



(12)发明专利申请

(10)申请公布号 CN 106843948 A

(43)申请公布日 2017.06.13

(21)申请号 201710016922.5

(22)申请日 2017.01.11

(71)申请人 上海鲲云信息科技有限公司

地址 201203 上海市浦东新区中国(上海)
自由贸易试验区祖冲之路2305号B栋
502室

(72)发明人 牛昕宇

(74)专利代理机构 上海光华专利事务所 31219

代理人 徐秋平

(51) Int. Cl.

G06F 9/445(2006.01)

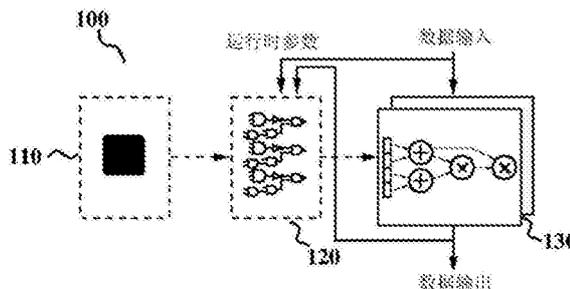
权利要求书2页 说明书6页 附图3页

(54)发明名称

片上生成硬件配置信息的芯片架构及其执行方法

(57)摘要

本发明提供片上生成硬件配置的芯片架构及其执行方法,芯片架构包括:配置生成单元、配置存储单元、及动态数据通路单元,其中,所述配置生成单元,连接所述动态数据通路单元,用于根据硬件运行时输入的动态参数生成配置文件信息,并将所述配置文件信息输入所述动态数据通路单元中,以供所述动态数据通路单元根据所述配置文件信息完成硬件重构;所述配置存储单元,连接所述动态数据通路单元,用于接收并存储由所述动态数据通路单元发送的所述配置文件信息。本发明硬件芯片的配置文件由片上电路生成,保证硬件电路重构能够在一个时钟周期完成,与此同时,增加片上配置文件生成的硬件面积,功耗远小于在芯片中存储所有可能硬件配置的架构。



1. 一种片上生成硬件配置的芯片架构,其特征在于,包括:配置生成单元、配置存储单元、及动态数据通路单元,其中,

所述配置生成单元,连接所述动态数据通路单元,用于根据硬件运行时输入的动态参数生成配置文件信息,并将所述配置文件信息输入所述动态数据通路单元中,以供所述动态数据通路单元根据所述配置文件信息完成硬件重构;

所述配置存储单元,连接所述动态数据通路单元,用于接收并存储由所述动态数据通路单元发送的所述配置文件信息。

2. 根据权利要求1所述的片上生成硬件配置的芯片架构,其特征在于,所述配置生成单元中预先定义有配置生成函数,所述配置文件信息是由所述动态参数经所述配置生成函数而生成的。

3. 根据权利要求1所述的片上生成硬件配置的芯片架构,其特征在于,所述动态数据通路单元包括:控制单元、动态数据路由单元、数据通路操作单元、及至少一个数据内存单元,其中,

所述控制单元,连接所述配置生成单元,用于接收所述配置文件信息,并将所述配置文件信息发送至所述配置存储单元;

所述动态数据路由单元,连接所述控制单元,用于将所述至少一个数据内存单元连接至所述数据通路操作单元;

所述数据通路操作单元,连接所述动态数据路由单元,用于以所述配置文件信息重构后的硬件电路来处理数据。

4. 根据权利要求3所述的片上生成硬件配置的芯片架构,其特征在于,所述控制单元包括:配置内存单元,用于存储所述配置文件信息,以供定义所述动态数据路由单元和所述数据通路操作单元的硬件功能。

5. 根据权利要求4所述的片上生成硬件配置的芯片架构,其特征在于,所述控制单元还包括:模式选择单元,连接所述配置内存单元,用于决定所述控制单元的运行状态是静态模式还是动态模式,当所述控制单元运行于所述静态模式,所述配置内存单元中的硬件电路配置文件保持其初始状态,不可被改写;当所述控制单元运行于所述动态模式,存储于所述配置内存单元的硬件电路配置文件可由所述配置生成单元改写。

6. 根据权利要求1或3所述的片上生成硬件配置的芯片架构,其特征在于,还包括:配置分配控制单元,连接所述动态数据通路单元,用于控制所述配置存储单元是否可以被更新。

7. 根据权利要求6所述的片上生成硬件配置的芯片架构,其特征在于,还包括:配置初始化单元和行解码器,连接所述动态数据通路单元,用于对所述配置分配控制单元、所述控制单元进行初始化。

8. 一种执行如权利要求1-7中任一所述的芯片架构的方法,其特征在于,包括:

