



(12) 发明专利申请

(10) 申请公布号 CN 111782574 A

(43) 申请公布日 2020.10.16

(21) 申请号 202010676108.8

(22) 申请日 2020.07.14

(71) 申请人 北京四季豆信息技术有限公司
地址 102206 北京市昌平区回龙观东大街
338号创客广场B3-35-001

(72) 发明人 谢艳伟 阎斌 胡剑锋 张国松

(74) 专利代理机构 北京睿派知识产权代理事务
所(普通合伙) 11597

代理人 刘锋

(51) Int.Cl.

G06F 13/42 (2006.01)

G06F 13/38 (2006.01)

G06F 15/78 (2006.01)

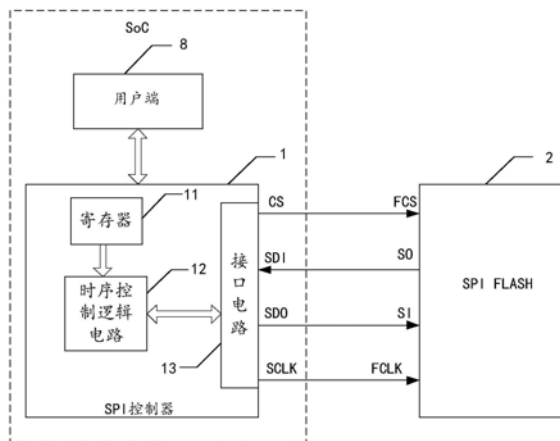
权利要求书1页 说明书9页 附图4页

(54) 发明名称

一种串行外设接口控制方法和串行外设接口控制器

(57) 摘要

本发明实施例公开了一种串行外设接口控制方法和串行外设接口控制器,本发明实施例的SPI控制器包括接口电路、寄存器和时序控制逻辑电路,通过寄存器获取可配置的寄存器数值,并以寄存器数值为触发值将指令转换为SPI FLASH需要的时序,SPI控制器的开发人员只需修改寄存器数值就能配置出不同SPI FLASH外接设备需要的时序,从而能够与市场上大多数厂家的SPI FLASH进行数据交换,不再需要为每个厂家的SPI FLASH开发专用的SPI控制器,由此,实现了一种通用的、高兼容的、可配置的SPI控制器。



1. 一种串行外设接口控制方法,其特征在于,所述方法包括:
获取系统时钟;
从寄存器获取时钟配置数据值和指令配置数据值;
根据时钟配置数据值将所述系统时钟调节为工作时钟;
获取指令,根据所述指令和所述指令配置数据值解析出与所述工作时钟对应的时序信号。
2. 根据权利要求1的方法,其特征在于,根据所述指令和所述指令配置数据值解析出与所述工作时钟对应的时序信号包括:
将所述指令解析为使能信号、协议时钟信号和输出的时序信号;以及
根据所述指令接收输入的时序信号。
3. 根据权利要求1的方法,其特征在于,将所述系统时钟调节为工作时钟包括:
调节所述系统时钟的极性;以及
调节所述系统时钟的时钟周期。
4. 一种串行外设接口控制器,其特征在于,所述控制器包括:
接口电路;
寄存器,用于存储时钟配置数据值和指令配置数据值;以及
时序控制逻辑电路,被配置为执行如下步骤:
通过所述接口电路获取系统时钟;
从寄存器获取时钟配置数据值和指令配置数据值;
根据时钟配置数据值将所述系统时钟调节为工作时钟;
通过所述接口电路获取指令,根据所述指令和所述指令配置数据值解析出与所述工作时钟对应的时序信号。
5. 根据权利要求4所述的串行外设接口控制器,其特征在于,所述接口电路包括时钟接口电路、输出接口电路、输入接口电路和使能接口电路。
6. 根据权利要求5所述的串行外设接口控制器,其特征在于,根据所述指令和所述指令配置数据值解析出与所述工作时钟对应的时序信号包括:
将所述指令解析为使能信号、协议时钟信号和输出的时序信号;以及
根据所述指令接收输入的时序信号。
7. 根据权利要求6所述的串行外设接口控制器,其特征在于,将所述指令解析为使能信号、协议时钟信号和输出的时序信号包括:
其中,所述协议时钟信号被配置为通过所述时钟接口电路输出;所述输入的时序信号或所述输出的时序信号被配置为通过所述输出接口电路输出;所述使能信号被配置为通过所述使能接口电路输出。
8. 根据权利要求6所述的串行外设接口控制器,其特征在于,根据所述指令接收输入的时序信号包括:
通过所述输入接口电路获取所述输入的时序信号。
9. 一种片上系统,其特征在于,包括:
如权利要求4-8中任一项所述的串行外设接口控制器。

一种串行外设接口控制方法和串行外设接口控制器

技术领域

[0001] 本发明涉及电子领域,具体涉及一种串行外设接口控制方法和串行外设接口控制器。

背景技术

[0002] SPI FLASH(SPI,Serial Peripheral Interface串行外设接口)是一种是内嵌SPI总线接口的串行FLASH,在市场上有很多型号,不同厂商的SPI FLASH控制方法存在差异,即使是相同厂家生产的不同型号的SPI FLASH,其控制方法也会存在差异。

[0003] SPI(串行外设接口,Serial Peripheral Interface)是一种通信协议。SPI协议以主从模式(Master Slave)工作,通常需要一个主设备(Master)和至少一个从设备(Slave)。

[0004] 片上系统广泛使用SPI FLASH作为外接设备,用于为片上系统装载程序或供片上系统读写数据。如果要在SPI FLASH上运行程序,其控制器必须是专用的,这就造成现有SPI控制器(也即串行外设即可控制器)很难做到兼容多种型号的SPI FLASH。

