



# [12] 发明专利申请公开说明书

[21] 申请号 200610002596.4

[43] 公开日 2006年7月19日

[11] 公开号 CN 1804809A

[22] 申请日 2006.1.10

[21] 申请号 200610002596.4

[30] 优先权

[32] 2005.1.11 [33] US [31] 11/032949

[71] 申请人 惠普开发有限公司

地址 美国德克萨斯州

[72] 发明人 J·A·贝纳维德斯

T·J·约翰逊 R·L·阿克爾曼

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 杨凯王勇

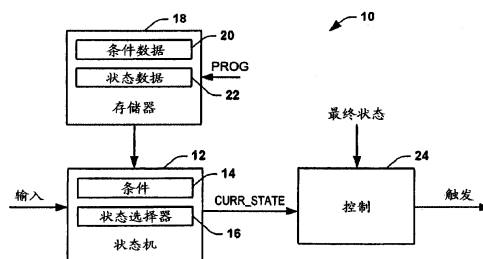
权利要求书3页 说明书24页 附图8页

## [54] 发明名称

产生触发信号的系统和方法

## [57] 摘要

一个公开的实施例可包括专用集成电路(ASIC)(10, 200, 354)。该ASIC包括存储器(18, 52, 102, 206), 该存储器存储定义用于启动多个状态之间转换的条件的条件数据以及定义与各个条件中的每一个相关联的下一个状态的下一个状态数据。状态机电路(12, 50, 106, 210)采用条件数据和下一个状态数据来作为对输入数据应用至少一个条件的函数从状态机电路的当前状态转换到下一个状态。至少一个条件由与当前状态相关联的条件数据定义。状态机电路(12, 50, 106, 210)基于状态机电路的当前状态将下一个状态数据与至少一个条件相关联。控制电路(24, 50, 70, 150, 210)响应于状态机电路的当前状态转换到多个状态中的至少一个预定状态来提供触发信号。



1. 一种专用集成电路 (ASIC) (10, 200, 354), 包括:  
存储器 (18, 52, 102, 206), 其存储定义用于启动在多个状  
5 态之间转换的条件的条件数据以及定义与各个条件中每一个相关联  
的下一个状态的下一个状态数据; 以及  
状态机电路 (12, 50, 106, 210), 其采用所述条件数据和所  
述下一个状态数据来作为相对于输入数据应用至少一个条件的函数  
而从所述状态机电路的当前状态转换到下一个状态, 所述至少一个  
10 条件由与所述当前状态相关联的条件数据定义, 所述状态机电路  
(12, 50, 106, 210) 基于所述状态机电路 (12, 50, 106, 210)  
的所述当前状态将下一个状态数据与所述至少一个条件相关联; 及  
控制电路 (24, 50, 70, 150, 210), 其响应于所述状态机电  
路 (12, 50, 106, 210) 的所述当前状态转换到所述多个状态中的  
15 至少一个预定状态来提供触发信号。
2. 如权利要求 1 所述的 ASIC (10, 200, 354), 其中所述状  
态机电路 (12, 50, 106, 210) 还包括多个条件部件 (14, 58, 108,  
110, 112, 116, 120, 122), 所述多个条件部件对应于实现由所述  
条件数据定义的各个条件的条件转移, 所述状态机电路 (12, 50, 106,  
20 210) 基于应用到所述输入数据的所述各个条件的结果来从所述当前  
状态转换到多个可用状态中的各个下一个状态。
3. 如权利要求 2 所述的 ASIC (10, 200, 354), 其中至少一  
个所述条件部件 (14, 58, 108, 110, 112, 116, 120, 122) 通过  
与至少一个信号匹配来实现所述各个条件中的至少一个, 所述各个  
25 条件中的所述至少一个根据所述状态机电路 (12, 50, 106, 210)  
的所述当前状态而改变, 所述至少一个信号定义所述输入数据。
4. 如权利要求 2 所述的 ASIC (10, 200, 354), 还包括发生  
系统 (66, 130), 所述发生系统操作上用于设置在对于所述多个条

件部件（14，58，108，110，112，116，120，122）中给定部件启动转换到所述多个状态的下一个状态之前满足与所述多个条件部件中所述给定部件相关联的各个条件的若干发生。

5 5. 如权利要求 4 所述的 ASIC（10，200，354），其中所述发生系统（66，130）还包括计数器（68，132），所述计数器提供已经满足了与所述多个条件部件（14，58，108，110，112，116，120，122）中所述给定部件相关联的所述各个条件的若干发生的指示，在指示发生数的值达到预定值时，所述状态机电路（12，50，106，210）启动从所述当前状态到与所述多个条件部件（14，58，108，110，112，116，120，122）中所述给定部件相关联的下一个状态的转换。

10 6. 如权利要求 2 所述的 ASIC（10，200，354），其中所述存储器（18，52，102，206）可编程为定义与所述多个条件部件（14，58，108，110，112，116，120，122）中每一个相关联的所述各个条件以及所述状态机电路（12，50，106，210）基于所述多个条件部件（14，58，108，110，112，116，120，122）为所述当前状态实现的所述各个条件而转换到所述多个状态中的哪个状态，其中根据分配给所述多个条件部件（14，58，108，110，112，116，120，122）的优先级来选择相应下一个状态。

20 7. 如权利要求 1 所述的 ASIC（10，200，354），其中所述状态机电路（12，50，106，210）配置为以与所述 ASIC（10，200，354）的时钟周期基本相当的速率从所述当前状态转换到所述下一个状态，所述速率取决于所述输入数据的频域。

25 8. 如权利要求 1 所述的 ASIC（10，200，354），其中所述至少一个条件还包括与所述当前状态相关联的多个条件，所述 ASIC（10，200，354）还包括选择系统（16，62），所述选择系统操作上基于相对于所述输入数据应用与所述当前状态相关联的所述多个条件并根据分配给所述多个条件的优先级来选择所述下一个状态。

9. 如权利要求 1 所述的 ASIC（10，200，354），还包括嵌入

式逻辑分析器（200，356），所述嵌入式逻辑分析器（200，356）包括：

所述状态机电路（12，50，106，210）和所述控制电路（24，50，70，150，210）；

5        相关数据总线（202，252，354），所述输入数据在其上传播；  
以及

数据捕获系统（214），其响应于指示限定的存储周期的存储信号并基于所述触发信号来存储来自所述相关数据总线（202，252，354）的数据集。

10        10. 如权利要求 9 所述的 ASIC（10，200，354），其中所述嵌入式逻辑分析器（200，356）还包括：发生系统（66，130），其操作上用于设置在启动转换到与给定条件相关联的下一个状态之前满足所述给定条件的若干发生；

