具有相关联的纹理加载指令的一组HL CF指令(130)。HL CF指令可包含(例如)多种有条件和/或循环指令。实例包含if指令、if-else指令(if-then-else指令)、if-if指令,或可借以控制处理器对指令的执行流的任何其它指令(例如,可控制指令指针寄存器的指令)。由编译器92识别的所述组CF指令可包含一个以上分支(例如,界定分叉路径),且每一分支的每一路径可包含一或多个纹理加载指令。另外,在一些实例中,所述组CF指令可为例如子例程等一组较大指令的一部分。

[0081] 在识别具有相关联的纹理加载指令的所述组HL CF指令之后,编译器92将所述组HL CF指令转换为具有判定结构的LL指令(134)。也就是说,根据本发明的各方面,编译器92产生具有判定结构的LL指令。另外,编译器92产生所述LL指令以限制包含于所述LL指令中的纹理加载指令的数目。举例来说,如上文所描述,编译器92可在纹理坐标上产生指令,使得HL CF指令的所有纹理加载指令不需要包含于经转换的LL指令中。

[0082] 根据本发明的各方面,编译器92可在运行时间处执行所述转换,例如,作为在执行含有HL CF指令的程序期间的运行时间编译器。因此,编译器92可随后输出经转换的LL指令以供执行(138)。然而,编译器92还可输出经转换的LL指令以存储到(例如)例如存储单元76

等存储媒体(138)。

[0083] 应理解,取决于实例,可以不同序列执行、可添加、合并或完全省去本文中所描述的方法(例如,图6中所示的方法)的某些动作或事件(例如,不是所有的所描述的动作或事件对于所述方法的实践来说都是必要的)。此外,在某些实施例中,可同时(例如,通过多线程处理、中断处理或多个处理器)而非循序地执行动作或事件。另外,虽然本发明的某些方面被描述为由单一模块或单元执行以用于清楚的目的,但应理解,本发明的技术可由与计算装置相关联的单元或模块的组合执行。

[0084] 在一或多个实例中,所描述的功能可以硬件、软件、固件或其任何组合来实施。如果以软件实施,则可将功能作为计算机可读媒体上的一或多个指令或代码而加以存储或传输。计算机可读媒体可包含计算机数据存储媒体或通信媒体两者,通信媒体包括促进将计算机程序从一处传递到另一处的任何媒体。数据存储媒体可为可由一或多个计算机或一或多个处理器存取以检索指令、代码和/或数据结构来用于实施本发明中所描述的技术的任何可用媒体。

[0085] 举例来说(且并非限制),所述计算机可读媒体可包括RAM、ROM、EEPROM、CD-ROM或其它光盘存储装置、磁盘存储装置或其它磁性存储装置、快闪存储器或可用于载送或存储呈指令或数据结构的形式所要程序码且可由计算机存取的任何其它媒体。同样,可恰当地将任何连接称作计算机可读媒体。举例来说,如果使用同轴电缆、光纤电缆、双绞线、数字订户线(DSL)或例如红外线、无线电及微波的无线技术从网站、服务器或其它远程源传输软件,则同轴电缆、光纤电缆、双绞线、DSL或例如红外线、无线电及微波的无线技术包含于媒体的定义中。如本文中所使用,磁盘及光盘包括压缩光盘(CD)、激光光盘、光学光盘、数字多功能光盘(DVD)、软磁盘及蓝光光盘,其中磁盘通常磁性地重现数据,而光盘使用激光光学地重现数据。以上各者的组合也应包含在计算机可读媒体的范围内。

[0086] 所述代码可由一或多个处理器执行,例如,一或多个数字信号处理器(DSP)、通用微处理器、专用集成电路(ASIC)、现场可编程逻辑阵列(FPGA)或其它等效整合或离散的逻辑电路。因此,如本文中所使用的术语“处理器”可指前述结构或适合于实施本文中所描述的技术的任一其它结构中的任一者。另外,在一些方面中,本文中所描述的功能性可提供于专用硬件模块和/或软件模块内。并且,可将所述技术完全实施于一或多个电路或逻辑元件中。

[0087] 本发明的技术可实施于广泛多种装置或设备中,包含无线手持机、集成电路(IC)或IC组(例如,芯片组)。本发明中描述各种组件、模块或单元来强调经配置以执行所揭示的技术的装置的若干功能性方面,但不一定需要通过不同的硬件单元来实现。而是,如上文所描述,各种单元可联合合适的软件和/或固件而组合于硬件单元中或通过互操作的硬件单元的集合(包含如上文所描述的一或多个处理器)来提供。