[0005] 对于不需要运行程序的SPI FLASH外接设备仅需使用SPI FLASH读写数据的应用场合,要想做到高兼容性,传统做法是使用通用SPI控制器,使用软件配置其寄存器,使用标准SPI控制器发送和接收数据。但是,这种做法会占用很多的CPU时间,且对于时序要求严格的SPI设备很难实现满足时序的要求。

[0006] 现有技术中,需要运行程序的SPI FLASH外接设备或使用SPI FLASH读写数据的应用场合,由于各厂家生产的SPI FLASH的指令系统不完全相同,需要为每一个SPI FLASH写一个专用的控制器。换一种类型的SPI FLASH就需要修改控制器,专用的SPI控制器灵活性差,在实际工作中,修改SPI控制器耗费大量的时间和精力。

发明内容

[0007] 有鉴于此,本发明的目的提供一种串行外设接口控制方法和串行外设接口控制器,以提高SPI控制器的通用性。

[0008] 第一方面,本发明实施例提供一种串行外设接口控制方法(SPI控制方法),包括:

[0009] 获取系统时钟;

[0010] 从寄存器获取时钟配置数据值和指令配置数据值;

[0011] 根据时钟配置数据值将所述系统时钟调节为工作时钟;

[0012] 获取指令,根据所述指令和所述指令配置数据值解析出与所述工作时钟对应的时序信号。

[0013] 优选地,根据所述指令和所述指令配置数据值解析出与所述工作时钟对应的时序信号包括:

[0014] 将所述指令解析为使能信号、协议时钟信号和输出的时序信号;以及

[0015] 根据所述指令接收输入的时序信号。

[0016] 优选地,将所述系统时钟调节为工作时钟包括:

- [0017] 调节所述系统时钟的极性;以及
- [0018] 调节所述系统时钟的时钟周期。
- [0019] 第二方面,本发明实施例提供串行外设接口控制器(SPI控制器),所述控制器包括:
- [0020] 接口电路;
- [0021] 寄存器,用于存储时钟配置数据值和指令配置数据值;以及
- [0022] 时序控制逻辑电路,被配置为执行如下步骤:
- [0023] 通过所述接口电路获取系统时钟;
- [0024] 从寄存器获取时钟配置数据值和指令配置数据值;
- [0025] 根据时钟配置数据值将所述系统时钟调节为工作时钟;
- [0026] 通过所述接口电路获取指令,根据所述指令和所述指令配置数据值解析出与所述工作时钟对应的时序信号。
- [0027] 优选地,所述接口电路包括时钟接口电路、输出接口电路、输入接口电路和使能接口电路。
- [0028] 优选地,根据所述指令和所述指令配置数据值解析出与所述工作时钟对应的时序信号包括:
- [0029] 将所述指令解析为使能信号、协议时钟信号和输出的时序信号;以及
- [0030] 根据所述指令接收输入的时序信号。
- [0031] 优选地,将所述指令解析为使能信号、协议时钟信号和输出的时序信号包括:
- [0032] 其中,所述协议时钟信号被配置为通过所述时钟接口电路输出;所述输入的时序信号或所述输出的时序信号被配置为通过所述输出接口电路输出;所述使能信号被配置为通过所述使能接口电路输出。
- [0033] 优选地,根据所述指令接收输入的时序信号包括:
- [0034] 通过所述输入接口电路获取所述输入的时序信号。
- [0035] 本发明实施例的SPI控制器包括接口电路、寄存器和时序控制逻辑电路,通过寄存器获取可配置的寄存器数值,并以寄存器数值为触发值将指令转换为SPI FLASH需要的时序,SPI控制器的开发人员只需修改寄存器数值就能配置出不同SPI FLASH外接设备需要的时序,从而能够与市场上大多数厂家的SPI FLASH进行数据交换,不再需要为每个厂家的SPI FLASH开发专用的SPI控制器,由此,实现了一种通用的、高兼容的、可配置的SPI控制器。从而解决了专用SPI控制器只能与特定厂家的SPI FLASH进行通信的不便之处。通过本发明实施例的SPI控制器,只需修改寄存器的配置数据值就能兼容市场上大多数厂家的SPI FLASH外接设备,大大缩短了SPI控制器的开发时间,避免了重复开发专用SPI控制器。

附图说明

- [0036] 通过以下参照附图对本发明实施例的描述,本发明的上述以及其它目的、特征和优点将更为清楚,在附图中:
- [0037] 图1是片上系统SoC的示意图;
- [0038] 图2是SPI FLASH的引脚示意图;
- [0039] 图3是本发明实施例的SPI控制器通信系统示意图;

[0040] 图4是本发明实施例的SPI控制器与SPI FLASH进行数据交换的时序图；

[0041] 图5是本发明实施例SPI控制器的控制方法的流程图。

具体实施方式

[0042] 以下基于实施例对本发明进行描述,但是本发明并不仅仅限于这些实施例。在下文对本发明的细节描述中,详尽描述了一些特定的细节部分。对本领域技术人员来说没有这些细节部分的描述也可以完全理解本发明。为了避免混淆本发明的实质,公知的方法、过程、流程、元件和电路并没有详细叙述。

[0043] 此外,本领域普通技术人员应当理解,在此提供的附图都是为了说明的目的,并且附图不一定是按比例绘制的。

[0044] 同时,应当理解,在以下的描述中,“电路”是指由至少一个元件或子电路通过电气连接或电磁连接构成的导电回路。当称元件或电路“连接到”另一元件或称元件/电路“连接在”两个节点之间时,它可以是直接耦接或连接到另一元件或者可以存在中间元件,元件之间的连接可以是物理上的、逻辑上的、或者其结合。相反,当称元件“直接耦接到”或“直接连接到”另一元件时,意味着两者不存在中间元件。