15        监控系统（204，250），其提供定义所述输入数据的多个信号，所述多个信号中每一个的值都随所述相关总线（202，252，354）上数据的性能函数而变化，所述状态机电路（12，50，106，210）将与所述当前状态相关联的所述条件应用到所述多个信号，以启动从所述当前状态到所述下一个状态的转换；以及

20        限定系统（208），其通过将给定存储周期内的数据限定为所述多个信号中至少一部分信号的函数来提供所述存储信号。

## 产生触发信号的系统和方法

5

### 相关申请的交叉引用

本申请涉及以下共同转让的同时待审的专利申请：“数据分析的系统和方法（SYSTEM AND METHOD FOR DATA ANALYSIS）”（代理人案号 200314514-1）；“限定数据捕获的系统和方法（SYSTEM AND METHOD TO QUALIFY DATA CAPTURE）”（代理人案号 200314251-1）；“控制数据捕获的系统和方法（SYSTEM AND METHOD TO CONTROL DATA CAPTURE）”（代理人案号 200314252-1），所有这些申请都与本申请同时期提交，并通过引用结合在本文中。

15

### 技术领域

本发明涉及产生触发信号的系统和方法。

### 背景技术

随着在单个集成电路芯片或芯片组上获得的电路集成度的升高，存在与芯片内部操作或芯片组内部操作相关的复杂性增加的趋势。已经开发了各种类型的内部和外部系统以便于芯片或芯片组的监控和/或分析操作。例如，逻辑分析器就是一种能帮助监控和分析操作中某些方面的装置。

25

### 发明内容

根据本发明的一种专用集成电路（ASIC）包括：存储器，其存储定义用于启动在多个状态之间转换的条件的条件数据以及定义与各个条件中每一个相关联的下一个状态的下一个状态数据；状态机电路，其采用所述条件数据和所述下一个状态数据来作为相对于输

入数据应用至少一个条件的函数而从状态机电路的当前状态转换到下一个状态，所述至少一个条件由与当前状态相关联的条件数据定义，状态机电路基于状态机电路的当前状态将下一个状态数据与所述至少一个条件相关联；以及控制电路，其响应于状态机电路的当前状态转换到所述多个状态中的至少一个预定状态来提供触发信号。

### 附图说明

- 图 1 示出产生触发信号的系统的实施例；
- 10 图 2 示出产生触发信号的另一系统的实施例；
- 图 3 示出产生触发信号的一部分系统的实施例；
- 图 4 示出产生触发信号的另一部分系统的实施例；
- 图 5 示出逻辑分析器的实施例；
- 图 6 示出可与产生触发信号的系统结合使用的监控系统实施例；
- 15 图 7 示出可实现逻辑分析系统的一个或多个实施例的计算机系统的示例；
- 图 8 是示出用于产生触发信号的方法实施例的流程图。

### 具体实施方式

- 20 图 1 示出了系统 10 的示例，该系统可用于从相关数据源（例如数据总线）产生数据的“触发（TRIGGER）”信号采样（或捕获）。该“触发”信号可用于激活或发起任何其它操作或过程（例如故障监控和数据分析等）。系统 10 可实现为专用集成电路（ASIC），诸如可用在计算机系统、路由器或可处理或传播数据的其它类型系统中。
- 25

系统 10 包括状态机电路 12，该电路包括多个状态。状态机 12 包括一个或多个条件 14，这些条件基于与一个或多个条件有关的“输入”来控制从当前状态到下一个状态的转换。该“输入”可对应于

一个或多个输入信号或数据位。这里所用的单数或复数形式的术语“数据”和“信号”交替用于标识一个或多个信息单位（例如一位或多位）。

5 一个或多个条件 14 可包括任何类型的一个或多个部件，包括离散逻辑、组合逻辑、寄存器、比较器或基于“输入”确定一个或多个预定条件或事件是否发生的其它电路部件。例如，一个或多个条件 14 可实现与“输入”有关的算术功能、逻辑功能和匹配功能中一个或多个的组合。一个或多个条件 14 还可根据状态机 12 的当前状态而改变（例如编程为改变）。在对于当前状态存在多于一个条件  
10 时，这些条件可对应于状态机 12 的单独条件转移（branch），用以基于应用到“输入”的每个条件转移的结果来从当前状态转换到各个下一个可用状态。

状态选择器 16 基于应用到“输入”的一个或多个条件 14 的结果来选择状态机 12 的下一个状态。可将一个或多个条件 14 中的任意个与多个状态中的每一个相关联。在将多于一个条件 14 应用到当前状态的“输入”数据时，状态选择器 16 根据来自相关条件中所选条件的结果来选择状态机的下一个状态。各种方案都可用于选择将  
15 使用哪个下一状态。例如，可分配一个级别来设置当前状态每一个条件之间的优先级（例如对应于优先编码器）。

20 附加或备选地，状态机 12 可编程为：在对于给定的一个或多个条件 14 启动转换到下一个状态之前，要求这种条件的多于一个发生（occurrence）。例如，状态机 12 可包括用于在已经满足了一个或多个条件 14 时对每个实例进行计数或累加的电路。状态机 12 还能实现以与系统 10 的时钟周期基本相当的速率从当前状态转换到相应的  
25 下一个状态。该转换速率将倾向于根据“输入”的频域而变化。例如，如果以比系统 10 的时钟（例如实现本系统的 ASIC 的系统时钟）慢的频率提供“输入”，则状态机 12 中的同步电路诸如可通过增加“伪”数据（其分开实际数据以便分析）来使“输入”与时钟信号

同步。

系统 10 还包括存储器 18，其存储用于配置状态机 12 的数据。存储器 18 可存储条件数据 20，来对状态机 12 应用到“输入”的一个或多个条件 14 进行编程。状态机 12 的一个或多个条件（例如电路）可与用于读取当前状态条件数据的存储器 18 通信地耦合。存储器 18 还可存储下一个状态数据 22，该数据定义对于当前状态所实现的各个条件中每一个的下一个状态。状态机 12 基于状态机的当前状态将下一个状态与一个或多个条件 14 相关联（例如通过从存储器 18 中读取相应的下一个状态数据）。如上所述，状态机 12 所应用的一个或多个条件 14 中每一个的下一个状态都可根据当前状态而改变。条件数据 20 和状态数据 22 通过 PROG 所示的程序信号是可编程的，以定义与状态机 12 的一个或多个条件 14 中每一个相关联的各个条件可用状态转换。存储器 18 可实现为寄存器阵列或其它可寻址存储器。

