



(12)发明专利

(10)授权公告号 CN 104272271 B

(45)授权公告日 2017.12.26

(21)申请号 201380023766.5

(22)申请日 2013.05.07

(65)同一申请的已公布的文献号

申请公布号 CN 104272271 A

(43)申请公布日 2015.01.07

(30)优先权数据

61/643,690 2012.05.07 US

13/888,357 2013.05.06 US

(85)PCT国际申请进入国家阶段日

2014.11.05

(86)PCT国际申请的申请数据

PCT/US2013/039944 2013.05.07

(87)PCT国际申请的公布数据

W02013/169773 EN 2013.11.14

(73)专利权人 密克罗奇普技术公司

地址 美国亚利桑那州

(72)发明人 凯文·基尔策 贾斯廷·米尔克斯

孙达尔·巴拉苏布拉马尼亚安

托马斯·爱德华·佩尔门

库沙拉·贾瓦盖尔

(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 沈锦华

(51)Int.Cl.

G06F 11/36(2006.01)

(56)对比文件

EP 0869434 A2,1998.10.07,说明书第1页
第10行到第64页第55行、图1-16.

US 2012/0095728 A1,2012.04.19,说明书第
[0002]-[0113]段、图1-8C.

JP 昭63-292246 A,1988.11.29,说明书第1
页左栏第3段到第5页右栏第3段、图1-9.

审查员 李维

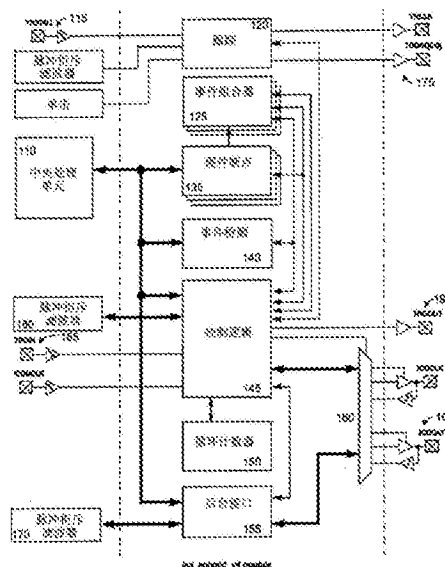
权利要求书2页 说明书15页 附图10页

(54)发明名称

具有指令跟踪能力的处理器装置

(57)摘要

本发明涉及一种具有调试能力的处理器装置,其具有:中央处理单元;调试电路,其包含跟踪模块及外部接口,其中所述跟踪模块产生包含关于所执行指令的信息的跟踪流,其中所述跟踪流通过所述外部接口输出,且其中所述跟踪模块进一步可操作以检测触发信号且在检测之后即刻将跟踪包插入到所述所产生跟踪流中。



1. 一种具有调试能力的处理器装置,其包括:

中央处理单元;

调试电路,其包含跟踪模块及外部接口;

其中所述跟踪模块产生包含关于所执行指令的信息的跟踪流,其中所述跟踪流通过所述外部接口输出;

且其中所述跟踪模块进一步可操作以检测异步触发信号且在检测之后即刻将所述异步触发信号与系统时钟进行同步且在对所述异步触发信号进行同步时产生跟踪触发包并将跟踪包插入到所产生的跟踪流中,

其中所述异步触发信号是通过所述处理器装置的外部引脚来施加的触发信号。

2. 根据权利要求1所述的处理器装置,其中通过内部事件产生额外触发信号。

3. 根据权利要求2所述的处理器装置,其中所述内部事件包括以下各项中的至少一者:断点、主清除、中断、异常执行条件、调试器溢出条件。

4. 根据权利要求1所述的处理器装置,其中所述跟踪流是基于包的。

5. 根据权利要求1所述的处理器装置,其中所述跟踪包包含关于触发源的信息。

6. 根据权利要求5所述的处理器装置,其中有条件地提供所述信息,其中所述条件可为用户定义的。

7. 根据权利要求1所述的处理器装置,其中所述处理器装置具有多个功能引脚且一个引脚可经指派以作为所述异步触发信号的触发输入操作。

8. 根据权利要求1所述的处理器装置,其中在接收所述异步触发信号时在所述跟踪流中与所报告指令的执行时间一致的点将所述跟踪包插入到所述跟踪流中。

9. 根据权利要求1所述的处理器装置,其中所述调试电路进一步包括可操作以组合多个事件的事件组合器,其中可至少通过以下各项中的一者形成事件:断点及所述异步触发信号。

10. 根据权利要求9所述的处理器装置,其中所述跟踪包识别导致事件的所述断点。

11. 根据权利要求1所述的处理器装置,其进一步包括与接收所述异步触发信号的外部引脚耦合的噪声拒斥滤波器。

12. 根据权利要求1所述的处理器装置,其中所述跟踪模块可编程以检测所述异步触发信号的正缘或负缘。

13. 一种用于在处理器装置内调试所执行代码的方法,其包括:

由中央处理单元CPU执行代码;

产生由所述CPU执行的指令跟踪流;

通过所述处理器装置的外部引脚接收异步触发信号;

在确定所述异步触发信号之后,即刻将所述异步触发信号与系统时钟进行同步;

在对所述异步触发信号进行同步时产生识别所述异步触发信号的跟踪触发包;以及将识别异步触发信号的跟踪包插入到所述跟踪流中。

14. 根据权利要求13所述的方法,其中通过内部事件产生附加触发信号。

15. 根据权利要求14所述的方法,其中所述内部事件包括以下各项中的至少一者:断点、主清除、中断、异常执行条件、调试器溢出条件。

16. 根据权利要求13所述的方法,其进一步包括在请求之后即刻通过调试接口输出所

述跟踪流。

17. 根据权利要求13所述的方法,其中在接收所述异步触发信号时在所述跟踪流中与所报告指令的执行时间一致的点将所述跟踪包插入到所述跟踪流中。

18. 根据权利要求13所述的方法,其中所述跟踪流是基于包的。

19. 根据权利要求13所述的方法,其中跟踪包包含关于触发源的信息。

20. 根据权利要求19所述的方法,其中有条件地提供所述信息,其中所述条件可为用户定义的。

21. 根据权利要求13所述的方法,其进一步包括:

至少通过以下各项中的一者形成事件:断点及所述异步触发信号;

组合多个事件以产生调试功能。

22. 根据权利要求21所述的方法,其中所述跟踪包识别导致事件的所述断点。

23. 根据权利要求22所述的方法,其中所述调试功能为触发输出信号或停止。

24. 根据权利要求13所述的方法,其进一步包括通过与外部引脚耦合的噪声拒斥滤波器对所述异步触发信号进行滤波。

25. 根据权利要求13所述的方法,其进一步包括编程所述处理器装置中的跟踪模块以检测所述异步触发信号的正缘或负缘。

具有指令跟踪能力的处理器装置

[0001] 相关申请案的交叉参考

[0002] 本申请案主张于2012年5月7日提出申请、标题为“PROCESSOR DEVICE WITH INSTRUCTION TRACE CAPABILITIES”的第61/643,690号美国临时申请案的权益,所述美国临时申请案的全文并入本文中。

技术领域

[0003] 本发明涉及处理器装置,特定来说具有集成调试能力的微控制器装置。

背景技术

[0004] 现代微处理器及微控制器包含高效地允许嵌入式开发工具借助于所谓的电路内调试器或仿真器装置分析运行程序的电路。为此,微控制器或微处理器提供支持调试功能的内部电路及可(举例来说)通过编程装置的多个功能引脚以操作为调试接口来启动的特定接口。此接口通常可配置为高速串行接口以允许实际装置与外部调试器或仿真器之间的快速通信。装置自身因此可在不启动此接口的正常操作模式中操作且相关联引脚可用于其它目的及使用此接口来与相应外部装置(例如可从及由外部主计算机操作的调试器或仿真器)交换数据的调试操作模式。所述调试器或仿真器还可操作为编程器,其中经由相同调试接口将程序转移到目标装置中。同时,主计算机与外部调试器或仿真器一起形成便宜分析及调试系统。

[0005] 现代处理器及微控制器在相应装置内侧提供调试功能的经延伸集合。举例来说,可在装置内设置若干个断点以允许装置实际上实时运行,当仅使用高速串行接口时此对于外部调试器将是不可能的且因此将需要外合芯片及昂贵调试电路。然而,这些内部调试电路的功能性当然由于有限量的硅面积(real estate)及其它原因而稍微受限制。

[0006] 举例来说,外部电路内调试器通常不支持追溯功能且当在调试模式中时甚至更复杂的电路内仿真器不能充分地支持跟踪功能。

发明内容

[0007] 因此需要经改进电路内调试系统,特定来说具有经改进电路内调试能力的独立式处理器装置。举例来说,在不以其他方式停止装置操作的情况下,根据各种实施例的处理器装置可允许(举例来说)通过外部引脚馈送到装置的调试单元的外部触发产生跟踪数据包。

[0008] 根据一实施例,一种具有调试能力的处理器装置可包括:中央处理单元;调试电路,其包含跟踪模块及外部接口;其中所述跟踪模块产生包含关于所执行指令的信息的跟踪流,其中所述跟踪流通过所述外部接口输出;且其中所述跟踪模块进一步可操作以检测触发信号且在检测之后即刻将跟踪包插入到所述所产生跟踪流中。

[0009] 根据又一实施例,所述触发信号可为在外部施加到所述处理器的触发信号。根据又一实施例,可通过内部事件产生所述触发信号。根据又一实施例,所述内部事件可包括以下各项中的至少一者:断点、主清除、中断、异常执行条件、调试器溢出条件。根据又一实施

例,所述跟踪流可是基于包的。根据又一实施例,所述跟踪包可包含关于触发源的信息。根据又一实施例,可有条件地提供所述信息,其中所述条件可为用户定义的。根据又一实施例,所述处理器装置可具有多个功能引脚且一个引脚可经指派以作为所述外部触发信号的触发输入操作。根据又一实施例,可在接收所述触发信号时在所述跟踪流中与所报告指令的执行时间一致的点将所述数据包插入到所述跟踪流中。根据又一实施例,所述调试电路可进一步包括可操作以组合多个事件的事件组合器,其中可至少通过以下各项中的一者形成事件:断点及所述外部触发信号。根据又一实施例,跟踪数据可识别导致所述事件的所述断点。根据又一实施例,所述触发信号可为在所述调试电路内同步到系统时钟的非同步信号。根据又一实施例,所述处理器装置可进一步包括与接收所述外部触发信号的外部引脚耦合的噪声拒斥滤波器。根据又一实施例,所述跟踪模块可可编程以检测所述外部触发信号的正缘或负缘。