[0045] 除非上下文明确要求,否则在说明书的“包括”、“包含”等类似词语应当解释为包含的含义而不是排他或穷举的含义;也就是说,是“包括但不限于”的含义。

[0046] 在本发明的描述中,需要理解的是,术语“第一”、“第二”等仅用于描述目的,而不能理解为指示或暗示相对重要性。此外,在本发明的描述中,除非另有说明,“多个”的含义是两个或两个以上。

[0047] 图1是片上系统SoC的示意图。

[0048] 参照图1,片上系统(SoC, System on a chip)通常包括CPU3、SPI控制器1和存储空间4。CPU3通过片上总线6与SPI控制器1和存储空间4通信连接。片上总线6可以是AHB总线(Advanced High performance Bus),存储空间4可以是DMA(Direct Memory Access,直接存储器访问),其中SPI控制器1通过SPI总线7与外接设备SPI FLASH2通信。在片上系统中, SPI控制器1通过片上总线6将CPU3传输的并行数据转换为串行数据,与外接设备SPI FLASH2进行数据交换;并把从机SPI FLASH2传输的串行数据转换成并行数据,通过片上总线6发送给CPU3。SPI控制器1会在被片上系统选中的情况下与片上系统进行数据的交换。

[0049] SPI FLASH2是一种非易失性、可擦除的存储器。SPI FLASH2广泛应用于手机、平板电脑等手持设备中,通过将数据存储在SPI FLASH2中,手机、平板电脑等手持设备可以为使用者提供较大的存储空间。

[0050] 由于不同外接设备厂商生产的SPI FLASH2的指令系统不完全相同,当SPI FLASH2被选中时,就要将对应的指令转换为对应SPI FLASH2需要的时序才能进行数据交换。在现有技术中,当主设备需要与SPI FLASH2(从设备)通信时,就需要为特定厂家的SPI FLASH2(从设备)定制专用的SPI控制器(主设备),这种定制的、专用的控制器灵活性较差。现有技术的另一中实现方式是通过软件的方式开发SPI控制器1(虚拟主设备),但是软件SPI控制器1的驱动程序占用CPU3资源,且难以满足SPI FLASH2(从设备)的时序要求。为了解决专用SPI控制器1(主设备)灵活性差以及软件SPI控制器1(虚拟主设备)难以满足多个厂家的SPI FLASH2(从设备)时序要求的问题,以及,开发人员不得不为每个SPI FLASH2开发特定的SPI

控制器1, 耗费精力的问题, 本发明实施例公开了一种通用型、高兼容以及可配置SPI控制器1 (主设备)。

[0051] 下面结合图2对SPI FLASH的引脚做详细介绍。

[0052] 图2是SPI FLASH的引脚示意图。SPI FLASH是内嵌SPI总线接口的串行FLASH。其中, SPI (串行外设接口, Serial Peripheral Interface) 是一种通信协议; FLASH也称为闪存存储器, 是一种非易失性存储器。SPI协议以主从模式 (Master Slave) 工作, 通常需要一个主设备 (Master) 和至少一个从设备 (Slave)。

[0053] 对于一个示例性的SPI Flash芯片, 引脚VCC和引脚VSS分别为电源和地, 用于为SPI FLASH供电。写保护引脚WP和挂起引脚HOLD用于数据保护和空闲模式的低功耗运行。FCS、S0、SI和FCLK是用于与主设备SPI控制器通信的引脚。其中, FCS为SPI FLASH的片选信号, 低电平有效, 片选信号由主设备提供; S0为SPI FLASH的串行数据输出引脚, 数据在时钟的下降沿由FLASH输出到主设备SPI控制器; SI为SPI FLASH的串行数据输入引脚, 用于接收主设备SPI控制器的传输指令、地址和数据, 在上升沿锁存到SPI FLASH器件中; FCLK为SPI FLASH的串行时钟引脚, 通常由主设备提供串行时钟信号, 并将主设备的串行时钟信号接入到SPI FLASH的串行时钟引脚。不同厂家的SPI FLASH的指令系统不完全相同, 例如, 片选信号通常是低电平有效, 在两个指令之间, 为了区别下一个指令的到来, 通常将片选信号拉高并持续若干个时钟周期后, 将片选信号再次拉低, 由此, 专用的SPI控制器发送的片选信号的时序通常是根据特定厂家的SPI FLASH的参数来确定的, 一经确定, 就会制作成特定的硬件电路, 对应到芯片上是逻辑门组成的逻辑。通过制造成芯片之后不可修改。因此, 现有的SPI控制器只能适应特定的厂家的SPI FLASH。不同的片上系统会用到不同厂家的SPI FLASH, 就需要开发人员重新开发SPI控制器。除了片选信号, 对应地, 不同厂家会有特定的时序要求, 类似地, 不同厂家的SPI FLASH对时钟周期的要求也可能是不同的, 专用的SPI控制器发送的时钟周期对应的时序信号也是固定的, 一经确定就会制作成硬件电路, 对应到芯片上是逻辑门组成的逻辑。通过制造成芯片之后不可修改。因此, 专用的SPI控制器只能适应特定厂家的SPI FLASH。

[0054] SPI控制器通过SPI总线与SPI FLASH连接。SPI总线需要4根通信线, 分别是MISO、MOSI、SCLK、和CS。其中, MISO用于主设备输入从设备输出; MOSI用于主设备输出从设备输入; SCLK为时钟信号, 由主设备提供; CS为从设备使能信号, 由主设备控制。