[0088] 已描述了本发明的各种实例。在不脱离本发明的范围的情况下,可作出各种修改。这些及其它方面属于所附权利要求书的范围内。

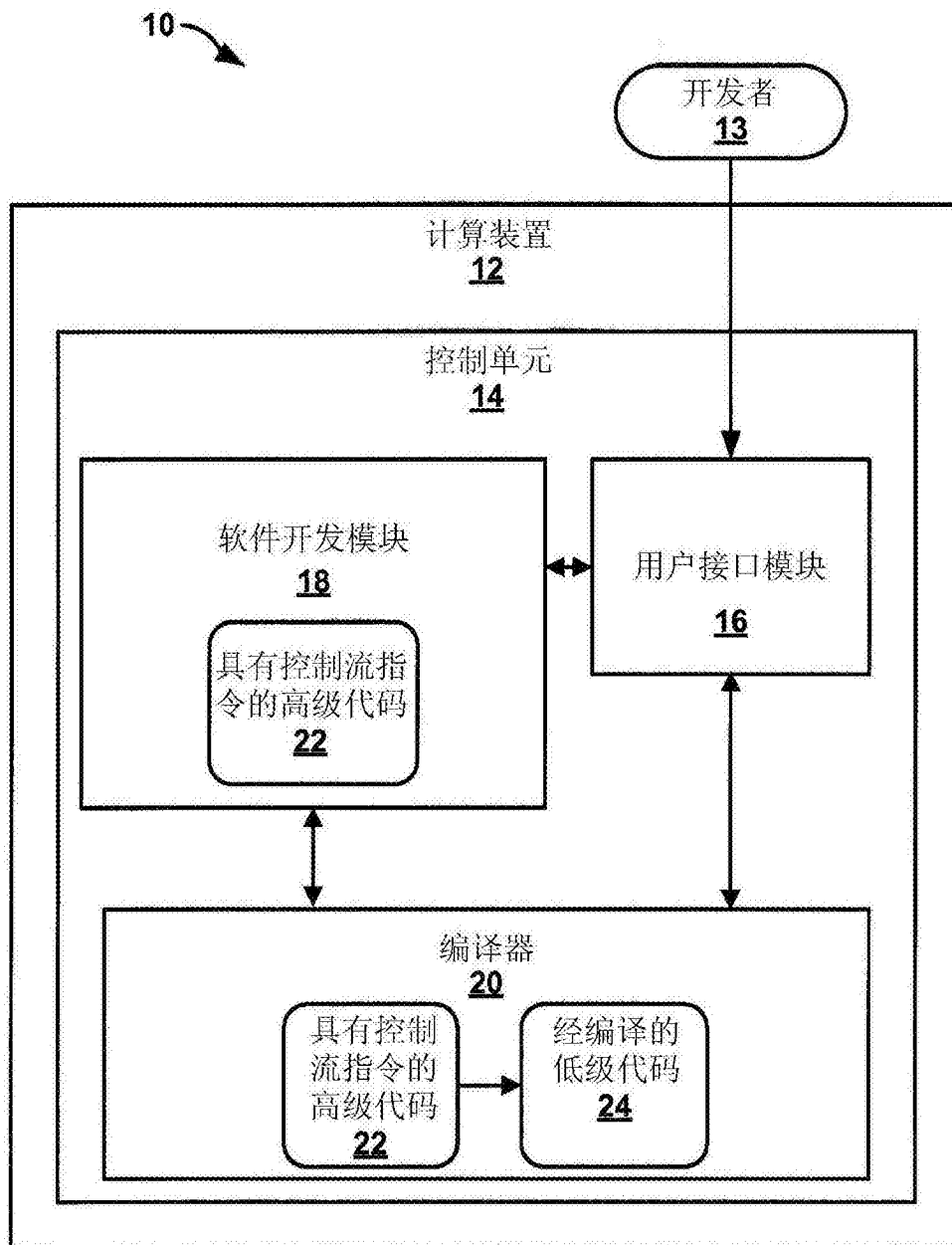


图1

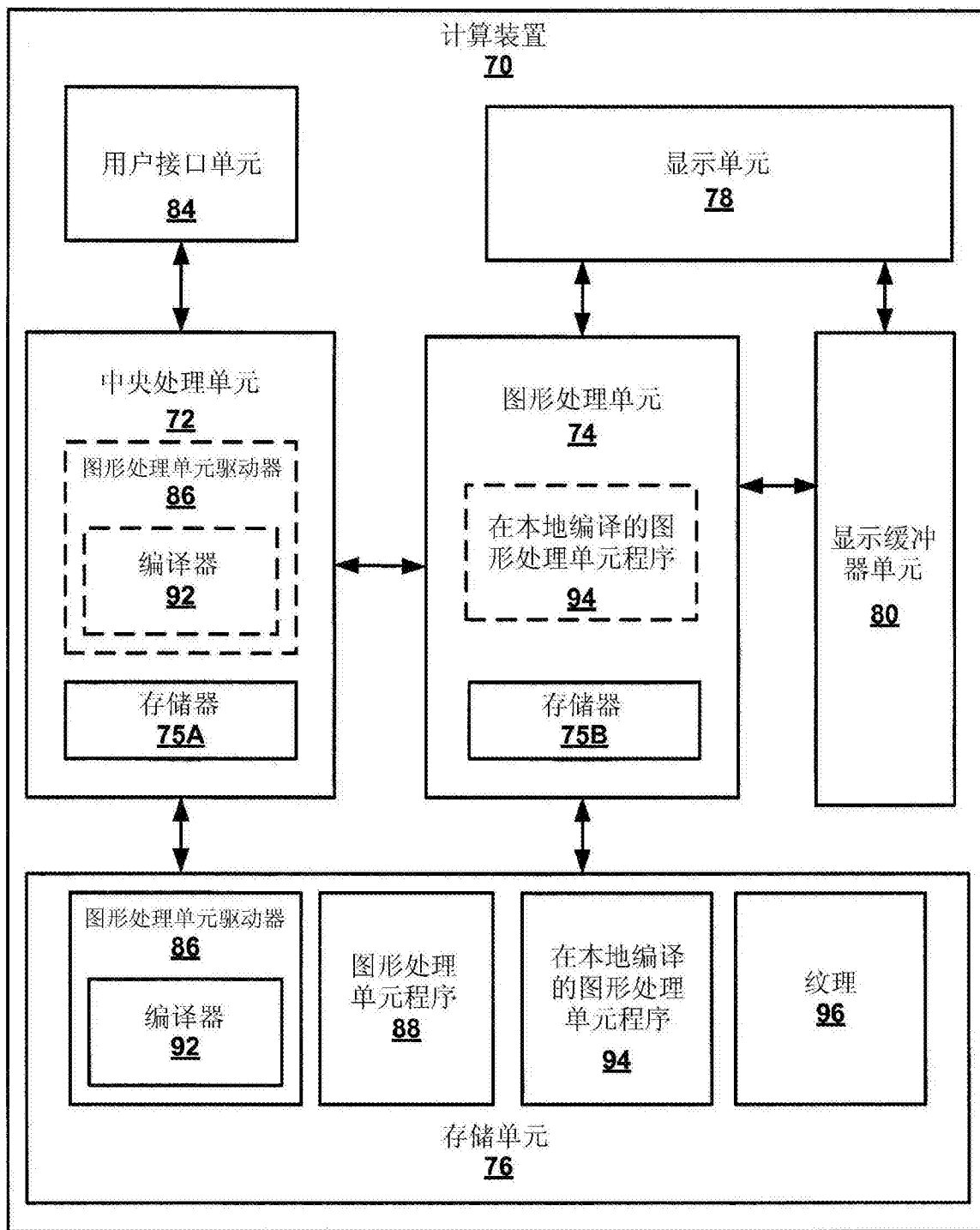


图2