状态机 12 向控制块 24 提供当前状态的指示，如 CURR\_STATE 所示。控制块 24 响应于状态机 12 的当前状态到多个状态中的至少一个预定状态的转换来提供“触发”信号，如“最终状态”所示。

“触发”信号可用于触发来自相关数据源的数据采样。例如，诸如本文所述的数据捕获系统可采用“触发”信号来存储来自数据源（例如总线）的数据。“输入”还可随响应于“触发”信号存储的数据的功能而变化。“触发”信号还可用于发起系统 10 内部或系统外部的另一功能或过程（例如硬件或软件）。

图 2 示出了可用于逻辑分析总线（诸如多位同步可观测性或调试总线）上提供的数据的状态机系统 50 的示例。分析系统 50 利用存储向量的存储器 52，该向量可包括定义用于实现状态机的一个或多个条件的屏蔽数据 54。存储器 52 还可包括定义可用状态之间状态和转换的状态数据 56。例如，存储器 52 可以是任何类型的系统可寻址存储器（例如寄存器阵列，诸如 CSR），其诸如可从实现分析系



统 50 的计算机系统的系统处理器写入。其它工具和接口也可用于对存储器 52 进行编程。还可读取存储器 52，以驱动状态机的状态转换。

分析系统 50 实现基于一个或多个输入信号 TRIG\_OUT\_LIST 在多个可用状态之间转换的状态机，该输入信号描述了总线上数据的性能特性。本领域的技术人员将了解和理解分析系统 50 可实现为分析在 TRIG\_OUT\_LIST 信号中提供的信息的各种方式。分析系统 50 可包括一个或多个条件部件 58，该条件部件控制状态机从当前状态 (CURR\_STATE) 到“下一个状态”的状态转换。CURR\_STATE 可包括一位或多位 (例如三位值)，用于确定将如何分析和捕获在总线 (例如调试总线) 上传播的数据。可能状态的序列、状态之间的转换、每个条件部件 58 所执行的功能都可编程为由屏蔽数据 54 和状态数据 56 定义的存储器 52 中的状态转换向量。

在图 2 的示例中，条件部件 58 描述为条件 1、条件 2 和条件 Q，其中 Q 是正整数 ( $Q \geq 1$ )，表示可为每个状态实现的条件转移和函数的数量。本领域技术人员将了解和理解，在分析系统 50 中可使用各种类型和其它数量的条件部件 58。例如，条件部件 58 对应于在 TRIG\_OUT\_LIST 上执行的条件逻辑和条件转移，以控制状态机的状态转换。条件部件 58 用比较块 (例如比较器电路) 60 来根据从存储器 52 读取的屏蔽数据 54 而在 TRIG\_OUT\_LIST 上实现其各自功能。

例如，每个条件部件 58 的比较块 60 都可实现与 TRIG\_OUT\_LIST 表示的至少一部分性能条件数据有关的按位屏蔽 (或匹配)。由此，比较块 60 可基于存储为屏蔽数据 54 的屏蔽向量来实现与每个周期的匹配。对于每个比较块 60，屏蔽数据 54 中的向量可以不同。对于给定的捕获会话，屏蔽数据 54 还可以是固定的，或者诸如通过对于一些或全部可用状态采用不同的屏蔽向量，屏蔽数据可随捕获会话而变化。当给定条件部件 58 的屏蔽向量与当前状态的 TRIG\_OUT\_LIST 匹配时，该条件部件向选择器 62 提供相应的输出，该输出指示已经满足了条件 (例如启动了向量)。

选择器 62 操作上用于基于来自条件部件 58 的输出来标识状态机的“下一个状态”。条件部件 58 可用作控制状态转换的元件的级别配置。例如，条件部件 58 可用作基于 CURR\_STATE 并基于 TRIG\_OUT\_LIST 来实现状态转换的优先编码器。作为优先编码器，选择器 62 可基于根据分配给各个条件部件 58 的优先级而启动了哪个条件部件来设置“下一个状态”。因此，条件部件 58 可操作为单独的条件转移，其可用于基于 TRIG\_OUT\_LIST 和与每个条件转移相关联的相应屏蔽数据 54 的比较来实现状态机的预定状态转换（例如预先编程为状态数据 56）。

选择器 62 向状态寄存器 64 提供下一个状态信息。由此状态寄存器 64 提供作为 CURR\_STATE 信号的当前状态的指示。如上所述，CURR\_STATE 可用于从状态数据 56 中选择下一个可用状态，以及（可选地）将要应用的屏蔽向量重新定义为当前状态的每一个条件部件 58。由于通过将条件部件 58 和选择器 62 实现为硬件（例如部分 ASIC），TRIG\_OUT\_LIST 可变化到系统 50 的每个时钟周期的速率，因此状态机能以高达每个时钟周期的速率来实现到下一个状态的转换。

系统 50 还可包括发生系统 66，该发生系统操作上用于在使选择器 62 能够对于给定条件部件转换到下一个状态之前要求一个或多个给定条件部件 58 的多个命中或发生。虽然其它的发生要求也可与分析系统 50 的其它条件转移结合使用，但出于说明目的，图 2 的示例假设发生系统 66 仅应用于条件 1。由此发生系统 66 向选择器 62 提供发生启动信号，该信号指示对于给定条件部件（例如条件 1）58 是否已经满足了预定发生数。由此选择器 62 可仅在例如发生启动信号指示已经满足了发生数时选择分配给条件 1 的下一个状态。

例如，发生系统 66 包括计数器 68，该计数器操作上用于在条件 1 的比较块 60 指示对于 CURR\_STATE 满足了相应屏蔽向量时对发生进行计数。存储器 52 可向发生系统 66 提供发生值（OCC\_VAL）。

OCC\_VAL 的值定义在与条件 1 相关联的屏蔽数据向量可使选择器 62 能够加载与条件 1 相关联的下一个状态向量之前要求的一个或多个发生的数目。对于状态机的不同状态，可对相同或不同的发生值进行编程。发生系统 66 对 OCC\_VAL 和计数器 68 提供的值进行比较，并基于该比较向选择器 62 提供发生启动信号。发生启动信号屏蔽与条件 1 相关联的下一个状态向量，直到计数器 68 的输出满足了 OCC\_VAL 为止。因此，直到已经满足了与条件 1 相关联的发生要求为止，状态机的下一个状态才会对应于与一个其它条件部件（条件 2 或条件 Q）58 相关联的一个下一个状态向量。

分析系统 50 还包括触发脉冲发生器 70。触发脉冲发生器 70 操作上基于与预定“最终状态”有关的 CURR\_STATE 产生“触发”信号，该“最终状态”可存储在存储器 52 中。触发脉冲发生器 70 还可包括促使该触发脉冲发生器在一个或多个预定条件下提供触发信号的附加逻辑。本领域技术人员将了解和理解可产生触发信号的各种方式，诸如基于期望的性能特性和设计要求。