[0010] 根据另一实施例,一种在处理器装置内调试所执行代码的方法可包括:由中央处理单元(CPU)执行代码;产生由所述CPU执行的指令跟踪流;在确定触发信号之后,即刻将识别所述外部触发信号的跟踪包插入到所述跟踪流中。

[0011] 根据所述方法的又一实施例,所述触发信号可为在外部施加到所述处理器的触发信号。根据所述方法的又一实施例,可通过内部事件产生所述触发信号。根据所述方法的又一实施例,所述内部事件可包括以下各项中的至少一者:断点、主清除、中断、异常执行条件、调试器溢出条件。根据所述方法的又一实施例,所述方法可进一步包括在请求之后即刻通过调试接口输出所述跟踪流。根据所述方法的又一实施例,可在接收所述触发信号时在所述跟踪流中与所报告指令的执行时间一致的点将所述数据包插入到所述跟踪流中。根据所述方法的又一实施例,所述跟踪流可是基于包的。根据所述方法的又一实施例,所述跟踪包可包含关于触发源的信息。根据所述方法的又一实施例,可有条件地提供所述信息,其中所述条件可为用户定义的。根据所述方法的又一实施例,所述方法可进一步包括至少通过以下各项中的一者形成事件:断点及所述外部触发信号;组合多个事件以产生调试功能。根据所述方法的又一实施例,跟踪数据可识别导致所述事件的所述断点。根据所述方法的又一实施例,所述调试功能可为触发输出信号或停止。根据所述方法的又一实施例,所述触发信号可为在所述调试电路内同步到系统时钟的非同步信号。根据所述方法的又一实施例,所述方法可进一步包括通过与外部引脚耦合的噪声拒斥滤波器对所述外部触发信号进行滤波。根据所述方法的又一实施例,所述方法可进一步包括编程所述跟踪模块以检测所述外部触发信号的正缘或负缘。

附图说明

- [0012] 图1展示根据各种实施例的集成调试模块的框图;
- [0013] 图2更详细地展示图1的硬件断点框图;
- [0014] 图3更详细地展示图1的事件组合器框图;
- [0015] 图4表示触发输入脉冲如何导致触发包在跟踪数据中经发射;
- [0016] 图5展示具有跟踪功能性的处理器装置内的调试单元的可能实施细节;
- [0017] 图6展示图1中所图解说明的跟踪块120的简化框图;
- [0018] 图7展示处理器执行及跟踪数据读出的时序图;

- [0019] 图8展示图6中所图解说明的跟踪子系统的实施例；及
- [0020] 图9展示所建议跟踪恢复硬件的框图，展示到微处理器的接口。
- [0021] 图10展示微控制器内部计时信号的时序图；
- [0022] 图11展示跟踪时钟与数据的关系；
- [0023] 图12展示使用具有根据各种实施例的嵌入式调试单元的微控制器及外部电路内调试 (ICD) 控制器的系统的框图。

具体实施方式

[0024] 根据各种实施例，目标装置内的调试单元可具有允许内部事件及/或外部装置任选地非同步地导致断点的触发输入，且断点项目记录于跟踪数据输出中。在一些情形中，外部触发无论如何不可停止处理器的执行，这是因为停止执行可产生不合意或甚至危险操作条件。根据一些实施例，所述触发输入导致允许外部事件与运行指令相关(或反之亦然)的特定跟踪记录。

[0025] 当记录指令数据失去跟踪时，在用户按压按钮时与此时代码正做什么之间存在分离。因此本质上通过在触发输入脉冲上形成跟踪包，其允许用户在分析跟踪流时对准CPU在代码执行流中的位置与发布触发输入脉冲的时间。

[0026] 根据各种实施例，与指令数据流的外围跟踪组合地，当检测到触发输入脉冲时指示发生触发输入脉冲的特殊包出现在流中。

[0027] 图1展示根据一实施例可集成于微控制器内的电路内调试模块的框图。然而，可在其它类型的芯片上调试电路中实施如上文所描述的一般概念。所述框图展示可实施于(举例来说)微控制器中且可由五个基本块组成的模块：

- [0028] • 断点比较逻辑135
- [0029] • 秒表循环计数器逻辑150
- [0030] • 控制及状态机逻辑145
- [0031] • 跟踪逻辑120
- [0032] • 事件组合器逻辑125

[0033] CPU 110通过内部总线与硬件断点单元135、事件检测单元140、控制逻辑单元145及后台接口155耦合。多路复用器160用于允许调试单元与外部调试器通过专用调试时钟及数据引脚165的耦合。控制逻辑145可与一或多个外部引脚耦合。图1展示示范性TRGIN引脚185。此引脚可为不具有其它功能的专用引脚。然而，特定来说在低引脚装置中，此引脚可为允许在程式控制下指派给不同外围装置且因此可根据其经编程指派执行不同功能的多功能引脚。举例来说，此引脚可借助于配置寄存器经编程以在功能上添加到跟踪触发以用作串行接口时钟引脚、数字I/O引脚、模拟输入引脚等。如图1中所展示，控制逻辑还提供可类似于引脚185(多功能引脚)的触发输出引脚190。跟踪模块120与跟踪停滞输入引脚115以及跟踪时钟及数据输出引脚175耦合。图1还展示可通过控制逻辑145配置的脉冲拒斥滤波器170及180。图1中未展示通过这些滤波器的信号路由。根据一些实施例，实施断点调试以使得在执行指令之前停止执行，所谓的“零打滑”操作。根据其它调试器实施例，此为不真实的且可导致关于代码在何处停止或“打滑”的问题，从而允许在停止处理器之前执行指令。外部事件(明显地)与指令执行流非同步。同样地，其操作不可与零打滑概念进行比较。

[0034] 当内部信号debug_en=1时,模块经启用且监视所有“HALT”事件,产生事件,执行数据捕获等。如果内部信号debug_en=0,那么所有调试逻辑经停用且模块经配置以消耗最小功率模式。

[0035] 为使调试较不具有侵入性,具有使数据实时流动离开装置的构件可为有用的。跟踪模块120支持将正被读取或写入的数据带到特定地址且将其发射出跟踪端口的的方法。此可描述为实时观察点。当使用观察点时不中断装置的正常操作。

[0036] 可使用与用于产生硬件断点相同的硬件执行数据捕获。与产生停止同时地使用数据捕获的断点将允许发生两个操作。然而,同时达成数据捕获及数据匹配可产生意想不到的结果,尤其是大于1的断点计数。当数据捕获及数据比较使用相同物理寄存器时,将在每一捕获时更新比较值。为此,内部或外部事件可产生导致处理器中的调试电路将相应数据包插入到跟踪数据流中的触发。外部事件可为施加到触发输入TRGIN 185的外部信号。内部事件可为任何事件,例如断点/观察点、主清除、中断、异常执行条件、调试器溢出条件。异常条件可包含中断、睡眠模式进入或退出、堆栈溢出/下溢、指令从非法地址的提取。表1及2展示一系列可能事件。这些表为非包含的或排他的且应被认为是触发事件的实例。其它内部事件或事件组合可用于产生触发信号以用于将特定数据包插入到跟踪流中。

[0037] 表1

[0038]

值	中断源
7'b00	0004 的系统中断
7'b01	未使用的/保留的
...	
7'b39	
7'b3A	8005 的 ICD_IRQ 0 (软件断点)
7'b3B	800D 的 ICD_IRQ 1 (软件断点)
7'b3C	8004 的停止项目
7'b3D	8004 的断点项目
7'b3E	8004 的事件组合器项目
7'b3F	8004 的其它调试项目

[0039] 表2

[0040]

值	复位原因
4'h0	经保留用于内部测试
4'h1	掉电
4'h2	RESET指令
4'h3	MCLR输入经断言 (在操作期间)
4'h4	看门狗定时器超时

[0041]

4'h5	指令ECC数据错误
4'h6	堆栈下溢

4'h7	堆栈溢位
4'b1xxx	经保留用于将来发展

[0042] 图2更详细地展示硬件断点单元135。硬件断点可经配置以在程序或数据存储器中的匹配地址上打断。为此,提供相应比较器220及230以及解码单元240,如图2中所展示。当断点经选择以在数据存储器上操作时,断点可另外经鉴定具有数据值及掩码,从而仅允许特定值产生断点事件。数据断点还可任选地设置为仅在读取或写入循环上打断。在所有情形中,所述断点具有计数器210以使得在产生断点事件之前特定事件必须发生N次。举例来说,根据一些实施例,此可设置为从1次到256次的任何值。

[0043] 针对单个断点展示图2的框图。然而所实施的断点的数目是根据各种实施例可变的,可存在许多断点。图2展示允许各种参数经编程以定义断点的触发要求的示范性实施例。根据其它实施例,可使用较少或较多此些参数。举例来说,可通过计数器210中的BPxCNT参数设置产生断点所需要的断点发生的数目。每一断点模块可具有相同寄存器。

[0044] 断点在事件通道定义当中列出且可用于开始或停止循环计数器150,设置或复位事件组合器级125,开始或停止跟踪单元120或进行堆栈快照。

[0045] 根据一个实施例,为了使断点经启用,必须设置控制寄存器ICDBPxCON的位BPEN。如果清除此位,那么此特定断点的所有电路经停用且将不产生断点事件。断点可经配置以仅在鉴定条件的每第N次发生时触发动作。举例来说,为将断点设置为在每第三次发生时触发,计数器210设置为BPxCNT=2。可与计数器210组合地使用相应控制寄存器以重新加载其值及/或监视当前状态。

[0046] 还可通过(举例来说)在相关联配置寄存器中设置相应位来基于执行上下文(主线代码、中断处理程序或任一者)而鉴定断点。断点可接着仅在程序依据选定上下文执行时发生。

[0047] 可通过在配置寄存器中设置相应位来使用又一断点参数,此允许监视程序计数器(PC执行地址)。程序存储器打断为零打滑,且在执行操作之前发生。PC将指示触发指令的地址。