将初始配置文件信息存储于动态数据通路单元中,定义动态数据通路单元的初始功能和配置生成单元的配置生成函数;

配置生成单元接收硬件运行时输入的动态参数;

配置生成单元根据输入的动态参数,生成新的配置文件信息;

在同一个时钟周期,配置生成单元将生成的配置文件信息输入动态数据通路,更新动态数据通路的硬件配置,从而重构硬件电路;

在同一个时钟周期,重构完成的硬件电路接收输入数据,基于配置后的优化硬件电路处理运行时数据。

片上生成硬件配置信息的芯片架构及其执行方法

技术领域

[0001] 本发明涉及逻辑可编程阵列芯片领域,特别是涉及片上生成硬件配置信息的芯片架构及其执行方法。

背景技术

[0002] 逻辑可编程阵列(FPGA)是一种在生产后可编程的集成电路芯片。芯片中电路提供可编程节点,可根据用户设定重新定义电路逻辑。相比于传统处理芯片CPU,FPGA可提供针对特定问题的高度优化电路,提升百倍级别计算性能。相比于传统集成电路芯片(ASIC),FPGA提供更灵活的计算方案。

[0003] 逻辑可编程阵列芯片中含有存储硬件电路配置文件的配置存储单元(CM)。与应用专用芯片相比(ASIC),配置存储单元支持对于硬件电路配置进行更新,从而在无需流片的情况下更新硬件电路,提供更强的通用性。在实际应用中,逻辑可编程阵列芯片所提供的通用性及硬件高性能支持其在各个领域的应用。当CM中的硬件配置文件更新完毕后,硬件电路采用同一配置处理数据。

[0004] 对于逻辑可编程阵列芯片,在CM的配置文件确认前,不同的电路配置可以进行设计及优化提供最优的硬件性能。然而在配置文件确认后,逻辑可编程阵列芯片与ASIC的处理方式一致:固定的硬件电路处理不同的数据。当应用执行中所支持的操作需要变换时,逻辑可编程阵列芯片需要支持所有可能的硬件电路,从而根据应用执行条件在可能的硬件电路中挑选。在此情况下,所有可能的硬件电路中只有1个硬件电路得到执行,其余电路处于闲置状态。当应用中操作的变换可能增多时,对应的闲置硬件电路增多,采用硬件支持应用的性能增加迅速消失:受限于动态操作(拥有多于一种执行操作可能的操作,如if-else语句),逻辑可编程芯片上所支持的硬件模块不再为特殊操作而定制,转为支持该动态操作所有的可能操作的通用电路,降低了所支持的定制硬件性能。

[0005] 为解决这一问题,可采用动态可重构技术,在应用执行过程中更新CM中的配置文件。动态可重构技术准备不同的硬件电路配置,每个配置对应动态操作的一种可能,并在动态操作在不同操作间切换时将不同配置文件写入CM,更新对应的定制硬件。每个硬件配置下都可保证最优的硬件性能。动态可重构方法解决了硬件电路在每个可能操作下硬件优化的问题,与之而来的是其所需的额外的更新配置文件时间,称为重构时间。为降低重构时间,部分可重构的概念被引入:每次重构配置文件只改变需要变动部分,从而降低重构时间。部分可重构技术的重构时间仍然在毫秒至秒级范围。为了进一步降低部分重构时间,配置文件中相同的内容被同时配置。在“An area-efficient partially reconfigurable crossbar switch with low reconfiguration delay,”FPL,2012,pp.400-406以及“Staticroute:A novel router for the dynamic partial reconfiguration of FPGAs,”FPL,2013,pp.1-7中,配置文件中相同的内容同时写入,减少了控制时间。然而减少后的配置时间仍然在次毫秒级。例如,在“A high I/O reconfigurable crossbar switch,”FCCM,2003,pp.3-10中,对于运行与150Mhz的硬件电路,配置优化后的部分重置电

路耗时220us。对于需要快速改变的动态操作,重构时间成本抵消了因为重置带来的性能优势。

[0006] 为进一步降低重构时间,专利U.S.Pat.No.5,426,378to Randy T.Ong entitled “Programmable Logic Device Which Stores More Than One Configuration and Means for Switching Configurations”,U.S.Pat.No.6,829,756B1to Stephen M.Trimberger entitled “Programmable Logic Device with Time-Multiplexed Interconnect”,和U.S.Pat.No.8,664,974B2to Rohe et al.entitled “Operational Time Extension”中描述了一种新的硬件重构方式:逻辑可编程芯片中包含多个CM,每个CM中存储针对动态操作中某一可能操作的优化电路,硬件电路根据应用情况在不同CM间切换。由于所有配置文件存储于芯片内部存储器中,硬件电路可在一个时钟周期内(10ns左右)完成电路切换。为支持在一个时钟内的硬件电路重构,所有可能的CM都需存储于芯片内部,大大增加了芯片的面积和功耗。对于不需要迅速重构的应用,这大大降低应用性能。