系统 50 还包括延迟系统 72，该延迟系统 72 操作上用于基于“触发”信号和 STOR\_QUAL 信号来产生 TRIG\_DELAY 信号。例如，假如没有宣称（assert）TRIG\_DELAY 信号而宣称了 STOR\_QUAL 信号和“触发”信号（例如对应于限定的触发事件），则延迟系统 72 可采用增加其值的计数器。在触发脉冲发生器已经宣称了“触发”信号之后，计数器可在每个限定的存储周期增加其值。由此，延迟系统 72 可包括比较器（或其它电路，未示出），该比较器比较计数器的输出与预定的计数器值，如 POST\_STORE 所示。POST\_STORE 值可以从相应系统可寻址存储器 52 中读取的用于实现期望触发延迟的预定值。诸如对于给定的捕获会话，POST\_STORE 值可编程为定义触发延迟值，该触发延迟值设置与相应触发事件有关的数据捕获点（例如在宣称“触发”信号时）。

例如，相应的数据捕获系统可在与触发事件有关的一个或多个

存储周期中任意个周期上基于 POST\_STORE 值(例如 POST\_STORE = 0) 来捕获数据集。数据集可包括触发事件之前的过去数据或与触发事件重叠的数据。根据计数器的大小, POST\_STORE 值还可设置为存储触发事件发生之后的将来数据。例如, 通过在触发事件之后的多个周期上从数据捕获系统读取数据并将数据存储存储在存储器中可以存储将来数据。由此可将 TRIG\_DELAY 信号与 STOR\_QUAL 信号一起提供给数据捕获系统, 用于控制诸如本文所述的数据捕获系统的操作。

图 3 示出了可用于控制在总线(诸如多位同步调试总线)上提供的数据的分析的分析系统 100 的示例。分析系统 100 包括可寻址存储器 102, 该存储器可被写入以存储定义多个状态以及用于基于 TRIG\_OUT\_LIST 来实现状态机电路 106 的条件的向量。分析系统 100 可包括一个或多个条件转移, 这些条件转移定义状态机电路 106 的可能转换。存储器 102 向多位总线 104 提供数据 (RD\_DATA), RD\_DATA 控制状态机电路 106 的操作。存储器 102 可以是诸如可从实现分析系统 100 的计算机系统的系统处理器写入的任何类型的系统可寻址存储器(例如寄存器阵列)。存储器 102 还可被读取, 以用 RD\_DATA 驱动总线 104。

通过另一个示例, 状态机电路 106 采用来自总线 104 的选定子范围的数据(例如存储在存储器 102 中的)来实现基于诸如本文所述的 TRIG\_OUT\_LIST 确定状态转换的条件。状态机电路 106 提供当前状态 (CURR\_STATE) 的指示, 基于该指示可产生“触发”信号和触发延迟 (TRIG\_DELAY) 信号(见图 4)。CURR\_STATE 可包括一位或多位(例如三位信号), 例如用于确定将如何捕获和分析在相关总线(例如调试总线)上传播的数据。

本领域技术人员将了解和理解状态机电路 106 可实现为分析在 TRIG\_OUT\_LIST 信号中提供的性能信息的各种方式。例如, 状态机电路 106 可采用具有多个条件转移的条件逻辑来实现状态机的相应

状态转换。可能状态序列、状态之间的转换以及在给定状态中执行的操作可编程为存储器 102 中的状态转换向量，并可从总线 104 读取。状态机电路 106 操作上用于根据状态机的 CURR\_STATE 基于 TRIG\_OUT\_LIST 信号中所包含的数据来实现状态转换。

5           在图 3 示例中，状态机电路 106 包括多个条件转移，每个条件转移基于相对于 TRIG\_OUT\_LIST 数据应用来自总线的屏蔽数据来实现用于在状态间转换的各个条件。每个转移包括“与”门 108、110 和 112，这些“与”门将来自总线 104 的屏蔽数据（例如在存储器 102 中编程的向量）和“同”门 114 的输出相“与”。“同”门 114 通过执行与 TRIG\_OUT\_LIST 信号和可编程 TRIG\_NOT 信号相关联的数据位之间的“同”功能来操作为按位反相器。由此“同”门 114  
10 向各个“与”门 108、110 和 112 的输入提供对应于 TRIG\_OUT\_LIST 数据的按位反相型式的相应多位输出。

          来自总线 104 的多位 IF\_MASK 向量被提供给“与”门 108 的另一个输入以及按位比较器 116 的输入。“与”门 108 的输出提供给比较器 106 的另一个输入。由此，“与”门 108 和比较器 116 的组合确定是否已经基于 TRIG\_OUT\_LIST 数据启动了 IF\_MASK 向量。例如，在“同”门 114 的输出（例如对应于 TRIG\_OUT\_LIST 的一位或多位）启动 IF\_MASK 向量时，宣称比较器 116 的输出，用于从  
15 总线 104 加载 IF\_NEXT 向量。由此比较器 116 的输出对应于“与”门 118 的多位输入的一个输入位。

          状态机电路 106 包括可启动状态转换的其它条件转移，诸如在 IF\_MASK 向量没由 TRIG\_OUT\_LIST 启动时。在图 3 的示例中，存在另两个分别由来自总线 104 的 ELS1\_MASK 向量和 ELS2\_MASK 向量控制的条件转移。总线 104 向“与”门 110 的输入提供 ELS1\_MASK 向量，以便实现与“同”门 114 的输出（例如对应于 TRIG\_OUT\_LIST 的按位反相型式）有关的另一个屏蔽条件。“与”门 110 向比较器 120 提供多位输出，该比较器将“与”门 110 的输出和来自总线 104 的  
20

ELS1\_MASK 向量相比较，以确定是否启动了由 ELS1\_MASK 定义的相应屏蔽。比较器 120 向“与”门 118 提供另一个加载控制信号，用于从存储器 102 加载 ELS1\_NEXT 向量。ELS1\_NEXT 向量定义从 CURR\_STATE 到存储器 102 中 ELS1\_NEXT 向量所定义的状态的状态转换。