[0055] 为了使得SPI控制器能够适应多种不同的SPI FLASH外接设备, 就需要SPI控制器能够根据不同SPI FLASH厂家的参数, 通过MISO、MOSI、SCLK、和CS四根通信线输出特定的时序信号到SPI FLASH对应的引脚上, SPI FLASH收到正确的时序信号, 才能响应主设备SPI控制器, 从而与主设备SPI控制器进行数据交换。

[0056] 本发明实施例公开了一种可配置的SPI控制器, 通过对SPI控制器进行配置, 以配置数据作为触发值, 从而能够在MISO、MOSI、SCLK、和CS四根通信线输入或输出特定的SPI FLASH需要的时序, 从而能够与市场上绝大多数SPI FLASH交换数据。

[0057] 图3是本发明实施例的SPI控制器通信系统示意图。

[0058] 参照图3, 将片上系统中的CPU、存储空间简化为用户端8。本实施例的SPI控制器1在被用户端8选中时, 通过SPI总线与SPI FLASH2进行通信。其中, SPI总线在SPI控制器1主设备端与从设备SPI FLASH2的引脚对应关系如表1所示:

[0059]	SPI总线	SPI控制器引脚	SPI FLASH的引脚
	MISO	SDI	SO
	MOSI	SDO	SI
	SCLK	SCLK	FCLK
	CS	CS	FCS

[0060] 表1

[0061] 本实施例的SPI控制器1包括接口电路13、寄存器11和时序控制逻辑电路12。其中，接口电路13包括时钟接口电路SCLK、输出接口电路SDO、输入接口电路SDI和使能接口电路CS。

[0062] 本实施例的寄存器11是一种参数配置寄存器，被配置为获取时钟配置数据值和指令配置数据值。时钟配置数据值包括时钟周期、时钟极性等；指令配置数据值包括时钟相位、连续写入的数据长度；参数配置寄存器还包括模式选择寄存器。

[0063] 例如，可以设置以下寄存器作为参数配置寄存器：

[0064] LENGTH寄存器，用于接收可配置的数据长度，例如，对于需要连续写入的情况，可以配置总数据长度，例如256比特或1024比特。

[0065] LATENCY寄存器，用于接收可配置的时钟相位，例如，在SPI控制器2的CS有效后，时钟上升沿到来所需的SCLK的周期数

[0066] DIV寄存器，用于接收可配置的分频数，例如：2, 4, 8分频。

[0067] MODE寄存器，用于配置SPI控制器的工作模式，例如：1为SPI总线的工作模式，0为I2C(双向二线制同步串行总线, Inter Integrated Circuit BUS)的工作模式。

[0068] CKP寄存器，用于配置SPI控制器的时钟极性，例如：1为上升沿采样，0为下降沿采样。

[0069] 下面以SPI控制器1输出到SCLK引脚的时钟周期时序信号为例来说明本实施例的SPI控制器是如何通过设置寄存器作为参数配置寄存器，以参数配置寄存器中的数值作为触发值，从而，使得本实施例的SPI控制器1能够在对应的接口电路输出特定的SPI FLASH需要的时钟周期(时序信号)，当更换了另一个厂家的SPI FLASH设备，只需修改参数配置寄存器，就能输出另一厂家的SPI FLASH需要的时钟周期。以此类推，只需修改参数配置寄存器的值，就能在SPI总线上生成另一厂家的SPI FLASH需要的时序信号，从而使得本实施例的SPI控制器能够适应市场不同厂家的SPI FLASH设备。

[0070] SPI控制器获取系统时钟，SPI控制器通过时序控制逻辑电路12获取寄存器11中的时钟配置数据，从而使得SPI控制器1能够根据系统时钟调节为工作时钟，并通过接口电路13输出到SPI FLASH2对应的总线。在一种可选的实现方式中，本实施例的SPI控制器1能够通过时序控制逻辑电路12获取用户端8的系统时钟。具体，时序控制逻辑电路12可以通过片上系统总线获取用户端8的系统时钟。

[0071] 根据时钟配置数据值将所述系统时钟调节为工作时钟，也即，根据寄存器的配置数值，对系统时钟的时钟周期进行调节，从而在SCLK端口上输出某厂家SPI FLASH2能够响应的时钟周期。

[0072] 获取系统时钟后，在一种可选的实现方式中，SPI控制器1可以获取DIV寄存器(用于接收可配置的分频数，例如：2, 4, 8分频)中的配置数据值，以DIV寄存器中的配置数据值

作为触发条件,当达到触发条件时,对系统时钟的频率转换为工作时钟频率(频率为周期的倒数)。具体地,通过DIV寄存器获取分频数,从而将系统时钟的时钟周期经分频后产生工作时钟的时钟周期。

[0073] SPI控制器1和SPI FLASH2只有在相同的时钟周期下才能进行串行数据的交换,不同厂家的SPI FLASH2最高工作频率可能不同。SPI控制器1通过片上系统获取的系统时钟通常是200MHZ,然而,外接设备SPI FLASH2无法响应200MHZ的频率(频率过高),为了与SPI FLASH2通信,时序控制逻辑电路12需要根据寄存器11中的时钟配置数据值将系统时钟的200MHZ进行分频。也即,以DIV寄存器中的配置数据值作为触发条件,当达到触发条件时,对系统时钟的频率转换为工作时钟频率,从而在SCLK接口上给出能够另SPI FLASH2正常工作的时钟周期。