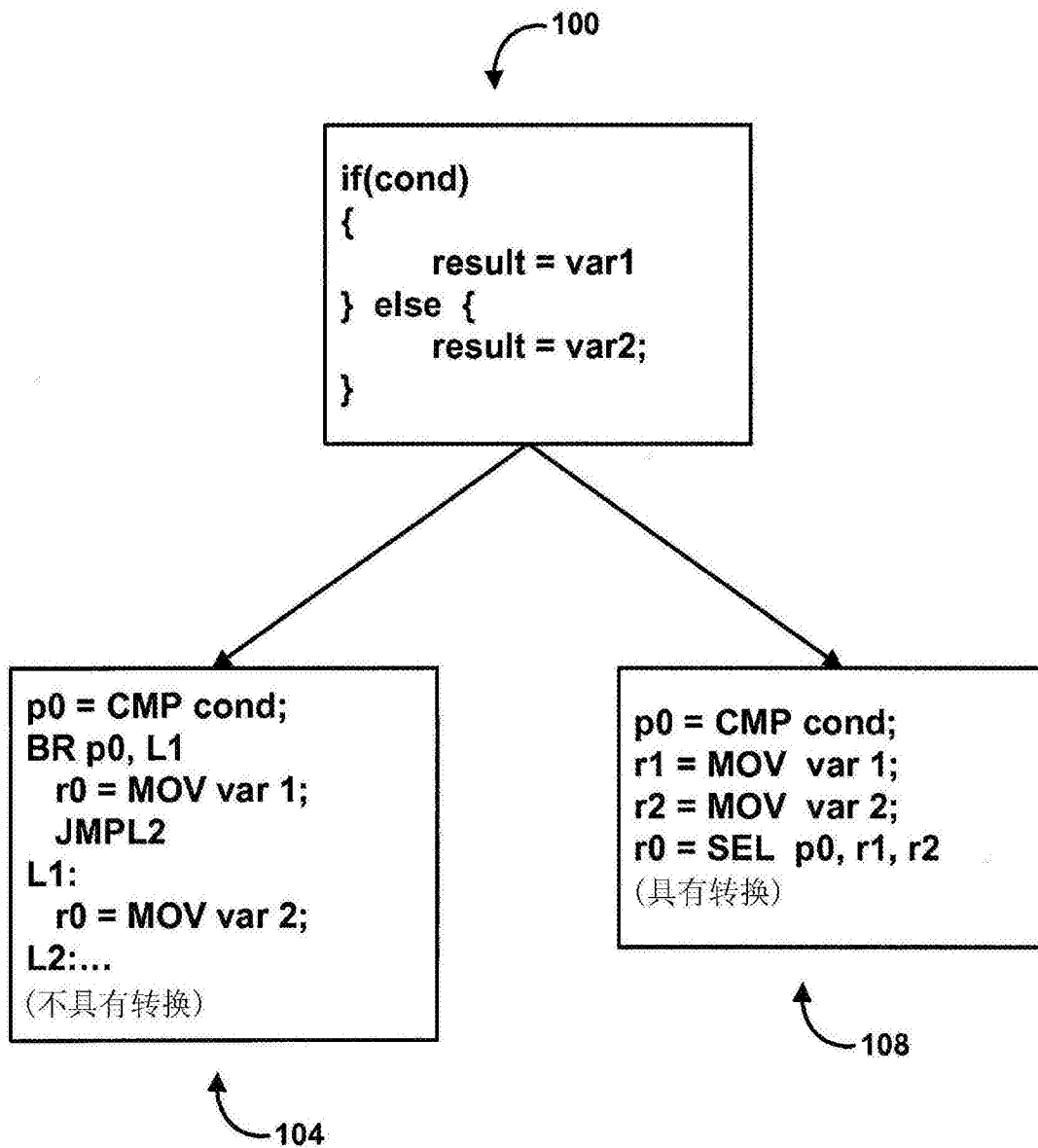


图3

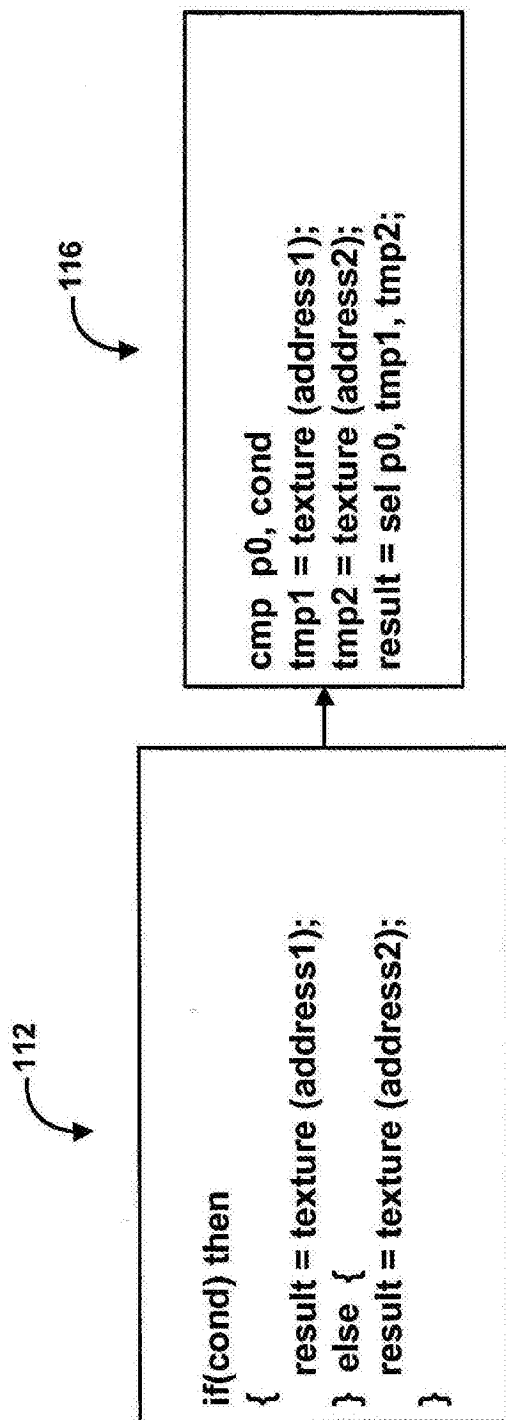


图4