ELS2\_MASK 向量可用于启动到与状态链中另一个条件转移相关联的下一个状态 (ELS2\_NEXT) 的状态转换。例如，ELS2\_MASK 可操作用于在不满足 IF\_MASK 向量和 ELS1\_MASK 向量条件时控制状态转换。在图 3 的示例中，ELS2\_MASK 向量和“同”门 114 的输出一起提供给“与”门 112 的输入。“与”门 112 执行按位“与”功能，以驱动比较器 122 的多位输入。比较器执行“与”门 112 的输出和 ELS2\_MASK 向量之间的按位比较。比较器 122 驱动“与”门 118 的多位输入的另一位，用于从总线 104 加载相关 ELS2\_NEXT 向量。也就是说，在基于 TRIG\_OUT\_LIST 启动 ELS2\_MASK 向量时，屏蔽系统可转换到 ELS2\_NEXT 向量所定义的状态链的另一个条件转移。

通过另一个示例，如下表 1 提供了表示可能状态转换的真值表，这些状态转换可由状态机 100 的状态机电路 106 基于各个比较器 116、120、122 的输出来实现。由此表 1 中的项对应于由比较器 116、120 和 122 提供的状态机 100 的条件转移的输出。例如，转移 1 (“与”门 108 和比较器 116) 对应于第一或最高优先条件 (例如 “if” 条件)，转移 2 (“与”门 110 和比较器 120) 对应于下一个最高优先条件 (例如 “else if” 条件)，而转移 3 (“与”门 112 和比较器 122) 对应于最低优先条件 (例如另一个 “else if” 条件)。各个比较器中每一个的输出值，即是启动 (由逻辑 “1” 表示) 还是禁止 (由逻辑 “0” 表示) 各个向量。在表 1 中，字母 “X” 表示与比较器 120 和 122 的各个输出相关联的 “无关” 状态。

各个比较器 116、120 和 122 中每一个的输出值指示是启动 (由

逻辑“1”表示)还是禁止(由逻辑“0”表示)各个向量(存储在屏蔽存储器 102 中的向量)。所提供的结果还假设,在实现相关转移的结果或状态转换之前已经满足了 IF\_MASK 所定义条件的任何定义的发生值。由此表 1 的结果定义了优先编码器,该优先编码器可用于根据预定级别以及启动或禁止了哪个条件转移来加载状态数据。当不满足任何条件时(例如所有条件都等于 0 时),多路复用器 124 保持其当前状态。本领域的技术人员将了解和理解可实现类似于表 1 中所示功能性的功能性以便基于本文所包含的教学而在包括硬件和/或软件的计算机系统内实现状态机的各种方式。

表 1

转移 1	转移 2	转移 3	结果
0	0	0	加载 CURR_STATE
0	0	1	加载 ELS1_NEXT 状态
0	1	X	加载 ELS2_NEXT 状态
1	X	X	加载 IF_NEXT 状态

在图 3 示例中,“与”门 118 至少部分基于比较器 116、118 和 120 的输出向多路复用器 124 提供多位控制输入 (LOAD\_SEL)。多路复用器 124 接收作为其输入的 IF\_NEXT 向量、ELS1\_NEXT 向量和 ELS2\_NEXT 向量以及从输出寄存器 126 的输出反馈的当前状态 (CURR\_STATE) 信号。多路复用器 124 基于 LOAD\_SEL 信号用输入 (IF\_NEXT、ELS1\_NEXT、ELS2\_NEXT、CURR\_STATE) 中的所选输入加载输出寄存器 126,以便向定义系统 100 的状态 CURR\_STATE 的寄存器 126 提供多位(例如三位输出)信号。也就是说,在满足与 IF\_MASK 相关联的条件时,“与”门 118 提供加载选择信号,用于加载 IF\_NEXT 向量以设置下一个状态。类似地,如果不满足 IF\_MASK 条件而满足了 ELS1\_MASK,则“与”门 118 可提供 LOAD\_SEL 信号来选择对应于到状态机另一个状态的转换的 ELS1\_NEXT 向量。此外,在不满足 IF\_MASK 和 ELS1\_MASK 条件而满足了 ELS2\_MASK 条件时,“与”门 118 将比较器 122 的输出

用于提供 LOAD\_SEL 信号来控制多路复用器 124 选择 ELS2\_NEXT 向量。在不满足任何条件时, LOAD\_SEL 信号可基于作为多路复用器 124 的输入反馈的当前状态来保持其当前状态。这些条件转换在以上表 1 中示出。

5           系统 100 还可包括发生系统 130, 该发生系统操作上用于要求相关的给定条件转移 (例如 IF\_MASK) 的多个命中或发生, 以启动到下一个状态的转换。虽然其它发生要求也可与其它条件转移或状态机电路 106 转移的组合结合使用, 但图 3 的示例假设发生系统 130 仅应用于 IF\_MASK 向量。此外, 虽然状态机电路 106 描述为包括三个状态转移, 但本领域技术人员将了解和理解各种其它类型的转移和其它数量的转移也可用在状态机电路 106 中。

10           发生系统 130 包括计数器 132, 该计数器操作上用于在对于当前状态满足了与 IF\_MASK 向量相关联的条件时对发生进行计数。存储器 102 可经由总线 104 向比较器 134 提供发生值 (OCC\_VAL)。

15           OCC\_VAL 的值定义在 IF\_MASK 向量可启动 “与” 门加载 IF\_NEXT 向量之前要求的一个或多个发生的数目。对于状态机的不同状态, 可编程相同或不同的发生值。比较器 134 比较 OCC\_VAL 和计数器 132 提供的计数器值。比较器 134 基于该比较驱动 “与” 门 136 的输入。

20           “与” 门 136 将反相 “触发” ( $\sim$ TRIGGER) 信号和比较器输出相 “与”, 以提供指示是否已经满足了发生数的输出信号, 如 OCC\_MET 所示。

25           由此 “与” 门 118 将多位输入相 “与”, 该多位输入包括 OCC\_MET 以及两个与用于控制多路复用器 124 的比较器 116、120 和 122 的各个输出有关的反相 “触发” 信号。相应地, 在还未满足 IF\_MASK 向量的发生要求 (例如计数器值小于 OCC\_VAL) 且还未宣称 “触发” 信号时, “与” 门 118 基于 CURR\_STATE 的 ELS1\_MASK 向量和 ELS2\_MASK 向量的应用来恢复。也就是说, OCC\_MET 屏蔽 IF\_NEXT 向量, 直到计数器 132 的输出满足 OCC\_VAL 为止。换言之, 直到



已经满足了与 IF\_MASK 向量相关联的发生要求，状态机的下一个状态才会对应于 ELS1\_NEXT 向量、ELS2\_NEXT 向量和 CURR\_STATE 中的一个。