[0074] 具体地,时序控制逻辑电路12获取寄存器11中的DIV寄存器中的数值。响应于为DIV寄存器中的数值2,将工作时钟调整为100MHZ;响应于DIV寄存器中的数值为4,将工作时钟调整为50MHZ;以此类推,响应于DIV寄存器中的数值为8,将工作时钟调整为25MHZ。

[0075] 在一种可选的实现方式中,可以通过以下电路描述语言来实现:

```
[0076] Module spi_controllor (clk,rstn,sclk,sdo,sdi,cs); //定义一个SPI控制器
```

```
[0077] Input clk; //获取系统时钟200MHZ
```

```
[0078] Input rstn; //获取复位信号
```

```
[0079] Output sclk; //输出sclk引脚输出SPIFLASH的工作时钟
```

```
[0080] Output sdo; //通过sdo引脚输出指令
```

```
[0081] Output cs; //通过cs引脚输出SPIFLASH的使能位
```

```
[0082] Input sdi; //通过sdi引脚接收SPIFLASH的数据
```

[0083] 根据参数寄存器DIV的时钟配置数据值,对系统时钟分频,从而产生工作时钟。DIV可以接收的时钟配置数据值为2,4和8,如果DIV为2,对系统时钟2分频;如果DIV为4,对系统时钟4分频;如果DIV为8,对系统时钟8分频。假设特定厂家的SPI FLASH的工作时钟为50MHZ,此时系统时钟为200MHZ,通过计算可得,需要对系统进行4分频。可以通过以下电路描述语言来实现:

```
[0084] //通过寄存器定义一个计数器,计数器的初始值为0,最大值为3,
```

```
[0085] Reg DIV[3:0] //定义寄存器,用于接收可配置的分频数
```

```
[0086] reg[3:0] cnt; //定义一个计数器
```

```
[0087] always@(posedge clk or negedge rstn) //以系统时钟的上升沿或复位信号的下降沿作为触发条件
```

```
[0088] begin
```

```
[0089] if (!rstn) //如果spi_controllor被复位
```

```
[0090] cnt<=1'b0; //将计数器清零
```

```
[0091] else if (cnt==div) //当计数器计数到最大值时,对计数器清零
```

```
[0092] cnt<=1'b0; //计数器的初始值为0
```

```
[0093] else
```

```
[0094] clko<=cnt+1; //计数器每获得一个系统时钟的上升沿加一
```

```
[0095] end
```


[0096] 通过寄存器DIV定义的计数器CNT作为工作时钟反转的触发条件,从而获得分频后的工作时钟CLK0。可以通过以下电路描述语言来实现:

```
[0097] reg clko;//定义工作时钟
```

```
[0098] always@(posedge clk or negedge rstn)//以系统时钟的上升沿或复位信号的下降沿作为触发条件
```

```
[0099] begin
```

```
[0100] if(!rstn)//如果spi_controllor被复位
```

```
[0101] clko<=1'b0;//工作时钟的初始值为0
```

```
[0102] else if cnt==div//当计数器计数到4时,
```

```
[0103] clko<=~clko;对工作时钟取一次反
```

```
[0104] end
```

[0105] 通过获取DIV寄存器的数值,以DIV寄存器的数值作为触发值,响应这一触发值,生成工作时钟,工作时钟具备特定SPI FLASH2需要的时钟周期,由此,本实施例的SPI控制器1能够在SCLK接口上给出特定SPI FLASH2需要的时钟周期。假设,跟换了另一个厂家的SPI FLASH2,需要不同的时钟周期,通过本实施例SPI控制器1,无需重新发开一个全新的SPI控制器1,只需修改DIV寄存器的数值,就可以产生另一个时钟周期。类似的,为了适应全新的SPI FLASH2在其他引脚对时序的特殊要求,只需修改其他寄存器的数值,并相应对应的寄存器数值,在对应的引脚生成正确的时序,就能与全新的SPI FLASH2交换数据。

[0106] 在一种可选的实现方式中,可以定义以下寄存器,实现对其他引脚时序信号的调整,例如:

```
[0107] Reg LENGTH[7:0]//定义寄存器,用于接收可配置的数据长度。
```

```
[0108] Reg LATENCY[2:0]//定义寄存器,用于接收可配置的时钟相位。
```

```
[0109] Reg MODE[1:0]//定义寄存器,用于配置SPI控制器的工作模式。
```

```
[0110] Reg CKP[7:0]//定义寄存器,用于配置SPI控制器的时钟极性,例如:1为上升沿采样,0为下降沿采样。
```

[0111] 类似地,对于采样时刻是上升沿还是下降沿,具体地,可以定义CKP寄存器,通过CKP寄存器的配置数据值作为触发条件,从而可以改变SPI控制器的采样极性。

[0112] 本实施例的时序控制逻辑电路12获取系统时钟,根据寄存器11将系统时钟按时钟配置数据转换为工作时钟,接下来,将用户端8的通信指令解析为与工作时钟对应的时序信号,并通过接口电路13发送或接收对应的时序信号,从而能够与外接设备SPI FLASH2交换数据。

[0113] 本实施例的SPI控制器1通过改变寄存器11的时钟配置数据值和指令配置数据值就可以提供不同SPI FLASH2厂家的正常工作的时序信号,由此,不需要为特定的厂家开发专用的SPI控制器1,只需要修改寄存器就能适应不同SPI FLASH2厂家的工作时序。

[0114] 获取指令,根据所述指令和所述指令配置数据值解析出与所述工作时钟对应的时序信号。将所述指令解析为使能信号、协议时钟信号和输出的时序信号;以及根据所述指令接收输入的时序信号。