[0048] 当清除相应控制位(例如BPAREN='0')时,当PC等于预定义地址时触发打断。当BPAREN='1'时,当PC属于预定义包含性地址范围内时触发打断。

[0049] 根据一些实施例,仅所执行指令可产生断点。如果PC位于未经执行的地址处,那么断点不触发。此包含:

- [0050] • 流动改变指令(CALL、RETURN等),
- [0051] • 跳跃指令(由BTFSS、BTFSC代表),或
- [0052] • 在PCL、FSR或其它两个循环指令之后的下一提取。

[0053] 当在控制寄存器中将另一控制位字段设置为01、10或11时,断点监视数据存取,地址及值两者。相关联位的三个状态选择使用读取还是写入循环来确定断点。

[0054] 在读取或写入数据(视情况)之后数据断点必然地导致指令执行的末端处的打断。在所有情形中,指令运行到完成。因此,“打断”实际上发生在下一指令执行时,且PC将指示继触发指令之后的指令。还可在已满足存储器地址及数据值限定符两者时触发打断。

[0055] 循环计数器150为用于提供秒表功能以使得可剖析用户代码的计数器。由相应控制寄存器控制循环计数器。计数器150可由四个8位计数器/寄存器组成。可在CPU的每Q循环

的末端处使计数器150递增;多循环指令(例如,GOTO)计数多次。

[0056] 为了允许由特定事件控制多个功能,所有可能源可组合成一个事件总线。此允许循环计数器150、跟踪单元120及事件组合器单元125使用相同设置来选择其动作。

[0057] 如图3中所展示的事件组合器300监视多个事件输入320且可基于那些输入的组合及序列而产生停止或触发输出190。在设置相应控制位时启用事件组合器300。经停用组合器300不产生输出事件。事件组合器300在事件通道定义当中列出且可用于开始或停止循环计数器150,设置或复位事件组合器级310,开始或停止跟踪单元120,或进行堆栈快照。当在相关联控制寄存器中设置所述级的相应控制位时独立地启用事件组合器级310。将在相关状态位中反映级的当前输出。级310具有如图3中所展示的隐含次序,且可以若干种方式组合:

- [0058] • 可由事件个别地启动级,
- [0059] • 在下一较低级是有效的时可由事件启动级,
- [0060] • 可由事件个别地撤销启动级,
- [0061] • 可由事件撤销启动级或当撤销启动下一较低级时。

[0062] 通过设置相应控制位,仅经组合事件的第(N+1)次发生将要输出事件发信号。N可设置为从0到255。如果满足经组合触发条件,那么使寄存器递减1。如果满足经组合触发条件,那么产生事件组合器事件且计数器经重新加载有预设置值。同样,在将新计数值写入到相应控制寄存器中的任何时间,重新加载计数器中的值。举例来说,为将断点设置为在第三次发生时触发,相应计数器值应设置为2。

[0063] 另外根据一些实施例,馈送到引脚TRGIN 185的外部信号允许用户输入产生将插入到跟踪流中的跟踪包,产生停止且任选地还触发TRGOUT信号。当“极性”=0(图6)时,触发输入为高态有效且升缘导致事件。当“极性”=1时,触发输入为低态有效且降缘导致事件。另一控制位可用于控制滤波器(举例来说)以定义:输入必须处于有效状态中达最小时间以便被意识到。接着忽略较短脉冲。

[0064] TRGIN事件可在事件通道定义当中列出且可用于开始或停止循环计数器150,设置或复位事件组合器级310,开始或停止跟踪单元120,或进行堆栈快照。如果启用跟踪,那么触发输入中的改变将产生跟踪包。

[0065] 当事件(例如断点)随经启用触发一起发生时,产生TRGOUT引脚195上的脉冲。通过设置相应控制位来配置基本触发输出信号操作。举例来说,这些控制位可控制:断言触发输出以大致用于触发事件的持续。相邻或重叠事件可使信号保持处于经断言状态中。所述控制位还可控制输出是否伸展到最小时间周期。一旦触发TRGOUT单击,便将忽略在定时周期内发生的更多事件。在单击超时且TRGOUT返回到零之后,可再次通过另一事件触发其。所述单击为边缘触发的,且将在预定义时间周期之后清除,即使事件信号持续。

[0066] 软件可通过设置相应控制位来导致触发输出。如果装置是唤醒的,那么在1个循环之后通过硬件清除位。TRGOUT还可通过写入相应控制位来清除,或将在装置唤醒时自动清除。

[0067] 图4展示规则跟踪流以及在外断言的“触发输入”信号。当“触发输入”脉冲出现在跟踪指令流的中间时,将“TRG TRC”包插入到规则跟踪流中以预示那时在流中发生触发。如果设置相应控制位,那么触发输入中的改变将产生跟踪包。如果极性位=0,那么事件将

在升缘上触发。如果极性位=1,那么事件将在降缘上触发。如图4中所展示,导致触发跟踪包的插入的在外部施加的信号与指令的执行非同步。由内部事件产生的触发跟踪包主要为同步的。然而,关于到跟踪数据流中的插入应用相同原则。

[0068] 图5是从触发引脚到跟踪包写入时钟及数据的概念的实施的简化图。展示根据各种实施例的特定信号,但发射包的跟踪模块的部分及一些跟踪功能性可以许多不同方式实现且因此未经展示。

[0069] 根据图5中所展示的实施例,触发信号通过外部引脚610非同步地传入且可借助于(举例来说)噪声拒斥滤波器620经受一些滤波。然而,此滤波器还可被省略或如果经实施那么可实施为模拟或数字滤波器或两者的组合。此外,根据一些实施例可实施极性选择单元630。为此,可提供直接接收噪声拒斥滤波器620的输出或触发信号的简单“异或”门630。“异或”门630的第二输入接收极性控制信号。因此,传入触发信号为反向的或非反向的。因此,用户可控制装置以总是在内部产生正脉冲,无论触发脉冲是什么极性。然而,可应用外部信号的其它处理且因此“触发输入”功能性本身并非关键的。虚线将“触发输入”区段与“触发输入”信号的跟踪使用分开。

[0070] 正触发输入脉冲在装置内为非同步的。其转到检测事件且寄存事件的跟踪模块的第一触发器640。所述第一触发器640还因为作为时钟信号的触发输入脉冲的连接而为非同步的。接着分别经由2个后续内部时钟 t_{q2} 、跟踪时钟 t_{q4} 使所寄存非同步信号触发器输出Q同步。根据一些实施例,这些内部时钟同步到指令时钟但不可对应于内部系统时钟的实际指令时钟,这是因为在一些实施例中,系统时钟实际上可使用正交时钟(举例来说 $4q$ 时钟)来执行一个指令循环。图10展示关于指令的执行的系统时钟 clk_{in} 及相关正交信号的示范性时序图。在此实例中,来源于 clk_{in} 的四个时钟 $q1$ 、 $q2$ 、 $q3$ 、 $q4$ 用于执行一个指令。跟踪时钟可来源于内部时钟 $q1$ 、 $q2$ 、 $q3$ 、 $q4$ 或可为相同的这些时钟。

[0071] 两个触发器660及670串联耦合且由 t_{q2} 及 t_{q4} 内部时钟信号计时,如图5中所展示。触发器670的输出Q接着提供经同步触发信号。触发器670的输出Q可与经实施以提供跟踪模块启用功能的“与”门670耦合。

[0072] 现在可(举例来说)通过调试单元的跟踪模块内侧的“与”门670安全地使用此经同步信号。所述经同步信号转到“带外”写入触发器680,其中“带外”不意指指令,且有效地将形成写入时钟。可如下实施将写入的数据:

```
begin
    if(...)
        else if (...)
[0073]     else if (generate_trigger_in_packet)
        data = TRIGIN PACKET
        else
end;
```

[0074] 因此,触发器680产生经同步写入时钟信号且调试单元产生具有将插入到规则跟踪流中的数据的实际数据包。将插入的数据可为预定的或可具有包含事件相依数据的预定及动态部分。

[0075] 如上文所展示,产生此包的机制可为(举例来说)大if-elseif-else陈述,其中给出将发送什么的优先级。包数据的产生可为优先化的,如上文所展示,其中各种“if-elseif”步骤允许其它输出信号的产生且因此形成预定义次序或优先级。根据一些实施例,触发输入跟踪包可具有相对低的优于其它事物的优先级,例如如以上实例中所展示的溢出错误及其它类似事件。

[0076] 根据一些实施例,当使实际“触发输入”写入信号同步时可产生实际跟踪触发数据。因此可明白一个概念,其为在此处此数据使包发展,且当写入时钟到来时其将要触发数据写入到可插入到跟踪流中的包中。其与其它自然跟踪包交互,且因此需要优先级等等。

[0077] 根据特定实施例,以下章节详述装置与外部调试工具之间的跟踪数据接口。其它实施例可使用不同实施方案。如(举例来说)图6中所展示的跟踪子系统提供可由调试工具捕获及分析的指令执行流的实时记录。跟踪操作假定调试工具可存取源代码及程序存储器内容,且可推断CPU操作的一些方面。

[0078] 当指令写入到PCL(直接或通过INDx)时,新的PC可在程序存储器中的任何地方。在此情形中,指令包由所谓的完整程序计数器(FPC)包替换。如图6中所展示的FIFO可用于匹配核心数据速率与调试工具数据速率(由TRSTALL支配)。FIFO由CPU编码器填充,且由数据编码器清空,如图6中所图解说明。FIFO保存高达4096个指令包,但有效操作大小可通过相应控制位选择。还将WATCH事件包置于FIFO中,因此运行中的指令包的数目通常为较少的。图11展示根据不同配置设置A、B、C及D的跟踪时钟与数据之间的关系。TRCLK输出在双态切换时输出具有每一时钟边缘的一个数据字。输出时钟速率总是与CPU指令速率有联系,且将在软件改变SYSCLK的情况下或在复位改变SYSCLK的情况下改变。根据相应配置,输出速率可总是为每指令周期两个跟踪字。所述速率可减小到每指令1个及1/2个,尽管此将可能导致FIFO溢出。TRCLK输出可经选择以与数据改变同相或异相,此取决于TRCPS控制位的设置。输入FIFO编码可不同于线编码,因此在TRDAT接口中每一FIFO位置可表示多达3个数据字。根据一些实施例,跟踪子系统内的FIFO的溢出还可作为事件用于产生跟踪触发信号以用于插入触发跟踪包。