5 如上所述，含有状态转换的向量的存储器 102 基于可编程输入数据是可编程的，如 PROG\_IN 所示。例如，存储器 102 可实现为系统可寻址存储器（例如寄存器阵列），其可编程为由实现系统 100 的计算机系统的系统处理器或由其它内部或外部工具经由相关接口可寻址的存储器。地址计数器 140 操作上用于提供写地址（ADDR）信号，以将所选项写入存储器 102（其具有来自 PROG\_IN 输入的数据）中。ADDR 可以是定义存储器 102 中地址的多位信号。还向存储器 102 提供写启动输入（WR\_EN），来启动写入指定项。地址计数器 140 可由输入该地址计数器的复位程序（RST\_PROG）启动。可控制地址计数器 140 按步长增加计数，用以基于 STEP\_TRIG\_PROG 输入来存取存储器 102 中的期望项。可由诸如处理器或者能够存取  
10 这种存储器 102 的其它期望接口经由系统可寻址存储器提供 STEP\_TRIG\_PROG、RST\_PROG 和 WR\_EN 输入中的每一个。

在正常操作期间，可基于读地址输入信号（RD\_ADDR 所示）经由总线 104 从存储器 102 中读取数据（RD\_DATA）。RD\_ADDR 输入被提供给多路复用器 142，对应于状态机的 CURR\_STATE 和地址计数器 140 提供的地址 ADDR 中所选的一个。多路复用器 142 基于“触发”信号选择读地址，以便在触发事件发生时，可在读数据总线 104 提供与当前状态相关联的转换信息。  
20

图 4 示出了可与图 3 中所示和描述的分析系统 100 结合使用的触发脉冲发生器 150 的示例。触发脉冲发生器 150 操作上用于提供  
25 “触发”信号和触发延迟信号（TRIG\_DELAY）。系统 150 基于估计各种信号状态的触发逻辑来提供“触发”信号。例如，触发逻辑可包括比较 CURR\_STATE 和预定 FINAL\_STATE 信号的比较器 152。比较器 152 驱动“与”门 154 的输入，该“与”门将比较器输

出和复位触发程序 (RST\_TRIG\_PROG) 信号的反相型式相“与”，该复位触发程序信号可以是存储在系统可寻址存储器中的用户可编程信号。“与”门 154 又驱动“或”门 156 的输入。

“或”门 156 还可接收强制触发 (FORCE\_TRIG) 输入信号，  
5 该信号可用于强制系统 150 宣称“触发”信号。另一个“与”门 158 将反相的 TRIG\_IN/MODE 信号和内触发 (TRIG\_IN) 信号相“与”，以便驱动“或”门 156 的另一个输入。“或”门 156 基于提供给“或”门的各个输入来驱动“与”门 160。此外，TRIG\_IN/MODE 信号和 TRIG\_IN 信号被提供给执行逻辑“或”功能的另一个“或”门 162，  
10 以便向“与”门 160 的另一个输入提供各个输出。由此“与”门 160 基于“或”门 156 和 162 的输出产生“触发”信号。本领域的技术人员将了解和理解可产生触发信号的各种方式，诸如基于期望的性能特性和设计要求。

系统 150 还包括触发延迟发生器 163，该触发延迟发生器操作上  
15 基于“触发”信号、STOR\_QUAL 信号和 TRIG\_DELAY 信号的反相型式来产生 TRIG\_DELAY 信号。例如，触发延迟发生器 163 包括“与”门 164，该“与”门将“触发”信号、TRIG\_DELAY 信号的反相型式和 STOR\_QUAL 信号相“与”。“与”门 164 驱动计数器 166，该计数器基于“与”门 164 的输出而启动。假如没有宣称 TRIG\_DELAY  
20 信号而宣称了 STOR\_QUAL 和“触发”信号（例如对应于限定的触发事件），由此计数器 166 增加其值。计数器 166 可经由 RST\_TRIG\_PROG 信号复位，该 RST\_TRIG\_PROG 信号可由诸如处理器或其它实用程序写入系统可寻址存储器。比较器 168 比较计数器 166 的输出和预定的计数器值 (POST\_STORE 所示)。POST\_STORE  
25 值可以是写入相应系统可寻址存储器用于实现期望触发延迟的预定值。

POST\_STORE 可编程为（例如在图 3 的存储器 102 中）定义触发延迟值，该触发延迟值设置与相应触发事件有关的数据捕获点（例

如在宣称“触发”信号时)。例如，逻辑分析系统的相应捕获缓冲器可捕获数据集，该数据集可基于最小 POST\_STORE 值（例如 POST\_STORE = 0）而在触发事件之前存储在捕获缓冲器中。在此情况下，捕获缓冲器将在触发事件时切断并停止存储来自总线的数据。

5 备选地，POST\_STORE 值可将触发延迟设置为使捕获缓冲器基于对应于捕获缓冲器大小的最大 POST\_STORE 值而在触发事件之后存储所有数据。在后一种情况下，捕获缓冲器将在触发事件之后用来自调试总线的数据填充捕获缓冲器。

另一个备选是基于 POST\_STORE 值将数据集存储在捕获窗口

10 中，该窗口位于上述两种捕获情况之间的任何地方。比较器 168 驱动“或”门 170 的输入。“或”门 170 的其它输入包括“触发”信号和 FORCE\_TRIG 信号。由此“或”门 170 基于比较器 168 的输出、FORCE\_TRIG 信号和“触发”信号而产生 TRIG\_DELAY 信号。可将 TRIG\_DELAY 信号与 STOR\_QUAL 信号一起提供给捕获缓冲器，

15 用于控制诸如本文所述的捕获缓冲器的操作。

图 5 示出了逻辑分析器系统 200 的示例。系统 200 用于从数据总线 202 获取数据。数据总线 202 例如可接收来自集成电路芯片中一个或多个源或者来自实现系统 200 的相关装置中任何地方的数据。本领域的技术人员将了解和理解可用于驱动具有数据的总线 202 的

20 各种方法和馈电结构。例如，数据总线 202 可操作为同步总线结构，该同步总线结构配置为传播来自实现系统 200 的集成电路中一个或多个预定位置的多位数据。在多处理器、多单元计算机系统中，例如总线 202 还可接收来自其它集成电路和其它电路板的、经由诸如纵横结构提供的数据。

25 监控系统 204 接收并监控在总线 202 上提供的数据。监控系统 204 可包括多个编程为和/或配置为基于在总线 202 上传播的数据来确定是否已经满足了某些性能条件的性能监控器/计数器。例如，监控系统 204 可配置为实现与总线 202 上数据子集有关的算术运算、逻辑

运算和匹配运算及其组合。监控系统 204 可提供相应的多位输出 (TRIG\_OUT\_LIST)，该多位输出指示正在监控的每个性能条件的结果。例如，监控系统 204 可在每个时钟周期宣称 TRIG\_OUT\_LIST 信号中的相应输出位，在每个时钟周期内都满足一些或所有数据总线 202 预定子集的给定条件。例如可响应于监控系统的相应性能计数器的值在给定时钟周期的增加来宣称 TRIG\_OUT\_LIST 中的每位。