发明内容

[0007] 鉴于以上所述现有技术的缺点,本发明的目的在于提供片上生成硬件配置信息的芯片架构及其执行方法,不同于动态重构、部分可重构、以及多CM的架构,硬件芯片的配置文件由片上电路生成,保证硬件电路重构能够在一个时钟周期完成,与此同时,增加片上配置文件生成的硬件面积,功耗远小于在芯片中存储所有可能硬件配置的架构。

[0008] 为实现上述目的及其他相关目的,本发明提供一种片上生成硬件配置的芯片架构,包括:配置生成单元、配置存储单元、及动态数据通路单元,其中,所述配置生成单元,连接所述动态数据通路单元,用于根据硬件运行时输入的动态参数生成配置文件信息,并将所述配置文件信息输入所述动态数据通路单元中,以供所述动态数据通路单元根据所述配置文件信息完成硬件重构;所述配置存储单元,连接所述动态数据通路单元,用于接收并存储由所述动态数据通路单元发送的所述配置文件信息。

[0009] 于本发明一实施例中,所述配置生成单元中预先定义有配置生成函数,所述配置文件信息是由所述动态参数经所述配置生成函数而生成的。

[0010] 于本发明一实施例中,所述动态数据通路单元包括:控制单元、动态数据路由单元、数据通路操作单元、及至少一个数据内存单元,其中,所述控制单元,连接所述配置生成单元,用于接收所述配置文件信息,并将所述配置文件信息发送至所述配置存储单元;所述动态数据路由单元,连接所述控制单元,用于将所述至少一个数据内存单元连接至所述数据通路操作单元;所述数据通路操作单元,连接所述动态数据路由单元,用于以所述配置文件信息重构后的硬件电路来处理数据。

[0011] 于本发明一实施例中,所述控制单元包括:配置内存单元,用于存储所述配置文件信息,以供定义所述动态数据路由单元和所述数据通路操作单元的硬件功能。

[0012] 于本发明一实施例中,所述控制单元还包括:模式选择单元,连接所述配置内存单元,用于决定所述控制单元的运行状态是静态模式还是动态模式,当所述控制单元运行于所述静态模式,所述配置内存单元中的硬件电路配置文件保持其初始状态,不可被改写;当所述控制单元运行于所述动态模式,存储于所述配置内存单元的硬件电路配置文件可由所述配置生成单元改写。

[0013] 于本发明一实施例中,所述芯片架构还包括:配置分配控制单元,连接所述动态数据通路单元,用于控制所述配置存储单元是否可以被更新。

[0014] 于本发明一实施例中,所述芯片架构还包括:配置初始化单元和行解码器,连接所述动态数据通路单元,用于对所述配置分配控制单元、所述控制单元进行初始化。

[0015] 为实现上述目的及其他相关目的,本发明提供一种执行如上任一所述的芯片架构的方法,包括:将初始配置文件信息存储于动态数据通路单元中,定义动态数据通路单元的初始功能和配置生成单元的配置生成函数;配置生成单元接收硬件运行时输入的动态参数;配置生成单元根据输入的动态参数,生成新的配置文件信息;在同一个时钟周期,配置生成单元将生成的配置文件信息输入动态数据通路,更新动态数据通路的硬件配置,从而重构硬件电路;在同一个时钟周期,重构完成的硬件电路接收输入数据,基于配置后的优化硬件电路处理运行时数据。

附图说明

[0016] 图1显示为本发明一实施例的片上生成硬件配置信息的芯片架构示意图。

[0017] 图2显示为本发明另一实施例的片上生成硬件配置信息的芯片架构示意图。

[0018] 图3显示为本发明又一实施例的片上生成硬件配置信息的芯片架构示意图。

[0019] 图4显示为本发明一实施例的片上生成硬件配置信息的芯片架构的物理版图。

[0020] 图5显示为本发明一实施例的运行所述片上生成硬件配置信息的芯片架构的流程图。

[0021] 图6显示为不同动态可重构系统的架构效率示意图。

具体实施方式

[0022] 以下通过特定的具体实例说明本发明的实施方式,本领域技术人员可由本说明书所揭露的内容轻易地了解本发明的其他优点与功效。本发明还可以通过另外不同的具体实施方式加以实施或应用,本说明书中的各项细节也可以基于不同观点与应用,在没有背离本发明的精神下进行各种修饰或改变。需说明的是,在不冲突的情况下,以下实施例及实施例中的特征可以相互组合。