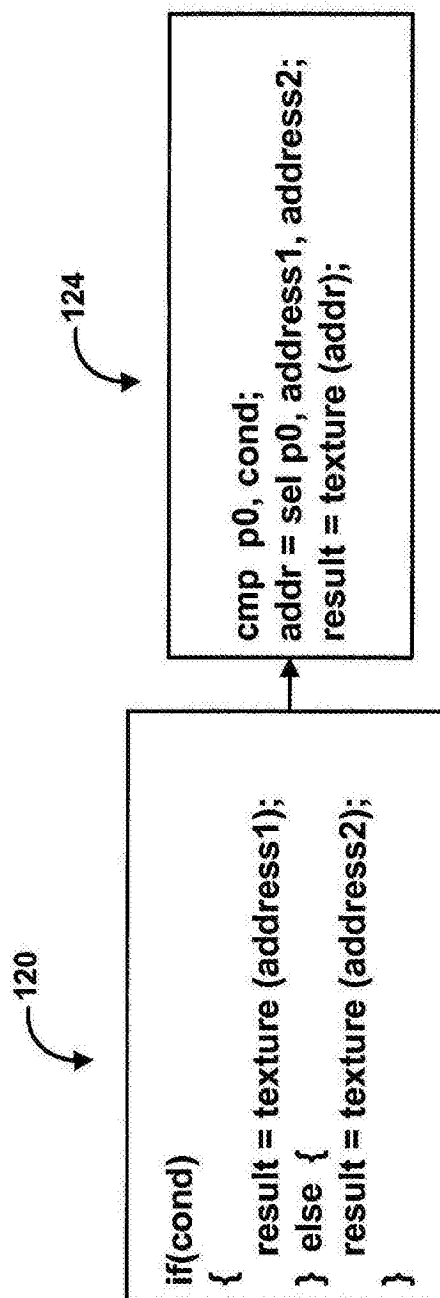


图5

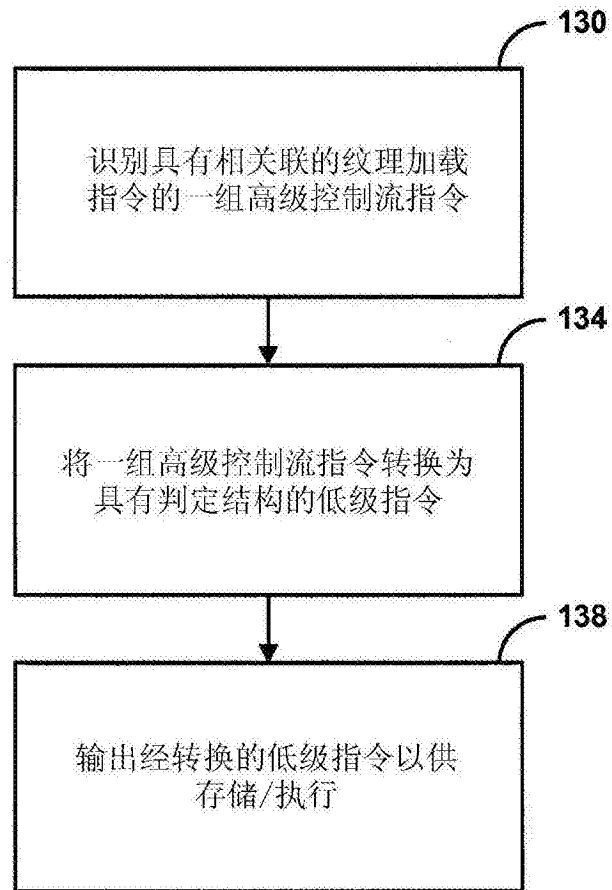


图6