[0079] 可实施跟踪停滞功能,其中配置位可确定TRSTALL输入是否具有效应。举例来说,当控制位TRXSE=1且信号TRSTALL=1时,跟踪FIFO将停止有效负载边界上的清空且停止时钟。当TRSTALL(tr_stall_pin)返回到‘0’时,计时将重新开始且FIFO将再次开始清空数据。当启用跟踪且FIFO为空的时,根据装置的睡眠模式发送IDLE或SLEEP包。根据一些实施例,还可驱迫FIFO为空的且可撤销启动跟踪。如果FIFO变为完整的,那么系统响应可取决于相应控制设置。在任一情形中,溢出状态(停滞的或未公布的数据)将持续直到FIFO为25%或75%完整(如由相应控制寄存器所选择)。根据一些实施例,可添加关于跟踪实施的其它功能且不可实施如所描述的一些功能。图7中展示典型跟踪信号时序图。跟踪有效负载包编码CPU核心所执行的指令及选定数据事件,且还提供跟踪流同步。根据一实施例,大多数包可由所谓的完整程序计数器(FPC)包的1个或2个七(7)位字或3个字组成。一般来说,包由“字1”及可选“字2”组成。然而,根据不同实施例可使用其它格式。根据特定实施例,“字1”的值识别包,且暗示是否存在“字2”。与指令执行同步的包(如EX、EXD及RESET)及非同步“事件”包包含WATCH、RESET及TRIGGER,如下文更详细地描述。

[0080] 依执行次序发射同步包。事件包在事件的时间附近出现在流中,但如果若干个事

件同时发生,那么一些报告将延期。在一些情形中,FPC为同步报告,但在其它时间其为事件。视需要插入输送层 (TR) 包RESYNC、IDLE及END以管理及识别接口数据流的各种状态。当分析指令跟踪时可丢弃TR包RESYNC及IDLE。

[0081] 依照相应控制位所规定而周期性地插入RESYNC包,以使得接收器可检验其经正确地同步。在大致对应于预定义时间间隔的时间,且只要在所述时间间隔内没有发送其它FPC,FPC就将添加到流。此提供接收器正在正确地跟踪指令流的检查。经插入FPC总是指示所跟随的指令的地址。通过包字1的值来确定包中的字的数目。如果RESYNC作为具有“字2”的不正确值的字1经接收,那么所述流不同步且错误应经加旗标。

[0082] 表3图解说明涉及2字包的最糟情况的情景,其中第二个字为0x7D,后续接着RESYNC。如果接收器适当地经同步,那么所接收字#3将为字1,且字#4将为字2,从而形成展示为包#2的完整RESYNC对。包#3以字#5开始,且将被正确地解释。

[0083] 表3

所接收字		包	
#	值	类型/数据	#
1	0x71	EXD 0xFD	1
2	0x7D		
[0084]			
3	0x7D	RESYNC	2
4	0x7D		
5	0x74	EX	3

[0085] 表4展示与复制第一字且将接收器抛出同步的时钟反弹相同的数据。字#1及#2作为具有data=0x71的EXD包经接收(但此为不正确解释),且字#3及#4似乎为RESYNC对。字#5被视为新RESYNC包的字1,但字#6并非0x7D,从而揭示不同步状态。字#6开始新包。

[0086] 表4

所接收字		包	
#	值	类型/数据	#
1	0x71	EXD 0xF1	1
2	0x71 ⁽¹⁾		
3	0x7D	RESYNC	2
4	0x7D		
[0087]			
5	0x7D	错误	注释2
6	0x74		
			3

注释 1: 接收重复字是时钟电缆中的阻抗不匹配的典型特征

2: 假定跟随任何 0x7D 包(字 1=0x7D)的第一非 0x7D 总是为字 1 (或 FPC 字#3)是充分的。接收器必须立即重新解释作为新包中的第一字的字#6。

[0088] 在跟踪开始时或在跟踪经延缓用于调试执行之后重新开始时发送的第一包将总是为FPC。完整程序计数器 (FPC) 包报告出现在流中的下一指令的绝对地址。FPC在这些情景中报告:跟踪的开始、在溢出之后的重新开始及在调试之后的重新开始,此仅仅指示下一指令的地址。

[0089] FPC的所有其它使用指示已执行指令,且在一些情形中替换本来经报告用于所述指令的包。当FPC表示变更指令的分支或程序计数器的执行时,所报告的值为分支目标地址。跟随FPC的指令包表示FPC所指向的指令的执行。

[0090] GOTO及CALL指令假定程序计数器PC[10:0]的最低有效位为调试环境所习知的 (以汇编代码的形式),且新PC的仅上部4个位在上部部分程序计数器 (UPC) 包中经报告。然而,其它实施例可报告更多或更少信息。所报告值可为0x0F& (PCLATH>>3),其中PCLATH表示锁存程序计数器的上部位的实施特定的寄存器。因为相对分支 (BRA) 的目的地为源代码所习知的,所以指令仅仅作为EX经报告。可实施多种多样的跟踪有效负载。表5展示不同有效负载信号的实例。

[0091] 表5

[0092]

记忆代码	描述	跟踪编码	跟踪编码		所发送的7位字的数目	在TRIEN = 0时经发送
			字1 (字3)	字2		
WATCH	数据跟踪观察点		0 ccccc d	ddd dddd	2	是

[0093]

FPC	全新PC, P = PC[14:0] 暗示当前指令的执行 连续地发送2个包。	100 pppp (PC[14:11])	ppp pppp (PC[10:4])	3	在 RESYNC 间隔内的 1 个 FPC
		101 pppp (PC[3:0])			
UPC	上部部分 PC, P = PC[14:11] 暗示当前指令的执行	110 pppp (PC[14:11])		1	否
EXD	执行指令, D =所存储 的数据	111 000d	dddd ddd	2	否
EX	执行指令	111 0100		1	否
STALL	不执行指令(驱迫到 NOP) PC 未 改 变 (章 节 3.19.3.5)	111 0101		1	否
SKIP	不执行指令(驱迫到 NOP) 使PC递增	111 0110		1	否
OVERFLOW	FIFO已溢出; 数据丢失	111 0111	111 0111	2 ⁽⁹⁾	是
INT	中断向量 中断向量N的向量化	111 1000	nnn nnnn 表3到8	2	否
RESET	CPU经复位 PC现在等于RSTVEC	111 1010	v00 nnnn 图3到4	2	是
ERROR	注意内部错误; 针对细 节参考硬件文档。	111 1010	xlx eeee 图3到4	2	是
TRIGGER	TRGTR = 1且触发输 入改变	111 1011		1	是
SLEEP	TREK = 2*b1X, FIFO 为空的、睡眠	111 1100		1	是

跟踪数据

[0094]

RESYNC	周期性重新同步。 将以相同间隔发送 FPC。	输 送	111 1101	111 1101	2 ⁽³⁾	是
IDLE	TREN = 2'b1X, FIFO 为空的(非睡眠、非 TRSTALL)		111 1110		1	是
END	TREN=2'b00		111 1111	111 1111	2 或 3 ^(2,3)	是
经保留			111 001x 111 1001		1	
注释1: 发送所有字段, 首先MSB						
2: 将至少两次发送END包且可能第三次以使得TRCLK在低状态中停止						
3: 接收器应处置OVERFLOW、RESYNC及END作为1字包; 参见附录A.2.3中的论述。						
图例: c =观察点的通道 p =程序计数器 d =写入数据						

[0095] 表6展示实际跟踪实例:

[0096] 表6

[0097]

循环		指令	包
	PC(0X)	操作码	类型/数据
1	123	MOVLW IIIHII(2300)	EX
2	124	MOVWF PCLATH	EXD 8'h23
3	125	MOVLW #3	EX
4	125	CALL 200	
5			UPC-4 ⁽¹⁾
6	2200	BTFSW.7	EX
7	2201	BRA \$+4	
8			FPC
9	2205	NOP (注释 2)	EX
10	2206	BRW	
11			FPC 220A ⁽¹⁾
11	220A	RETLW#77	
12			FPC

[0098]

13	126	NOP	EX
14	127	CALLW	
15			FPC 2377 ⁽¹⁾
16	2377	NOP	EX
17	2378	GOTO 500	
18			UPC-4 ⁽¹⁾
19	2500	NOP	EX
20	2501	RETURN	
21			FPC
22	128	NOP	EX

[0099] 注释1:如果TRFPCB=1,那么此指令报告FPC。

[0100] 2:循环8的PC值并非15'h2205,这是因为PCLATH=8'h23(来自循环2);UI应对此错误加旗标。

[0101] 图8展示跟踪子系统700的简化框图。子系统700从核心及WATCH事件信号接收指令代码,且格式化此数据以用于在TRDAT信号上递送。序列控制器710负责将指令及WATCH数据加载到FIFO中。每一数据包经编码为单个16位字。在q34期间,对WATCH信号进行取样,且最高优先级的信号经编码、加载及复位。如果断言一个以上信号,那么仅加载最高优先级信号且其它信号必须等待后续q34机会。因为报告优先级基于断点数目,所以可次序颠倒地报告事件。在q12期间,来自先前指令的数据经编码及加载(操作码编码于q3中,且总线数据为稳定q3-q3,且所有数据在上升q1处是有效的)。一般来说,每一指令周期发生此。针对分支及呼叫指令以及中断循环,在第一循环期间未加载任何东西,且在第二循环期间编码包(所谓的“经驱迫NOP”),从而允许以正确PC值发射UPC及FPC。因此,可在分支期间加载两个WATCH包。基于当前核心操作而编码SKIP及STALL。

[0102] 序列控制器710可每指令循环加载两次,且流管理器730可每指令循环卸载两次,从而需要每指令循环高达四个(4)存储器循环。FIFO控制器720管理序列控制器710所提供的的数据。数据在由730请求时以相同次序经递送。流管理器730将16位FIFO字重新格式化成在TRDAT信号上发射的数据字。一些包(例如,EX)针对每一FIFO字产生单个TRDAT字,而其它包(例如,FPC)产生更多。依据需要,将输送管理包RESYNC、IDLE及END插入到流中,且根据TRSTALL输入暂停读出操作。存在在每一指令循环周期期间发射的两个(2)TRDAT字。TRCLK信号将在系统复位(数据将不会丢失)期间暂停(伸展循环的一部分)。