通过写入相关的存储器 206，性能条件是可编程和定义的。相关存储器 206 可以由一个或多个程序（输入）信号可编程的计算机系统（例如控制和状态寄存器阵列（CSR））内的一个或多个系统可寻址存储块。输入信号可用于设置监控系统 204 要对总线 202 上的数据执行的期望逻辑、匹配和/或算术运算。存储器 206 可提供（或者监控系统 204 可读取）PROG\_MON 信号，以为监控系统 204 所监控的每个性能条件编程性能条件。可存在与编程监控系统 204 要估计的每个性能条件相关联的存储器 206 的单独块。例如，可由内部存储器（例如经由系统可寻址存储器）或从外部装置或系统实用程序，通过写入分配给监控系统 204 各个性能监控电路的存储器 206 中的预定地址位置，来对存储器 206 中的相应块进行编程。

监控系统 204 向限定系统 208 和分析系统 210 提供 TRIG\_OUT\_LIST 信号。TRIG\_OUT\_LIST 信号可提供为多位总线上的数据，其包括监控系统 204 监控的每个性能条件的各个输出。例如，在满足监控系统 204 正在实现的特定条件时，可由监控系统 204 在时钟周期内宣称 TRIG\_OUT\_LIST 信号中的相应位（或多位）。TRIG\_OUT\_LIST 信号中相应位（或多位）的宣称可对应于增加监控系统 204 各个性能监控电路中的相应计数器或其它跟踪电路。由此，多位输出 TRIG\_OUT\_LIST 提供关于在总线 202 上提供的数据中是否已经满足某些条件的指示，并且另一个信号 212 可提供随时间与这种性能相关联的值。本领域的技术人员将了解和理解，监控系统 204 可编程和配置为监控与总线 202 上的数据相关联的一个或多个条件

中的任意个。

限定系统 208 相对于监控系统 204 提供的 TRIG\_OUT\_LIST 数据执行匹配和限定功能。限定系统 208 向相关数据捕获系统 214 提供 STOR\_QUAL 信号，以便标识是否应从数据总线 202 中捕获数据。

5 例如，可经由 PROG\_SQ 信号将限定系统 208 编程为诸如对与编程数据有关的所选一组或一小组 TRIG\_OUT\_LIST 数据执行限定逻辑或匹配功能。例如，匹配功能可实现可匹配的屏蔽功能，该功能确定是否应在每个时钟周期基于由 TRIG\_OUT\_LIST 信号表示的变量的结果来从数据总线捕获数据。由此，匹配功能可提供 STOR\_QUAL

10 信号，以标识与监控系统 204 正在监控的性能条件的结果相关联的一个或多个模式。

分析系统 210 配置为提供状态机，该状态机基于来自监控系统 204 的 TRIG\_OUT\_LIST 数据所指示的性能信息来执行总线上数据的内部逻辑分析。分析系统 210 提供“触发”信号和触发延迟

15 (TRIG\_DELAY) 信号，以便控制用于从总线 202 获取数据集的捕获会话。例如，分析系统 210 可实现为状态机结构（例如 Mealy 或 Moore），该状态机结构基于监控系统 204 所实现的性能条件在状态之间转换。如本文所述，当性能条件满足时，可在时钟周期内宣称 TRIG\_OUT\_LIST 中的各个数据，以便启动分析系统 210 所要执行的

20 逻辑分析。分析系统 210 可基于 TRIG\_OUT\_LIST 信号和 STOR\_QUAL 信号向数据捕获系统 214 提供 TRIG\_DELAY 信号。如上所述，还可将“触发”信号提供给限定逻辑块 208。

分析系统 210 可用向量 (PROG\_TRIG) 来配置，该向量基于实现的条件来定义相关电路要用于分析 TRIG\_OUT\_LIST 的一组条件，

25 以及可能发生的一组可能状态转换。分析系统 210 还可采用具有附加状态转换的条件转移，对于每个转移，该状态转换可基于 TRIG\_OUT\_LIST 数据以及基于分析系统 210 实现的状态机的当前状态而变化。当分析系统 210 转换到分析系统的一个或多个可编程状

态（该状态设计为使“触发”信号宣称）时，触发事件或条件可能发生。

例如，状态机可包括使分析系统 210 宣称“触发”信号的“最终状态”。此外，在转换到与给定条件相关联的下一个状态之前，  
5 可要求给定条件的一个或多个发生中的预定个。例如，可将值编程（例如经由 PROG\_TRIG 信号）为设置与至少一部分（例如一个或多个）TRIG\_OUT\_LIST 数据相关联的给定条件的发生数，其必须被满足以便对于给定条件启动到下一个状态的转换。也可存在可编程构件，以强制分析系统 210 宣称“触发”信号。

10 分析系统 210 还可定义与预定事件的发生有关的数据集。在示例性逻辑分析系统 200 中，预定事件对应于正在宣称“触发”信号，当状态机转换到其多个状态中的给定状态时，这可能发生。分析系统 210 可基于与“触发”信号有关（例如之前、之后或与之重叠）的 STOR\_QUAL 信号来对多个限定存储周期进行计数或跟踪。分析系统 210 又响应于对预定数量存储周期的计数或跟踪来提供  
15 TRIG\_DELAY 信号，同时相对于“触发”信号的宣称而启动。

在另一个示例中，诸如在宣称“触发”信号时，分析系统 210 可编程为（例如经由 PROG\_TRIG 信号）相对于触发点来调节数据捕获的定时。例如，该定时可编程为：定义在触发事件发生之前、  
20 在触发事件发生之后或在包括触发事件的某窗口之内捕获缓冲器是否要存储数据。例如，该窗口可根据数据捕获系统 214 所采用的一个或多个缓冲器或与用于存储来自总线 202 的数据的一个或多个缓冲器共同使用的其它存储器的大小而变化。

25 数据捕获系统 214 操作上用于至少部分基于来自限定逻辑的 STOR\_QUAL 信号并基于由分析系统 210 提供的“触发”和 TRIG\_DELAY 信号来存储来自总线 202 的数据。数据捕获系统 214 包括捕获缓冲器控制逻辑，该逻辑可设置为定义要存储的数据量、要存储的数据类型以及将如何存储数据。例如，数据捕获系统 214

的控制逻辑可包括配置为激活数据捕获系统 214 用以响应于 STOR\_QUAL 和 TRIG\_DELAY 信号来读取和存储来自总线 202 的数据的硬件配置。数据捕获系统 214 可向可由系统处理器读取的相关存储器诸如系统可寻址存储器提供其相应的输出信号 (OUT)。