[0115] 下面以读指令为例说明根据所述指令和所述指令配置数据值解析出与所述工作时钟对应的时序信号。

[0116] 图4是本发明实施例的SPI控制器与SPI FLASH进行数据交换的时序图。

[0117] 本实施例的时序控制逻辑电路12能够通过用户端8获取指令。时序控制逻辑电路12根据所述指令和所述指令配置数据值解析出与所述工作时钟对应的时序信号包括将所述指令解析为使能信号、协议时钟信号和输出的时序信号。其中,所述协议时钟信号被配置为通过所述时钟接口电路输出;所述输入的时序信号或所述输出的时序信号被配置为通过所述输出接口电路输出;所述使能信号被配置为通过所述使能接口电路输出。以及根据所述指令接收输入的时序信号。通过所述输入接口电路获取所述输入的时序信号。

[0118] 具体地,SPI控制器1的CS引脚接SPI FLASH2的FCS引脚,时序控制逻辑电路12通过CS引脚发送使能信号。SPI控制器1的SCLK引脚接SPI FLASH2的FCLK引脚,时序控制逻辑电路12通过SCLK发送工作时钟信号。SPI控制器1的SDI引脚接SPI FLASH2的S0引脚,时序控制逻辑电路12通过SDI引脚发送指令和地址。SPI控制器1的SDI用于接SPI FLASH2的S0引脚,时序控制逻辑电路12通过SDI接收SPI FLASH2的反馈数据。其中,使能信号低电平有效,通过本实施例的寄存器11可以配置为低有效;工作时钟也能够通过本实施例的寄存器11调节时钟周期;发送指令和地址能够通过本实施例的寄存器11配置为上升沿采样,反馈数据能够通过本实施例的寄存器11配置为上升沿采样或发送指令和地址能够通过本实施例的寄存器11配置为下降沿采样,反馈数据能够通过本实施例的寄存器11配置为下降沿采样。

[0119] 参照图4,SPI控制器与SPI FLASH通信,首先将使能信号CS拉低。

[0120] 时序控制逻辑电路工作时钟信号通过接口电路CLK发送到SCLK引脚,时序控制逻辑电路在工作时钟信号CLK的上升沿通过SD0引脚发送读指令和地址。

[0121] 具体地,S1是指SPI控制器在第0个时钟周期到第7个时钟周期的上升沿通过SD0接口发送读指令,S2是指SPI控制器在第8个时钟周期到第31个时钟周期的上升沿通过SD0接口发送地址。

[0122] SPI FLASH通过SI引脚接收到读指令和地址后,将所述地址中的数据通过S0引脚反馈给SPI控制器的SDI引脚,SPI控制器的SDI在工作时钟的上升沿对反馈数据进行采样。

[0123] 具体地,S3是指SPI控制器在第32个时钟周期到第39个时钟周期的上升沿,通过SPI控制器的SDI引脚接收SPI FLASH反馈的数据。

[0124] 通过S1、S2和S3,SPI控制器实现了一次与SPI FLASH的数据交换。

[0125] 在SPI控制器与SPI FLASH通信时,使能信号高有效或低有效,工作时钟的频率、发送指令和地址能够通过本实施例的寄存器11配置为上升沿采样或下降沿采样,反馈数据能够通过本实施例的寄存器11配置为上升沿采样或下降沿采样。也就是说,具体的参数都是可以通过寄存器配置,以寄存器中的数值作为触发信号,从而获得不同厂家的SPI FLASH的时序信号。由此,本实施例能够提供一种通用的、可配置的、高兼容的SPI控制器。

[0126] 图5是本发明实施例SPI控制器的控制方法的流程图。参照图5,本实施例的控制方法包括:

[0127] 步骤100:获取系统时钟。

[0128] 在一种可选的实现方式中,SPI控制器能够通过片上总线获取系统时钟。

[0129] 步骤200:从寄存器获取时钟配置数据值和指令配置数据值。

[0130] 在一种可选的实现方式中,SPI控制器能够通过寄存器获取数值,例如获取时钟配置数据值和指令配置数据值。

[0131] 步骤300:根据时钟配置数据值将所述系统时钟调节为工作时钟;调节所述系统时钟的极性;以及调节所述系统时钟的时钟周期。

[0132] 在一种可选的实现方式中,可以获取时钟配置数据值,根据时钟数据值对系统时钟进行分频从而调节所述系统时钟的时钟周期,也即,将系统时钟的时钟频率调节为工作时钟的时钟频率。可以将系统时钟的上升沿或下降沿作为触发条件,从而调节系统时钟的极性。

[0133] 步骤400:获取指令,根据所述指令和所述指令配置数据值解析出与所述工作时钟对应的时序信号。将所述指令解析为使能信号、协议时钟信号和输出的时序信号;以及根据所述指令接收输入的时序信号。

[0134] 本实施例的SPI控制方法,通过从寄存器获取时钟配置数据值和指令配置数据值,将系统时钟为工作时钟,由此,能够在根据系统时钟将指令解析为工作时钟对应的实现信号,通过将指令解析为使能信号、协议时钟信号和输出的时序信号,并通过接收输入的时序信号从而获取对应的数据。本发明实施的SPI控制方法能够根据寄存器获取时钟配置数据值和指令配置数据值,从而能够适应不同厂家的SPI FLASH,由此,解决了传统的SPI控制器只能与特定厂家的SPI FLASH通信的技术难题;本实施例SPI控制器只有在片上系统选中时才进入工作状态,由此克服了通过软件的方式开发的SPI控制器占用CPU资源的技术问题。

[0135] 以上所述仅为本发明的优选实施例,并不用于限制本发明,对于本领域技术人员而言,本发明可以有各种改动和变化。凡在本发明的精神和原理之内所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