[0103] 图9展示具有与操作为跟踪接收器的调试工具820耦合的根据各种实施例的微控制器810的系统800。所述调试工具可为(举例来说)由申请人制造的Real-ICE电路内仿真器。接收器820执行跟踪同步,丢弃所有IDLE及输送包,且将剩余包的捆绑件发射到远程主机830(举例来说,个人计算机或工作站),在此处执行包流解释。

[0104] • 字1分析—将包转换为3字宽的数据总线。

[0105] • 检查二位字节(RESYNC、OVERFLOW等),且当(a)二位字节并非呈连续字的形式或(b)遇到未实施的字1代码值时对同步错误加旗标。

[0106] • 丢弃IDLE及其它输送包,且将剩余值堆叠到FIFO中。

[0107] • 将全部包发射到远程主机。

[0108] 当断言TRSTALL时,接收器820准备接受高达多于6个TRDAT字(2个连续FPC包的等效物)加上可在重新同步管线中排队的2个字。当释放TRSTALL时,保证包字1对准。来自一串IDLE包的大多数或有时候全部IDLE包可被丢弃且不经运送到远程主机以减小频宽。未实施的操作码及FPC字3(7'h5x)还应加旗标为同步错误,且被视为1字包。

[0109] 以单字包(例如,代码7'h5x)的样式编码FPC的第3个字。无论是否同步,都可假定跟随(例如)7'h5x的任何字的字为新包的字1。出于跟踪字1的目的,OVERFLOW、RESYNC、SLEEP及END包应被视为1字包。当7'h7D被视为1字包时,以下包接着将总是为字1值,即使其为另一7'h7D(成对RESYNC字)。字1对准的(假)数据传递到二位字节分析,其中在两个RESYNC不连续地出现在流中时意识到同步失败。同样地,OVERFLOW(7'h77)可在接收器不同步时出现,且必须经准确地解释,即使第二个字不匹配。此还可对同步错误加旗标。类似推理可适用于SLEEP及END包,这是因为如果接收器不同步,那么仅一个7'h7C或7'h7F可在末端出现,且对于接收器来说暂停同时等待第二值将为不适合的。在另一极端处,三个相同字可出现,且根本不应使接收器混乱。

[0110] 针对完整分析,主机830必须将跟踪数据与用于将微处理器编程的原始源代码进行比较。针对大多数指令,跟踪数据宣布发生执行,但不包含操作细节。当分支仅产生EX包(TRFPCB=0)时,可仅通过检查源代码来确定分支目的地。类似地,根据特定实施例,写入到PCLATH仅发射部分数据,且需要源知识来完成评估。取决于操作,STALL包可在受到影响的指令之前或之后。STALL包可以用用户显示器上的突出符号被看作是非典型操作已发生(例如,文件选择寄存器(FSR)写入到非易失性存储器)的示意。WATCH报告可使触发指令滞后许多包。如果相同观察在发生读出之前触发,那么密布的观察可实际上丢失。因为观察点仅识别数据地址,所以跟踪分析必须能够重建数据存取指示符值,无论依据直接寻址模式(需要BSR及操作码的知识)还是依据间接模式(需要FSR的知识)。观察数据总是来自观察事件的最近发生。注意一些FPC包暗示指令已执行(例如,RETURN指令),但其它例子仅为告知性的。

[0111] 图12展示具有运行发展程序且(举例来说)经由USB接口与外部调试器/编程单元520连接的主机(例如个人计算机)的典型调试/编程系统500。外部调试器编程单元520提供可供应在调试器/编程器520内产生的供电电压的专用接口。然而,其它实施例可经由专用电源供应供应电压或目标系统可为自供电的。可通过同步串行接口向实际调试/编程接口提供由调试器/编程单元520所提供的单向时钟信号ICDCLK及双向数据线ICDData。因此,最少三个连接线(ICDCLK、ICDData及参考电势(GND))可用于耦合调试器/编程单元520与目标系统510(其最低限度可为根据如上文所描述的各种实施例具有调试/编程接口的微控制器)。另外,如上文所描述的外部触发输入信号540可在如图12中所展示的目标系统510或电路内调试器/编程器520中实施且跟踪信号由目标装置510通过专用信号线550转发到电路内调试器/编程器。

[0112] 此系统允许用户编程在主机上运行的调试程序以设置具有如上文所阐释的条件的各种断点且在实时执行程序时接收跟踪信息。尽管调试软件记录各种断点关于其在源代码中的位置,但调试器/编程器520将断点信息传递到设置相应断点且配置其相关联寄存器的目标装置。同样,由调试器/编程器520将跟踪能力的设置及配置传递到目标装置510。举

例来说,可设置通过存储于存储器中的数据值的匹配触发的特定断点。用户接着通过在主 PC 530上运行的调试器软件开始目标装置510的软件的执行。仅在检测到断点时停止目标软件的执行。然而,在目标程序的执行期间连续地转发跟踪信息。主计算机530可评估此跟踪数据且以任一文本形式或使用图形显示使其可用。

[0113] 本发明实施例通过实施如上文所论述的特征而允许跟踪流的更好分析。各种实施例因此使芯片上调试能力的当前技术水平进展,且将允许给更多不同用户带来高技术调试能力。