[0023] 需要说明的是,以下实施例中所提供的图示仅以示意方式说明本发明的基本构想,遂图式中仅显示与本发明中有关的组件而非按照实际实施时的组件数目、形状及尺寸绘制,其实际实施时各组件的型态、数量及比例可为一种随意的改变,且其组件布局型态也可能更为复杂。

[0024] 请参阅图1,本发明提供一种片上生成硬件配置的芯片架构100,包括位于同一片芯片中的:配置存储器110、配置生成器120、动态数据通路130。配置存储器110连接所述动态数据通路130,用于存储配置生成器120的硬件配置数据,配置生成器120连接所述动态数据通路,用于接收硬件运行时参数输入,生成配置文件信息,并输入动态数据通路130。

[0025] 请参阅图2,在一实施例中,动态数据通路130包括:控制单元132、动态数据路由单元134、数据通路操作单元136、各数据内存单元140。控制单元132用于接收硬件运行时生成的配置文件,并将该配置文件输入至配置存储器110,控制单元132的数量可以为多个,每个控制单元132配备一个动态数据路由单元134,动态数据路由单元134将各个数据内存单元

140连接至数据通路操作单元136,数据通路操作单元136用于根据配置后的硬件电路处理数据,具体由现场可编程阵列芯片中的可编程硬件支持。需要说明的是,数据内存单元140可位于芯片内部,也可位于芯片外部,经由片上存储单元的I/O接口或片外内存单元的I/O接口连接至动态数据路由单元134。配置分配网络138连接配置生成器120(配置生成器120由配置存储器110配置)与控制单元132,配合控制单元132选择合适配置输入,重构动态数据通路130。

[0026] 控制单元132具体包括:配置内存132a,用于存储硬件配置文件,以供定义动态数据路由单元134以及数据通路操作单元136的硬件功能,还可以包括:模式选择单元132b,用于决定控制单元132的运行模式是静态模式还是动态模式:当控制单元132运行于静态模式,配置内存132a中的硬件电路配置文件保持其初始状态,不可被改写;当控制单元132运行于动态模式,存储于配置内存132a的硬件电路配置文件可由配置生成器120改写。

[0027] 请参阅图3,为了支持配置分配网络138的不同功能需求,在一实施例中,片上生成硬件配置的芯片架构还包括:配置分配控制单元138a,与动态数据通路130相连,详细的,与模式选择单元132b相连。配置分配控制单元138a中含有专门为其服务的配置存储器,所述配置存储器存储的内容用于定义配置分配控制单元138a的硬件架构。配置分配控制单元138a可以控制其所包含的各个配置存储器中哪些可以更新,哪些需要在运行时保持不变,可以更新的配置存储器通过配置分配网络138连接至配置生成器120。

[0028] 请参阅图3,在一实施例中,片上生成硬件配置的芯片架构中还包括:配置初始化单元160,用于对配置分配控制单元138a以及控制单元132进行初始化,以及行解码器170。其中配置初始化单元160在芯片启动阶段触发初始化流程,控制行解码器170逐行更新配置内存132a以及配置分配控制单元138a中的配置存储器。需要说明的是,在典型的基于片上生成硬件配置的芯片架构中,配置分配网络138、控制单元132、配置分配控制单元138a、行解码器170,以及动态数据路由单元134可被包含在同一可重构电路150中。

[0029] 请参阅图4,基于所提出的片上生成硬件配置的芯片100的物理版图如图所示(四个大箭头指竖向的4个长方形,为内存单元140,其存储数据,并结合片上重构单元,具体结合如图3所示),其中,动态数据路由单元134与控制单元132等被包含在可重构电路150中。芯片100的各个数据内存单元140连接至可重构电路150中,可重构逻辑单元被配置为配置生成器120、动态数据通路130,以及数据通路操作单元136。

[0030] 请参阅图5,本发明还提供一种针对片上生成硬件配置的芯片架构的配置运行方法,该运行方法包括以下步骤:

[0031] 步骤S501:【硬件初始化】将初始配置文件存储于动态数据通路中的配置内存,定义动态数据通路的初始功能,以及配置生成器的配置生成函数。

[0032] 步骤S502:【动态参数接收】配置生成器接收运行时输入的动态参数。

[0033] 步骤S503:【硬件配置生成】配置生成器根据输入的动态参数,生成新的配置文件。

[0034] 步骤S504:【硬件重构】在同一个时钟周期,配置生成器将生成的配置文件输入动态数据通路的配置内存,更新动态数据通路的硬件配置,从而重构硬件电路。

[0035] 步骤S505:【数据处理】在同一个时钟周期,重构完成的硬件电路接收输入数据,基于配置后的优化硬件电路处理运行时数据。

[0036] 以下举例说明本发明提出的基于片上生成硬件配置的芯片的具体应用:

```

[0037]   int* x, int* y
         for i ∈ 1 → N - 1 do
             a = y[i-1];
[0038]   y[i] = (x[a+1] + x[a+2]) * x[a+3];
         end for