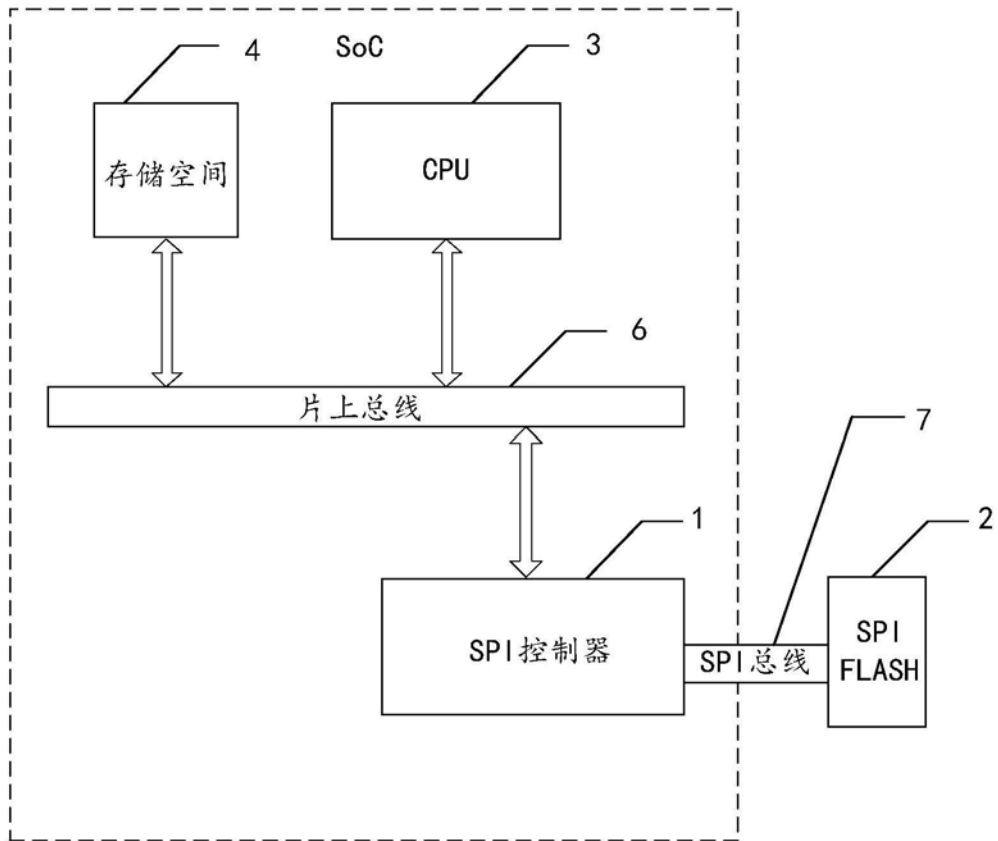


图1

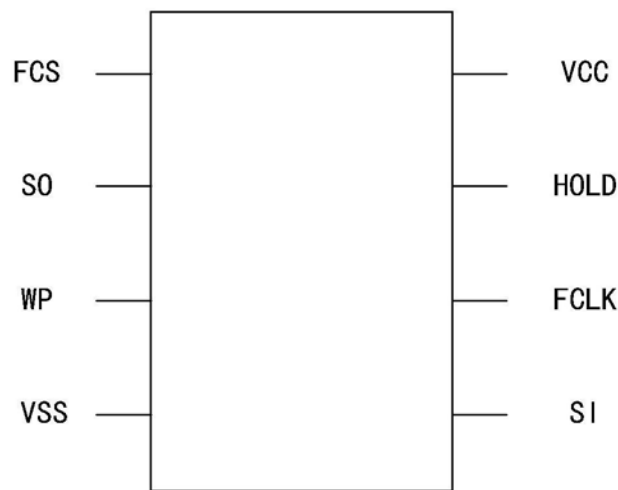


图2

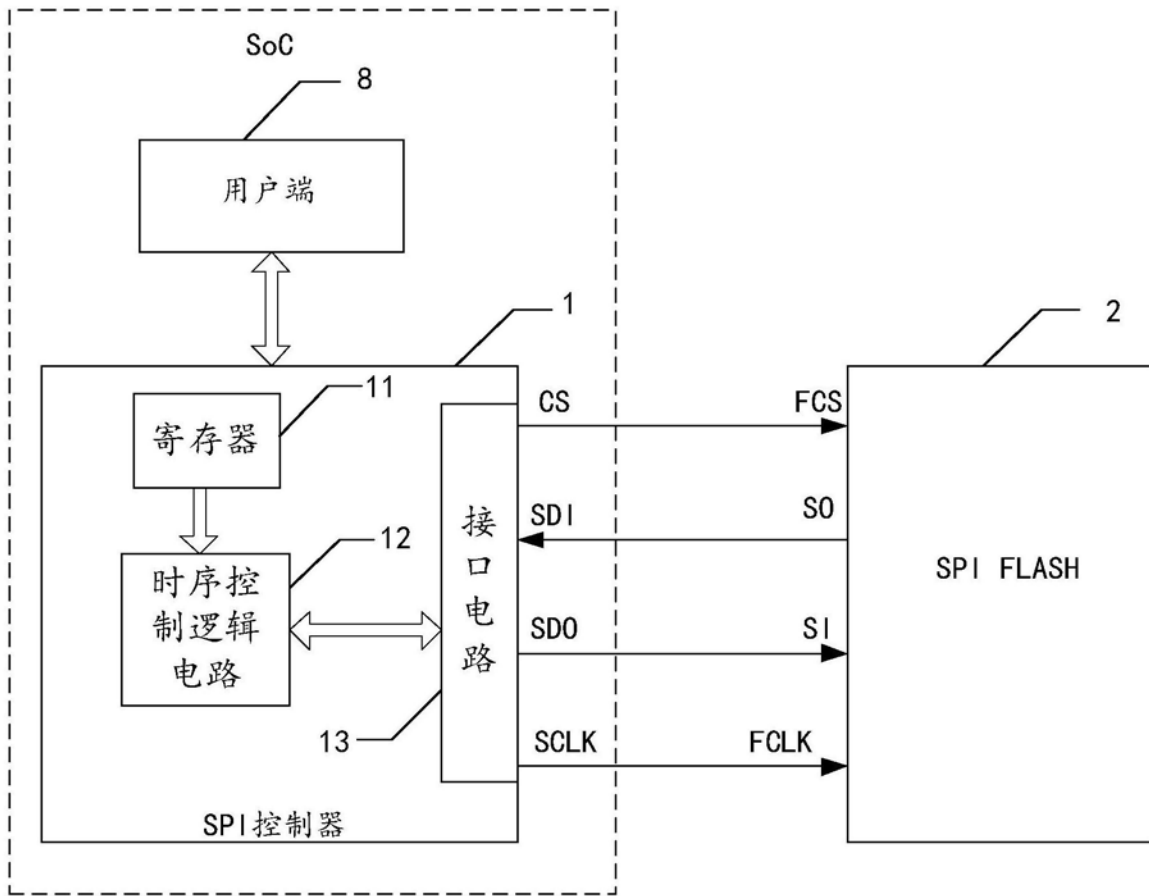