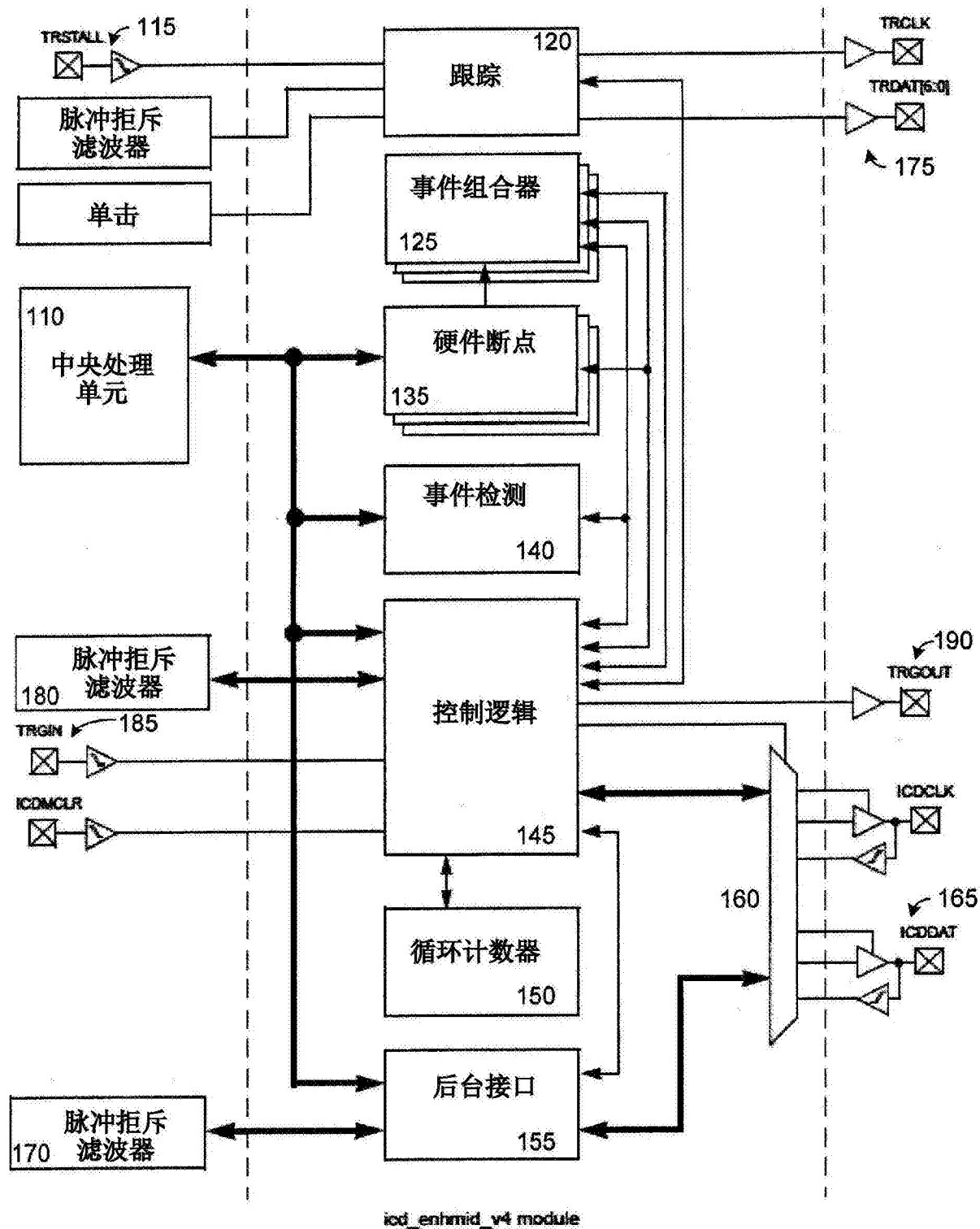


图1

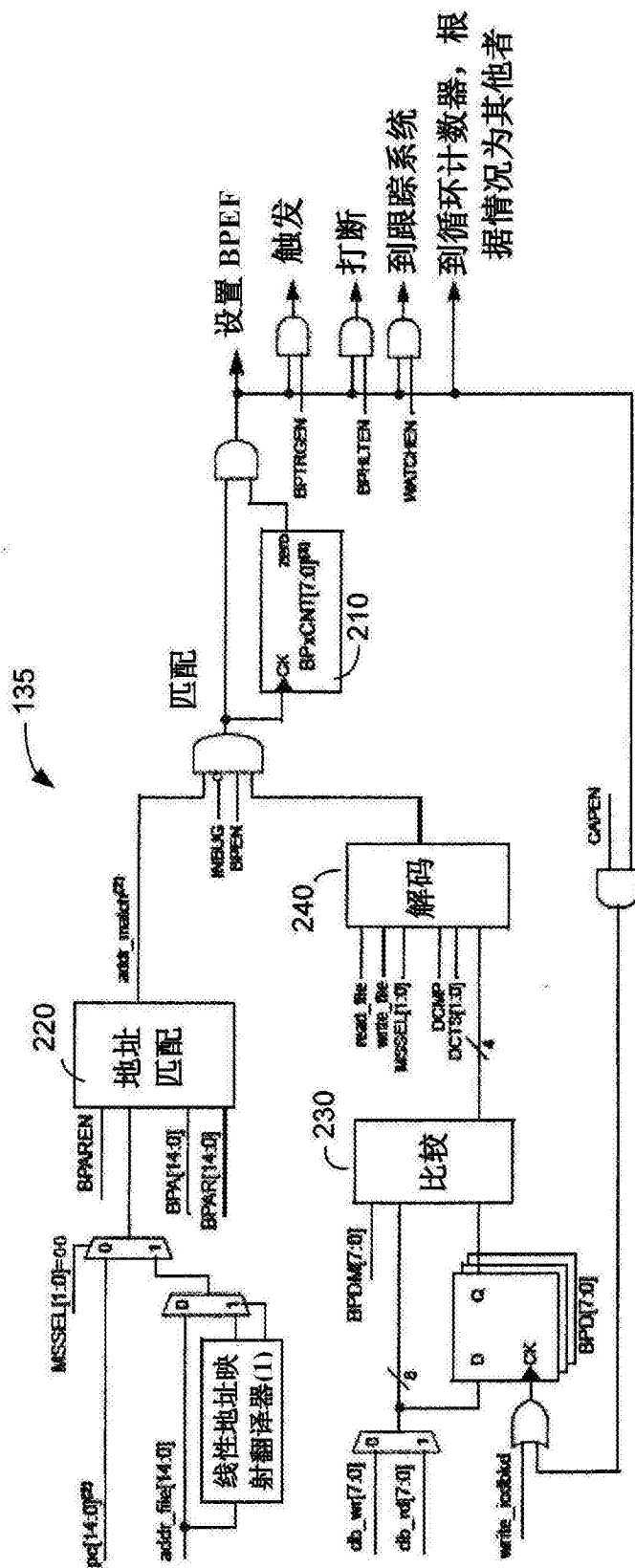


图2

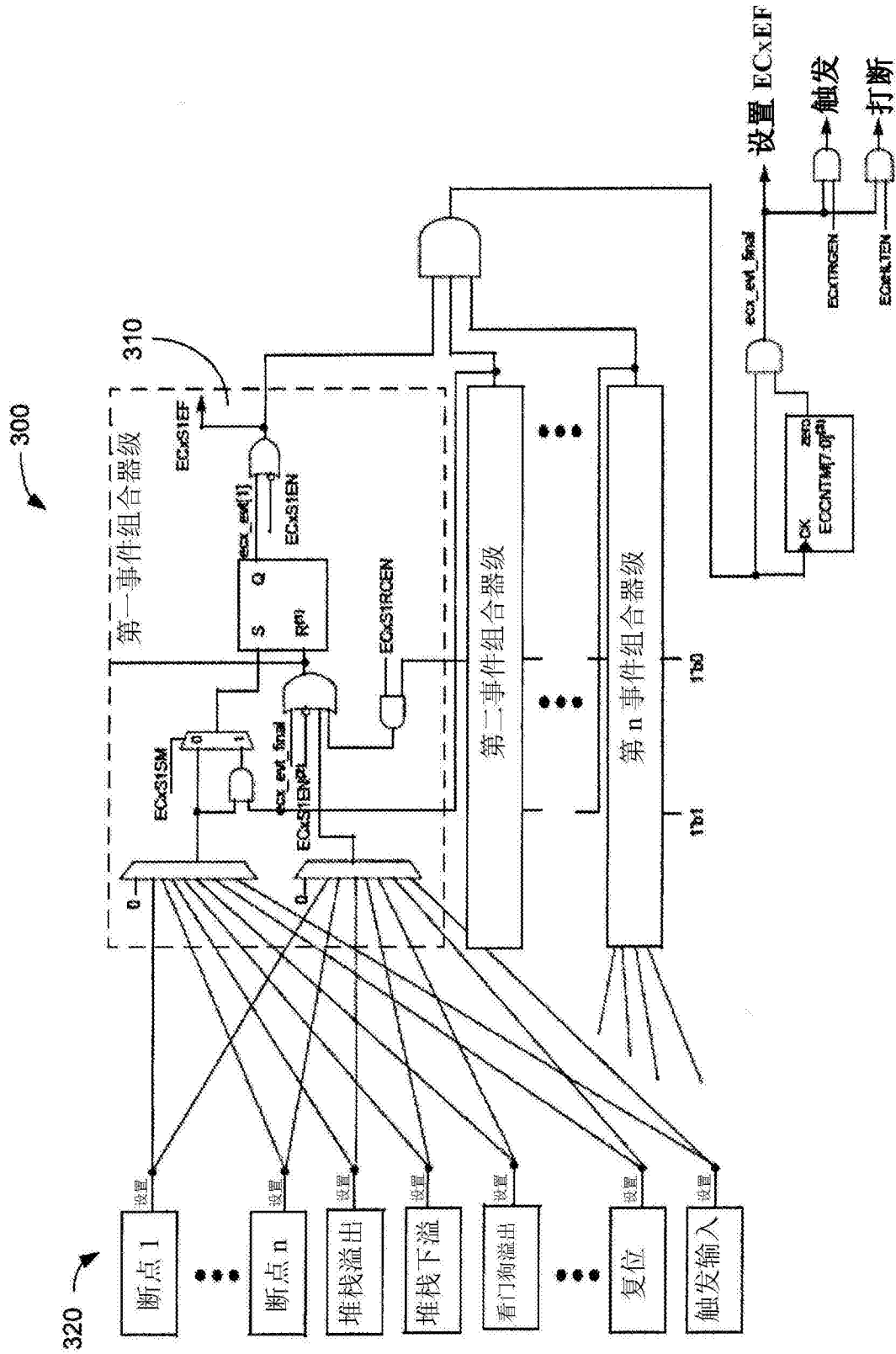


图3

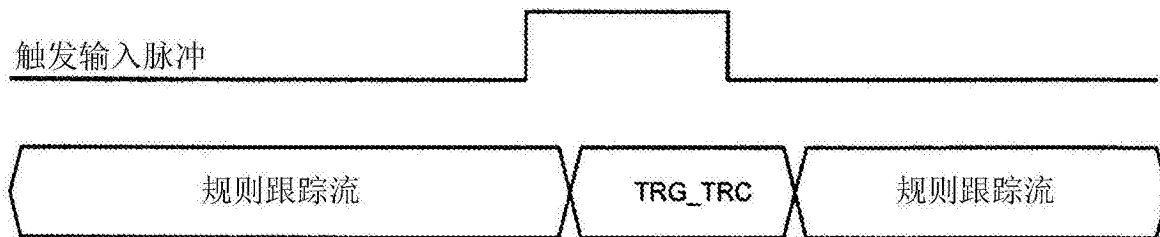


图4