```

[0039] 在上述算法中x与y分别为函数的输入与输出,其中,x的数据方位地址a由上一个计算周期中y的结果决定。参照图2的硬件架构,数据x存储于数据内存140中。因此,每个计算周期,数据通路操作单元136所读取的x数据地址依赖于运行时的动态参数y[i-1]。在每个计算周期,数据通路操作单元136与数据内存单元140之间的路由连接需要动态改变。

[0040] 集成电路100可以支持此类动态数据互联。本例中y[i-1]作为动态参数输入配置生成器120。配置生成器120的输出通过配置分配网络138输入至控制单元132,改写配置内存132a中的配置文件。改写后的配置文件会更新硬件电路,完成动态数据路由单元134的硬件重构。重构后的动态数据路由单元134将根据y[i-1]定义的数据地址从数据内存单元140中读取数据到数据通路操作单元136进行处理。由于配置生成器120可以根据动态参数生成对应的硬件电路配置,芯片无需包含多个配置存储器以支持不同的硬件电路。与此同时,由于硬件配置由芯片内部电路生成,集成电路100的电路动态重构可以在一个时钟周期内完成。

[0041] 每个时钟周期,配置生成器120接收动态参数,生成针对动态数据通路130的配置文件。对于上述算法例子,每个时钟周期,动态数据路由单元134从数据内存140读取3个x数据至数据通路操作单元136。在同一时钟周期内,数据通路操作单元136生成一个数据处理结果y[i]。芯片内的数据路由资源(动态数据路由单元134)每个时钟周期完成重构,实现本时钟周期内所需的硬件电路,保证正确的数据可以读取到数据通路操作单元136,而不必在硬件上实现所有可能的数据连接。

[0042] 除此之外,本发明提出的方法的有益效果可用硬件架构效率E来表示:

$$[0043] \quad E = \frac{R}{O} \quad (1)$$

[0044] 其中,R为架构的重构性,表示为公式(2),O为重构成本,表示为公式(3):

$$[0045] \quad R = \min(r_{cap}, r_{cfg}) \quad (2)$$

[0046] 其中, r_{cap} 为动态操作的可能操作个数, r_{cfg} 为硬件架构支持的不同硬件配置个数,在图2的案例架构中,每个动态数据连接(例如x[a+1])拥有4个数据连接可能,对应数据内存单元140的四个输出端口,与此同时,硬件架构中的控制单元每个配置内存132a包含两个数据比特,对应4种可能的硬件配置,故而R为4。

[0047] 重构成本O表示为了支持动态重构带来的额外面积成本与时间成本:

$$[0048] \quad O = o_a \cdot o_t \quad (3)$$

[0049] 其中, o_a 为面积成本,表示为了支持动态重构所需的额外面积与初始无动态重构功能芯片的面积的比例,时间成本 o_t 定义为公式(4):

$$[0050] \quad O_t = \frac{t_r + t_p}{t_p} \quad (4)$$

[0051] 其中, t_r 为重构时间, t_p 为两次硬件重构之间的时间间隔。

[0052] 理想的可重构架构可在0重构成本 ($O=1$) 的前提下提供无上限重构能力 ($r_{\text{cfg}}=\infty$)。无上限重构能力代表当动态操作的可能操作数为任意值时,其重构能力都能够提供 $R=r_{\text{cap}}$ 。这种理想情况如图6中所中301所示。

[0053] 动态可重构系统的架构效率E如图6中所示,其中,302表示传统的动态重构流程的效率:硬件配置文件存储于芯片外部,通过芯片接口写入可重构芯片完成重构,303表示片上多个配置存储器架构的效率,304表示专利所提出架构的效率。

[0054] 对于基于芯片外部存储配置文件的重构架构,所有配置文件可在片外存储介质中保存。由于没有芯片上存储资源的限制,理论上基于芯片外部存储配置文件的重构架构可以支持无上限的重构能力 ($r_{\text{cfg}}=\infty$)。其重构成本主要来自于时间成本。以现场可编程真理芯片 (FPGA) 为例,一个配置文件最小单元为3232比特,芯片外部到芯片配置文件的通信带宽为400MB/s。因此,重构芯片内任意电路最少需要1.01微秒。基于FPGA的硬件电路时钟频率在100-200MHz内。1.01微秒对应的时钟周期为101-202个时钟周期。对于最小的101个周期,由于算法1需要每个时钟重置电路 ($t_p=1$),对应的重置时间 $t_r=101$, $O_t=102$,图6中的硬件架构效率302基本为0。