图3

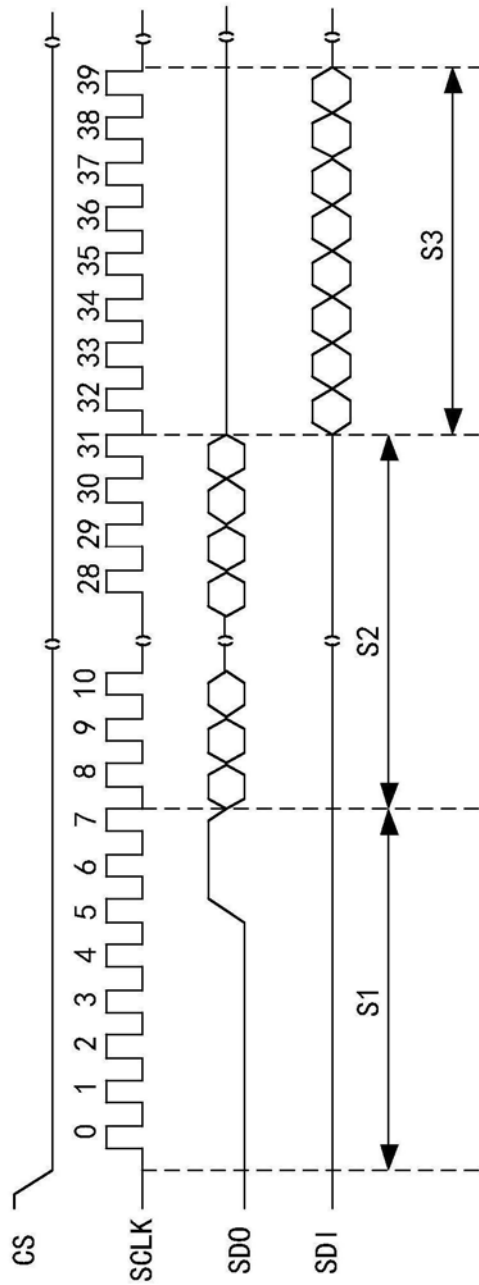


图4

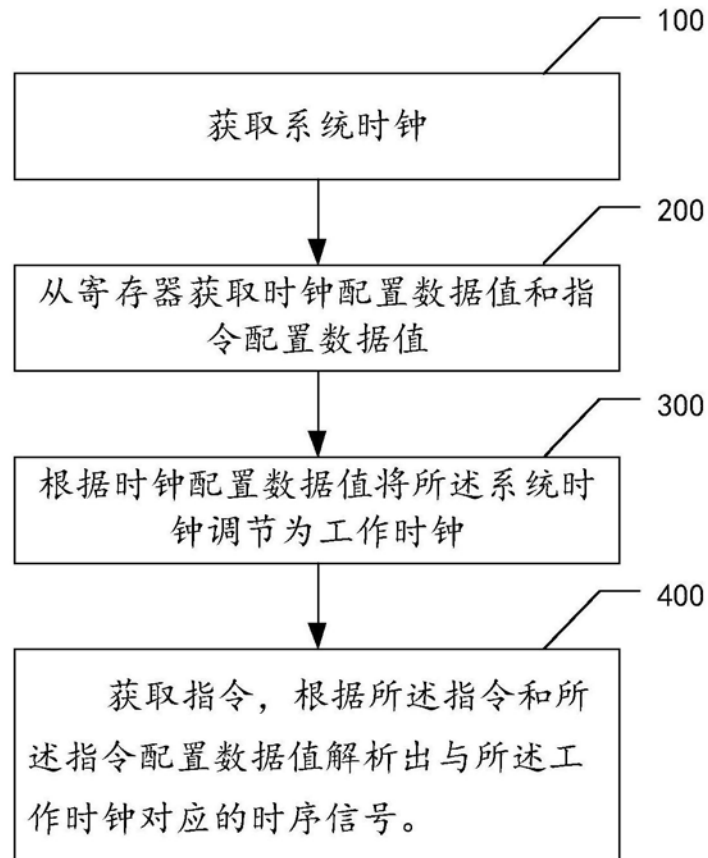


图5