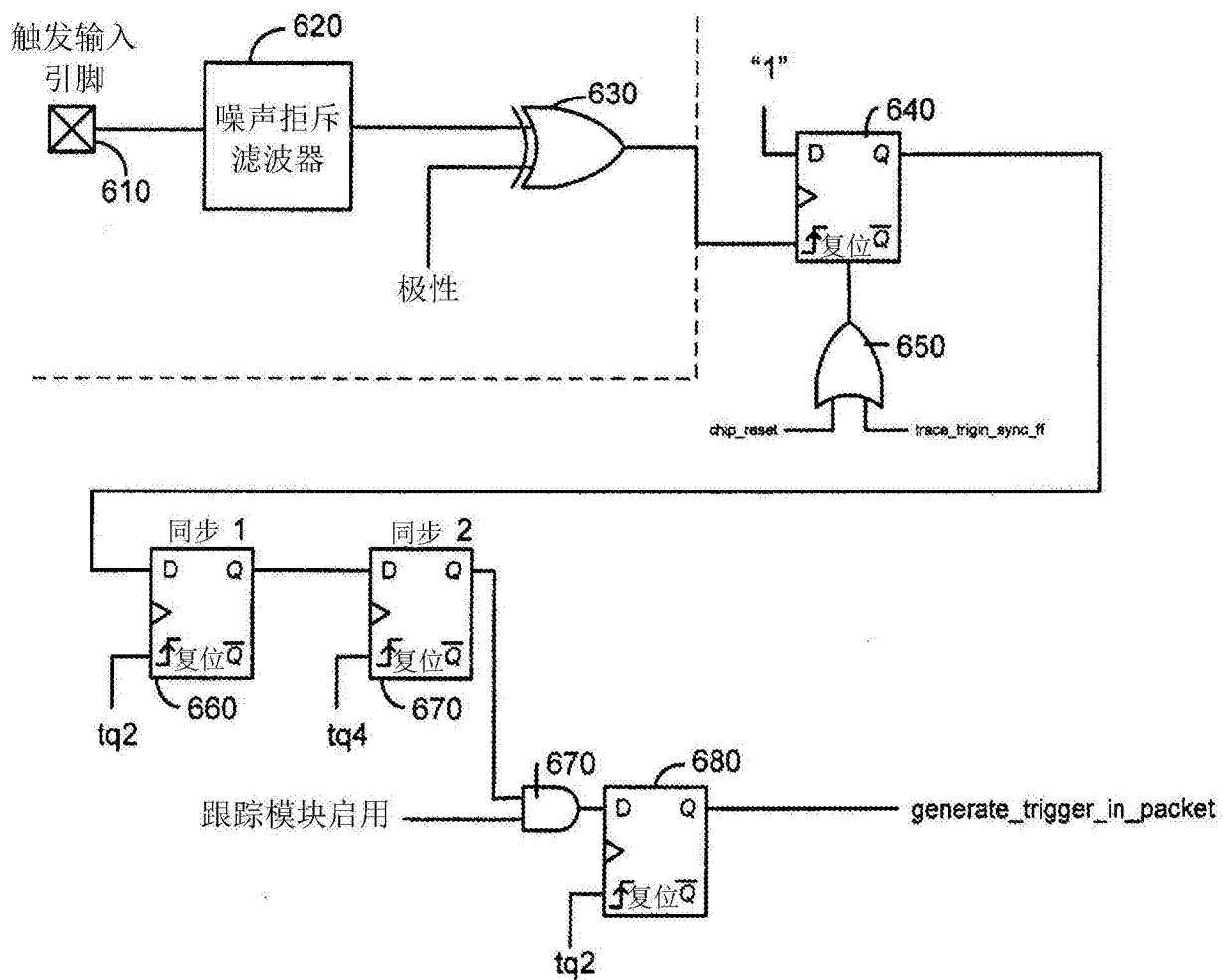


图5

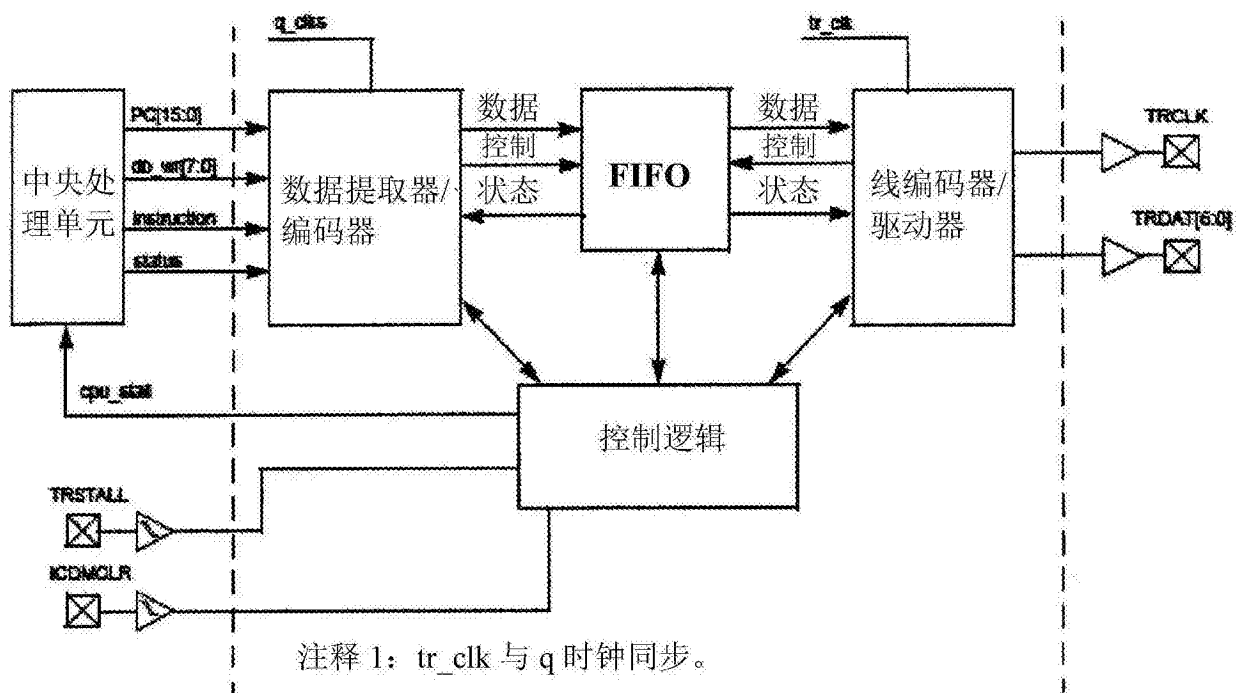


图6

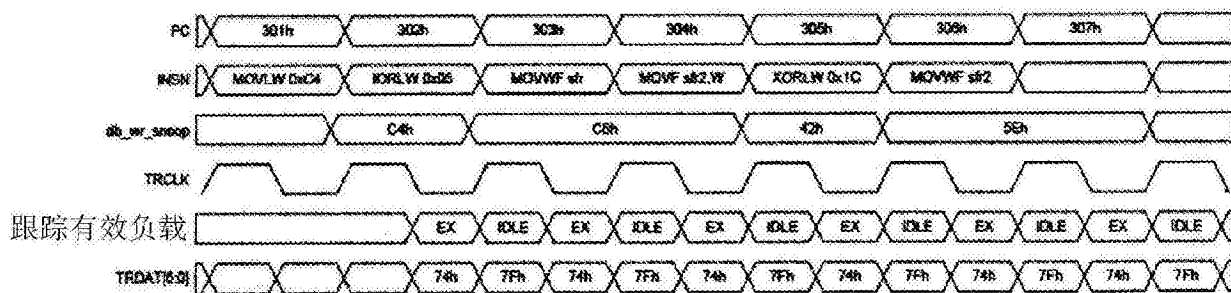


图7

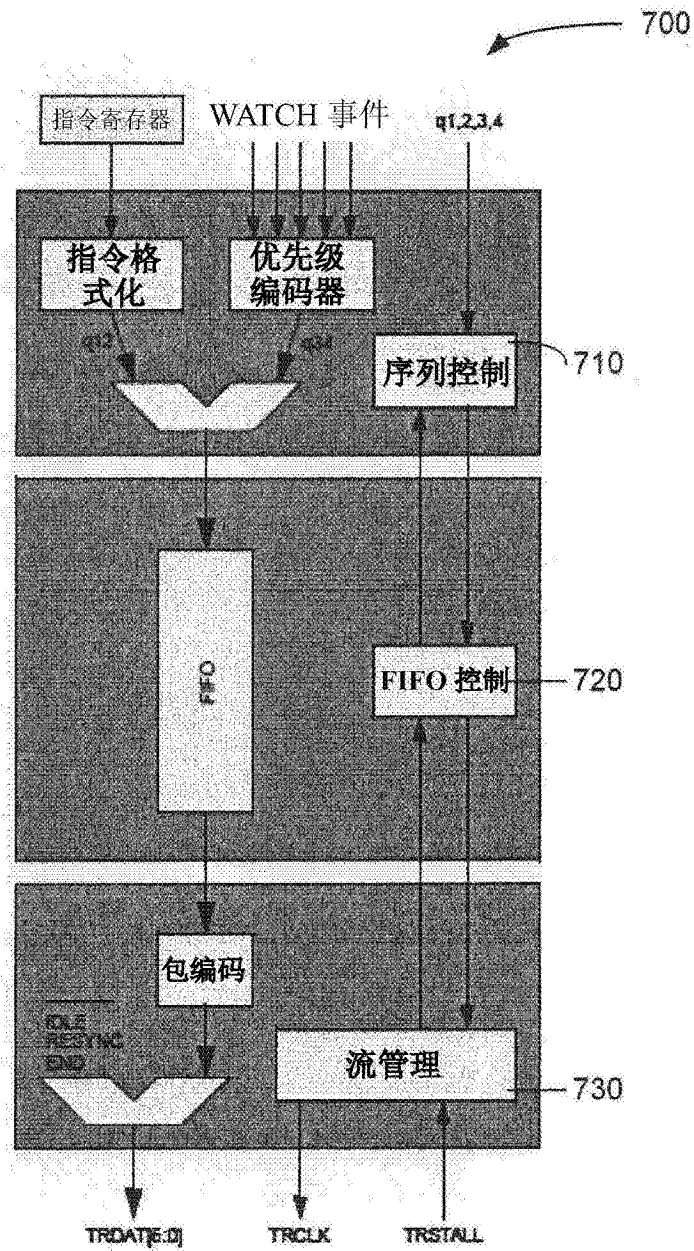


图8