[0055] 基于多个片上配置存储器的重构架构可避免高重构时间,由于在一个周期内能够完成硬件重构(仅需进行片上配置文件切换),其重构时间成本为1 ($O_t=1$)。由于所有硬件配置文件需要存储于芯片内部,芯片面积随着所需支持的重构可能增多而增多。当 r_{cap} 增加时,其所需的存储空间现行增加用以保持硬件重构能力 $r_{\text{cfg}}=r_{\text{cap}}$ 。如图6中的303所示,对于上述算法中的例子,当动态数据访问的可能数据路由增加至32个时,由于存储32个配置文件带来的芯片面积成本为6.15(芯片面积增加了6.15倍),其硬件架构效率为5。

[0056] 图6中304显示了本专利所提出的重构架构的效率,如图1、2所示,芯片内部的配置生成器120生成配置文件,实时配置动态数据通路130,在一个时钟中期内其完成配置生成、电路配置、数据处理流程。因此,其重构时间成本1 ($O_t=1$)。与此同时,基于片上配置生成的重构架构无需为每个配置准备配置存储器,其面积成本仅为为动态数据通路增加配置可更新性的成本。对于典型FPGA芯片,其面积成本小于1.01(增加片上配置生成的FPGA架构面积增加小于1%)。如图6所示,本发明所提出的架构的效率接近最优效率301。

[0057] 综上所述,本发明有效克服了现有技术中的种种缺点而具高度产业利用价值。

[0058] 上述实施例仅例示性说明本发明的原理及其功效,而非用于限制本发明。任何熟悉此技术的人士皆可在不违背本发明的精神及范畴下,对上述实施例进行修饰或改变。因此,举凡所属技术领域中具有通常知识者在未脱离本发明所揭示的精神与技术思想下所完成的一切等效修饰或改变,仍应由本发明的权利要求所涵盖。