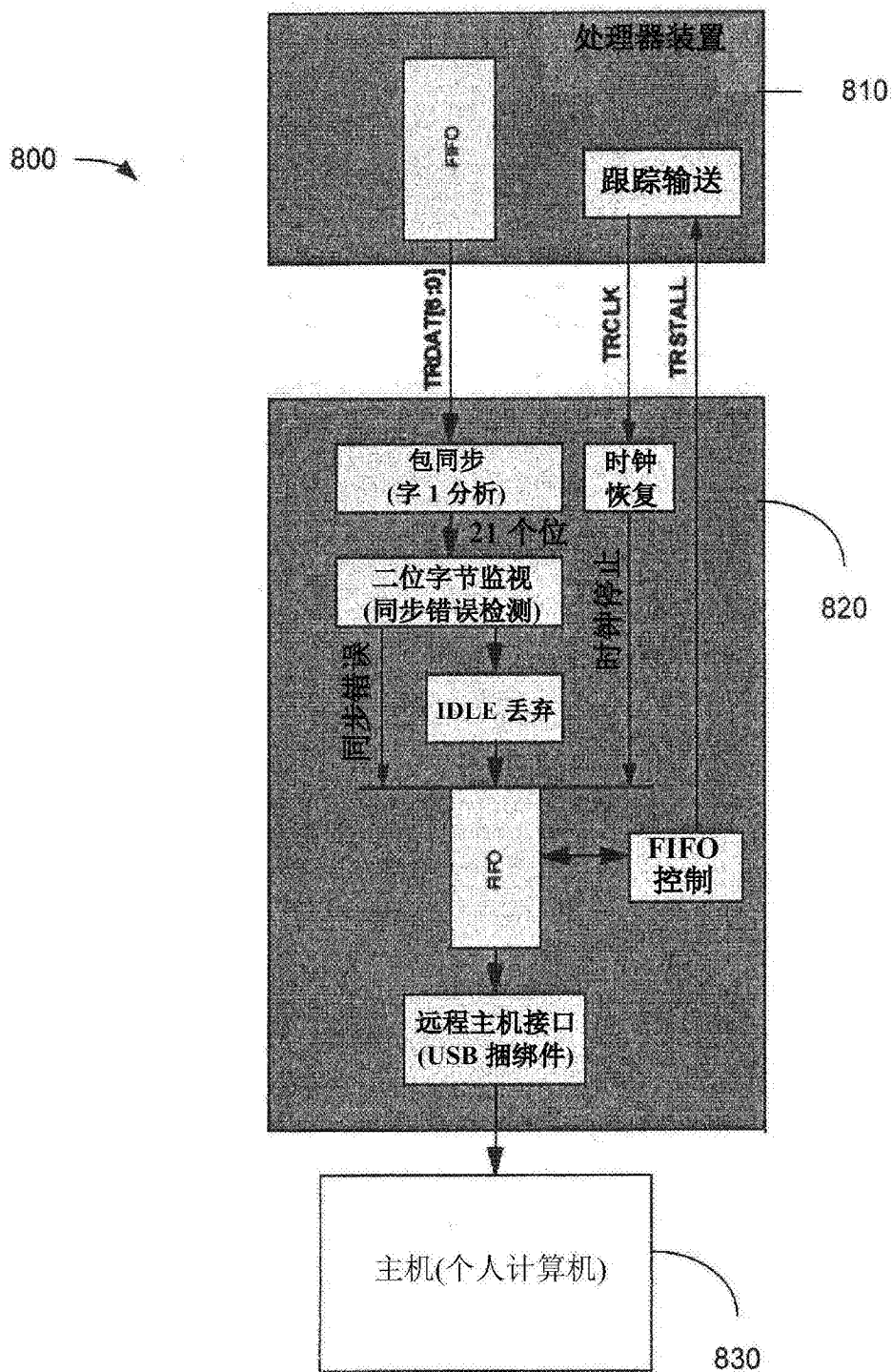


图9

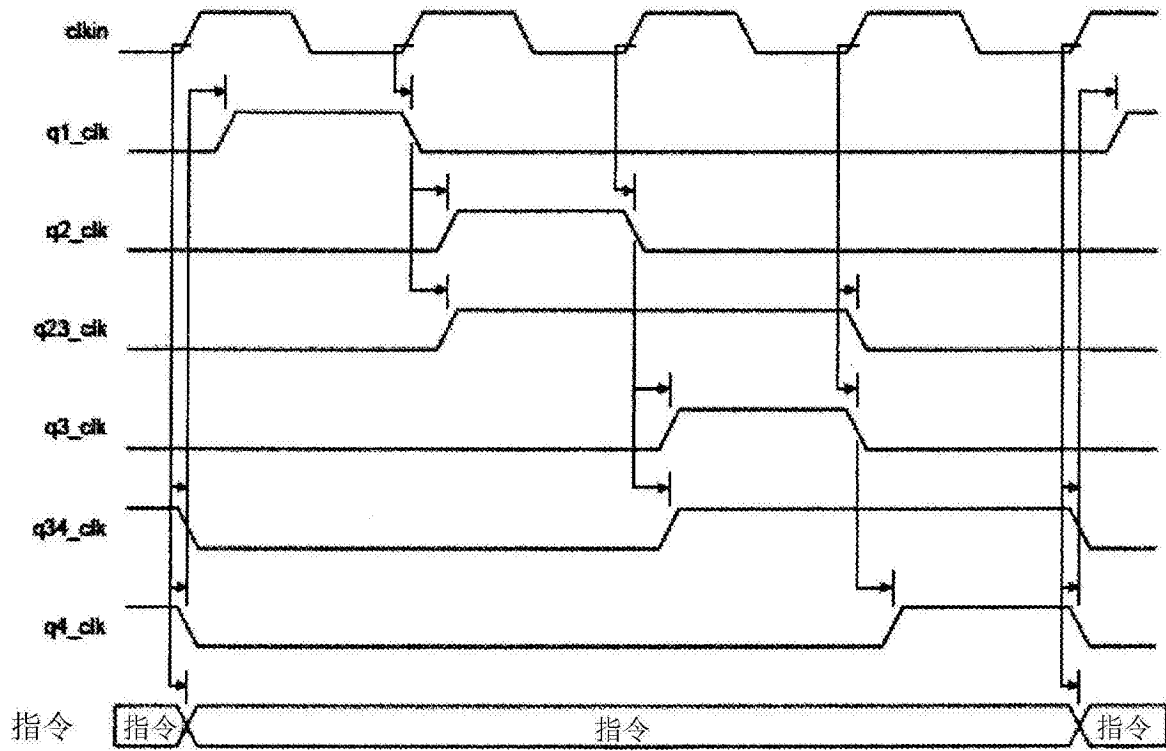


图10

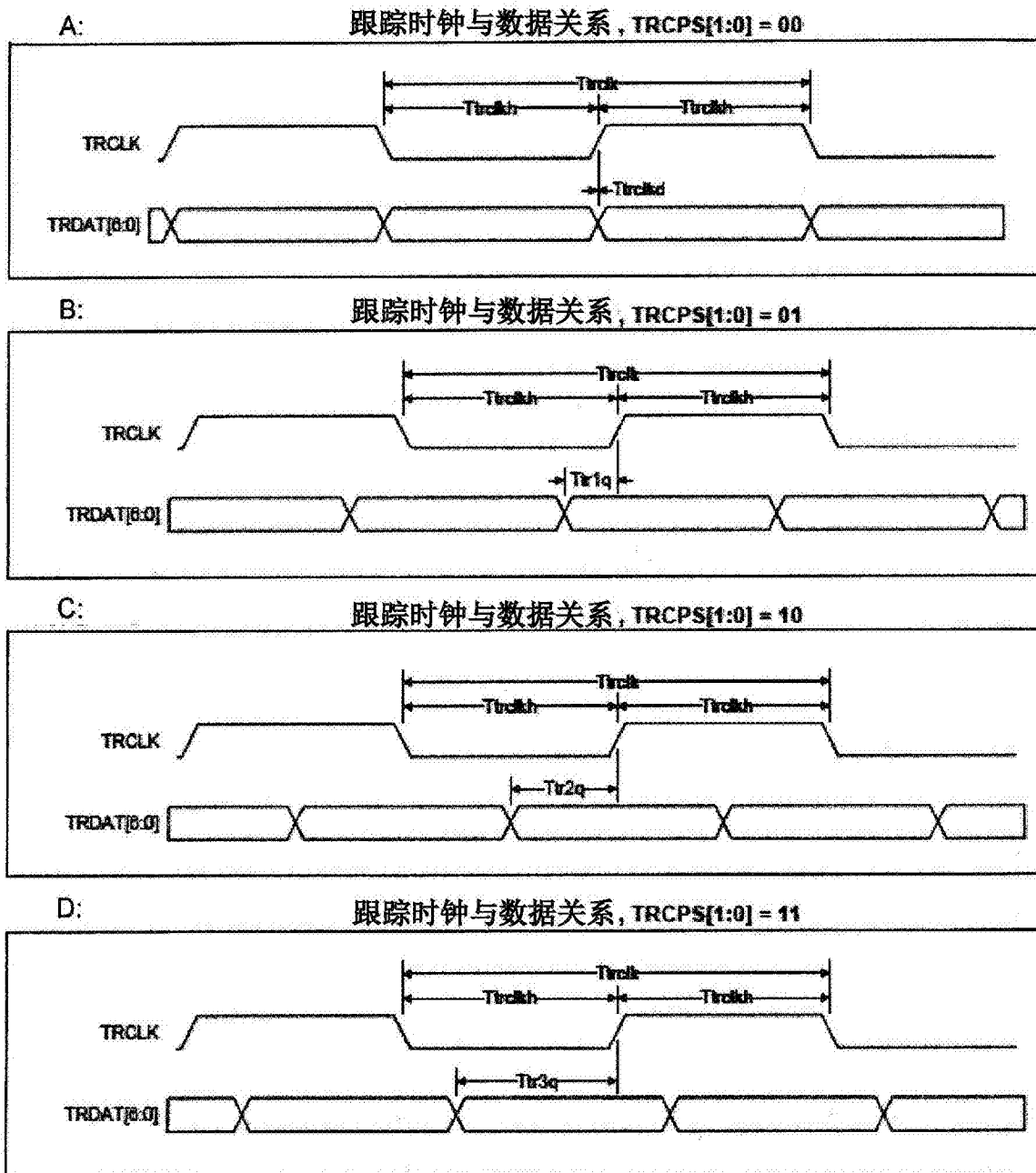


图11

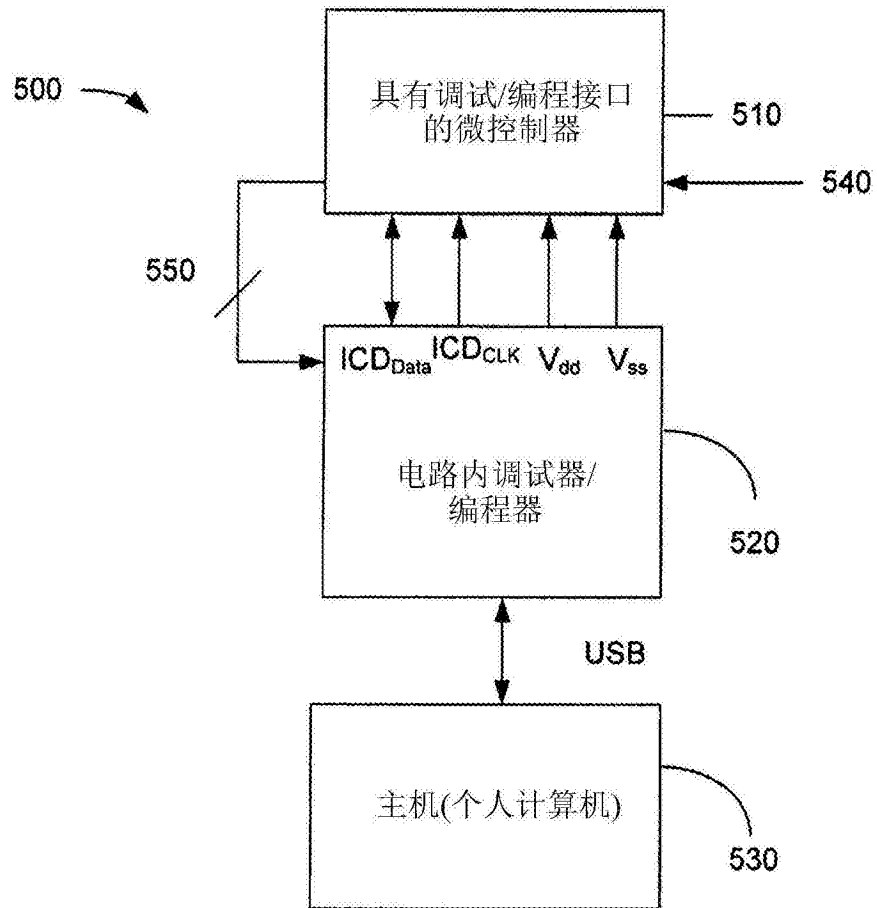


图12