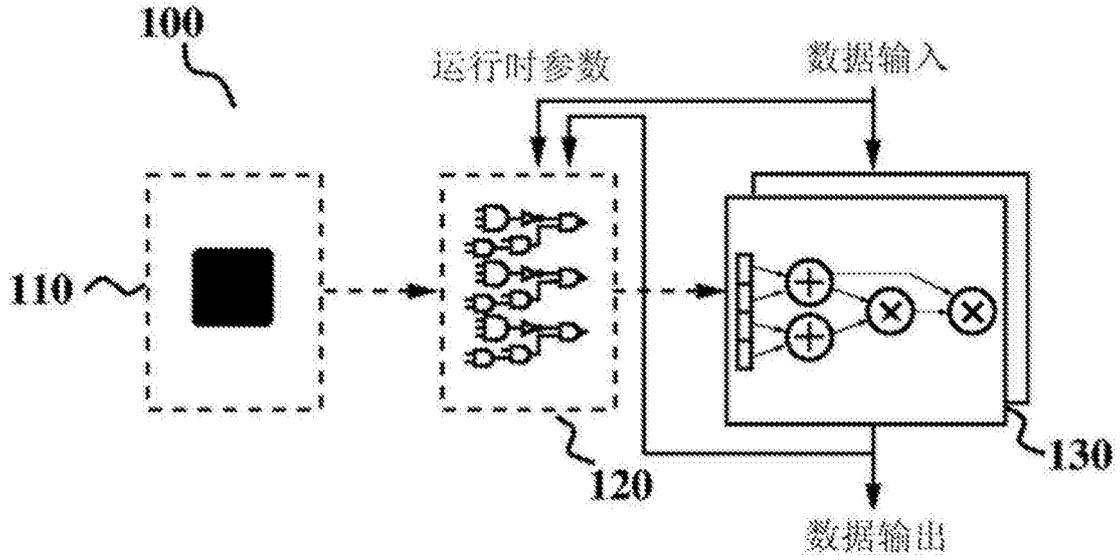


图1

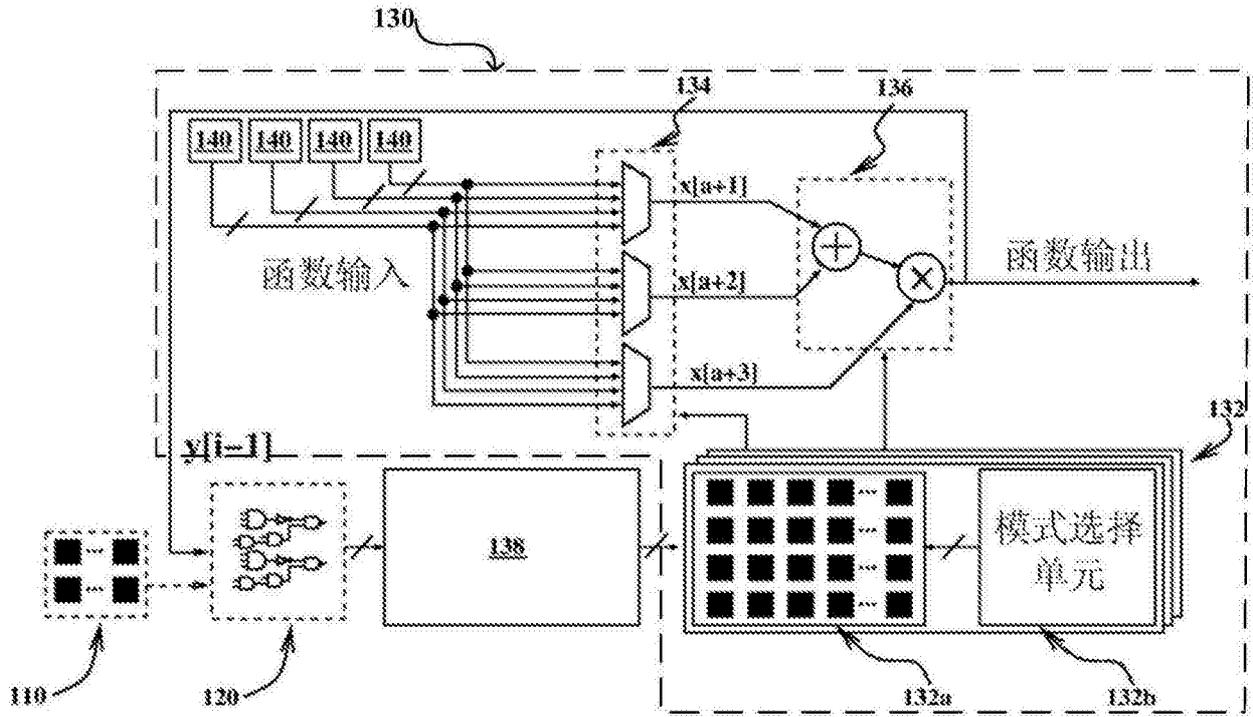


图2

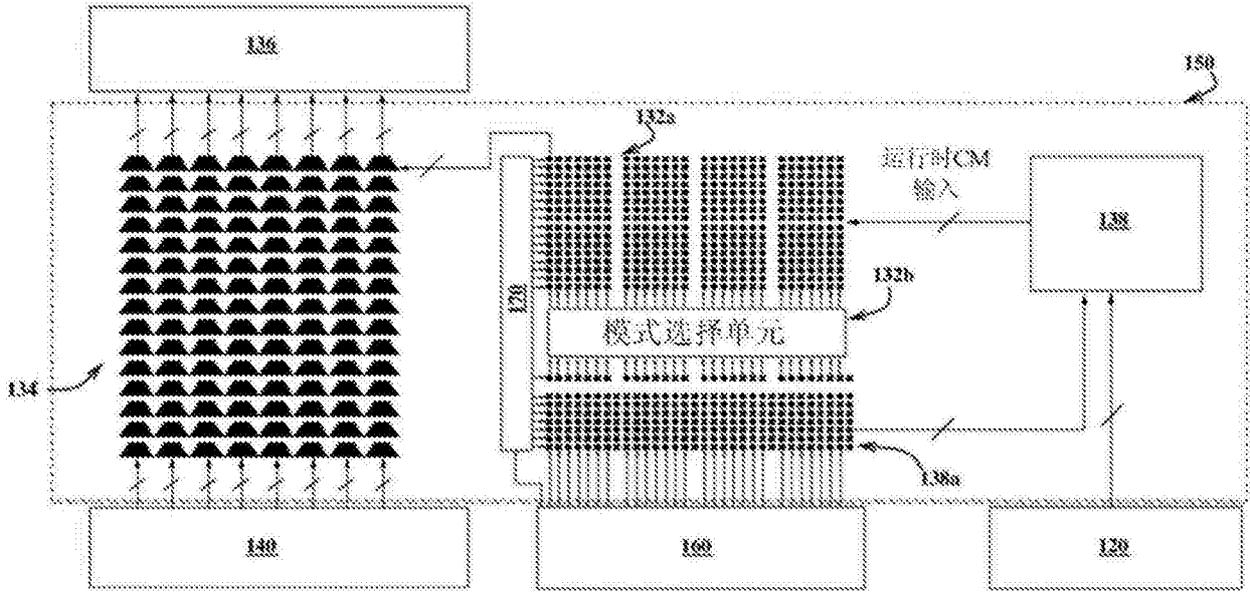


图3

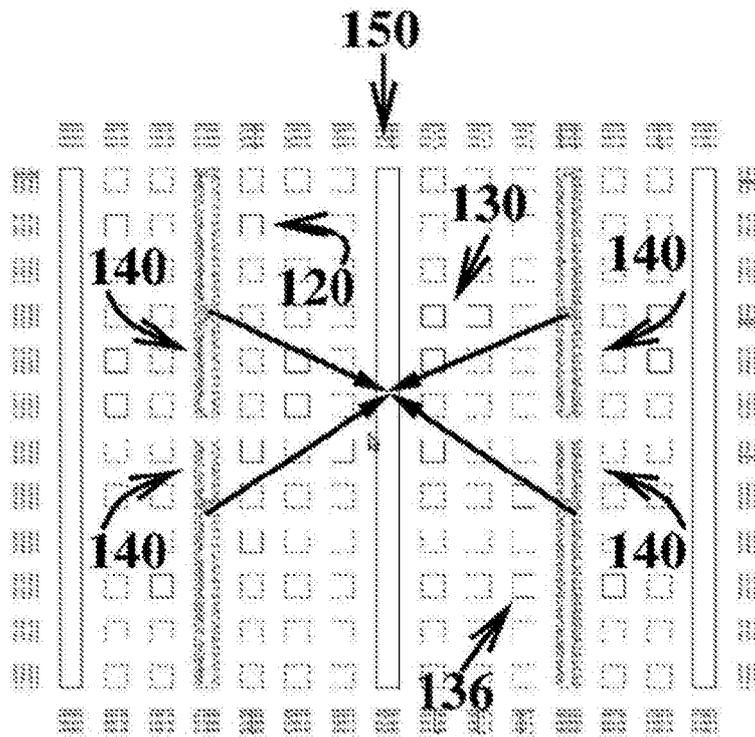


图4

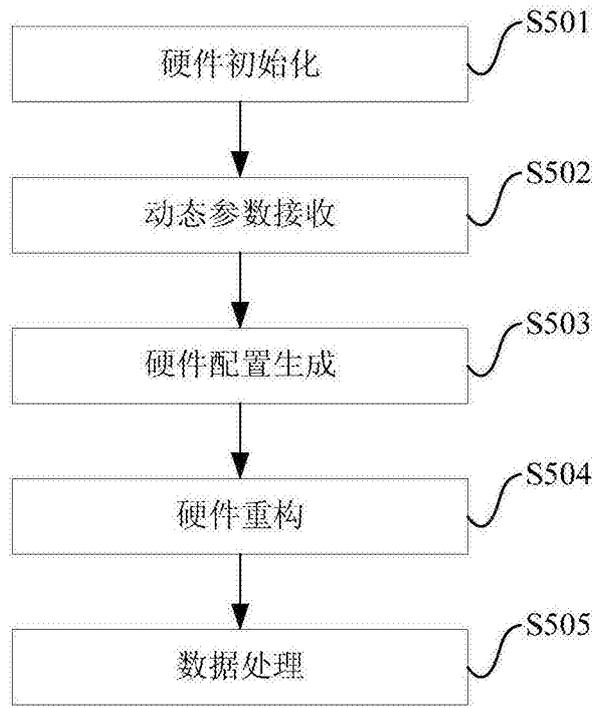


图5

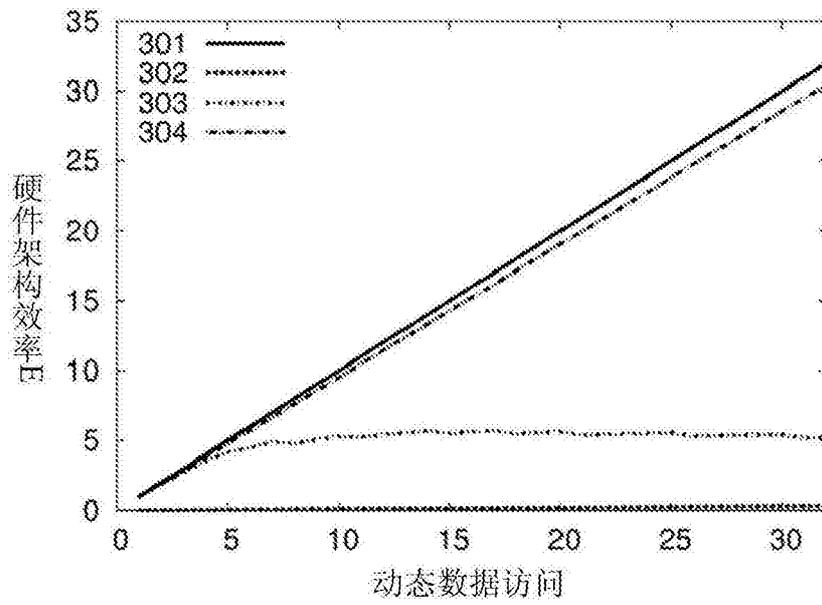


图6