5           本领域的技术人员将理解可用于将程序数据输入到系统 200 各个部分以及用于存储来自系统 200 的输出 OUT 数据的各种类型存储器结构 (例如寄存器阵列、缓冲器、RAM、高速缓存等)。此外, 包括监控系统 204、限定系统 208、分析系统 210 和数据捕获系统 214 (或者其中至少一部分) 的系统 200 可实现为专用集成电路 (ASIC) 的一部分。ASIC 可实现为芯片组 (诸如部分计算机系统、路由器或其它复杂电路) 内部的集成逻辑分析器。

10           图 6 示出了可用于监控与总线 252 (诸如可观测性总线) 上的数据相关联的性能特性的性能监控系统 250 的示例。性能监控系统 250 可实现为在 IC (诸如组成计算机系统、路由器或其它电子系统的一部分芯片组的 IC) 内实现的一部分逻辑分析器系统。性能监控系统 250 包括多个表示为性能监控计数器 (PMON/计数器 0 和 PMON/计数器 1 直到 PMON/计数器 N) 254 的子系统, 其中 N 是正整数, 而 N+1 表示 PMON/计数器 254 的数量。PMON/计数器 254 共同驱动对应于 TRIG\_OUT\_LIST 指示的多位输出信号的输出总线 256。由此, 输出总线 256 可包括 N+1 位, 一位与每个 PMON/计数器 254 相关联。

15           每一个 PMON/计数器 254 可实现为可编程逻辑的配置, 诸如可编程逻辑器件 (PLD)、现场可编程门阵列、其它硬件或硬件和软件的组合。每个 PMON/计数器 254 可编程为实现在总线 252 上选择的部分数据或子范围数据的运算或功能。例如, 每个 PMON/计数器 254 可实现与来自总线 252 的一个或多个选定位有关的匹配功能。PMON/计数器 254 还可对总线 252 上的一位或多位实现逻辑功能 (例如 “反”、“与”、“或”、“异或”、“或非”、“同”以及其它逻辑功能和功能的组合)、算术功能 (例如加、减、乘、除等) 以

及逻辑和算术功能的组合。

5 系统可寻址存储器 258 操作上与每一个 PMON/计数器 254 相关联，以便对要相对于总线 252 上的数据执行的期望运算或功能进行编程。系统可寻址存储器 258 可由系统处理器 270 以及相关诊断实用程序（未示出）或者能够写入系统可寻址存储器 258 的其它装置存取。系统可寻址存储器 258 中的数据对各个 PMON/计数器 254 中每一个执行的特定运算或功能进行编程。

10 在图 6 示例中，PMON/计数器 0 示为包括条件块 260 和计数器 262。条件块 260 实现关于数据总线 252 上一个或多个所选数据位的性能条件，该条件可包括对数据执行运算或功能，诸如算术功能、逻辑功能或逻辑和算术功能的组合。可根据来自系统可寻址存储器 258 的 PROG\_PMON\_0 信号对 PMON/计数器 0 执行的特定逻辑和/或算术功能进行编程。PROG\_PMON\_0 信号还可确定要对来自总线 252 的哪个数据实现性能条件，诸如通过标识这种数据的各个地址的方式。

15 例如，PROG\_PMON\_0 信号可包括设置性能条件（例如逻辑功能和/或算术运算）的一位或多位，对从总线 252 选择的数据执行该性能条件。条件块 260 基于对数据的函数或运算应用来向计数器 262 提供条件信号（PMON 0）264。条件块 260 可在每个时钟周期或其它选定时间间隔执行性能条件。当性能条件满足时，条件块 260 诸如在一个或多个时钟周期上宣称对应于 PMON 0 的其输出 264（例如时钟周期内的逻辑“高”）。例如，如果在多个时钟周期上满足性能条件，则条件块 260 可在多个时钟周期上将 PMON 0 保持在宣称的状态。备选地，条件块 260 可触发 PMON 0 输出信号。PMON 0 20 对应于形成 TRIG\_OUT\_LIST 信号的部分输出总线 256。可响应于在每个时钟周期增加各个 PMON/计数器 254 的值来宣称 TRIG\_OUT\_LIST 中的每位。TRIG\_OUT\_LIST 又可用于驱动本文所述的触发状态机。作为双重使用 PMON/计数器 254 的结果（例如驱



动各个性能计数器值并驱动触发状态机)，可获得硬件的有效使用。

5 输出条件信号 PMON 0 还可调节与条件块正在监控的数据相关联的性能的测量。在图 6 示例中，监控的 PMON 0 根据在给定时钟周期内是否满足条件块 260 实现的性能条件来增加（或减少）计数器 262。计数器 262 提供 PCOUNT 信号，该信号的值指示各个性能监控子系统所监控的性能的测量。例如，PCOUNT 信号的值可指示诸如在给定捕获会话期间或多个会话上满足条件块 260 所实现的性能条件的次数。如果需要，可将计数器 262 复位。

10 虽然会理解可与关于 PMON/计数器 0 所示和所述的类似地配置 PMON/计数器 1 到 PMON/计数器 N 中的每一个，但出于简化说明的目的，从图 6 中省略了其它 PMON/计数器 1 到 PMON/计数器 N 的内部内容。也就是说，每个 PMON/计数器 254 可编程和/或配置为执行各个性能条件，这些性能条件基于是否满足该条件来驱动相关计数器。每当基于性能条件增加（或减少）计数器时，还在总线 256  
15 上的 TRIG\_OUT\_LIST 中（例如在时钟周期内）宣称从各个 PMON/计数器 254 输出的相应 PMON。由此与 TRIG\_OUT\_LIST 信号相关联的总线 256 上 N 位中的每一位都根据每一个 PMON/计数器 254 中条件块所实现的性能条件来提供与总线 252 上所选数据部分相关联的性能指示。虽然 PMON/计数器 254 已被描述为可编程的，但还考  
20 虑到可以硬连线一个或多个 PMON/计数器 254，以便实现固定的性能监控条件。

系统 250 还可包括另一个通用计数器 266，该计数器增加计数器  
值以在每个时钟周期（或在某其它周期间隔上）提供参考计数信号。  
由此计数器 266 的值可与来自计数器 262（以及其它 PMON/计数器 254  
25 的计数器）的 PCOUNT 信号进行比较或估计，以确定满足了条件块  
260（以及其它 PMON/计数器 254 的其它条件块）所实现的各个性能  
条件的频率指示。例如，当处理器 270 执行对应于诊断实用程序的  
指令时可采用计数器。计数器 266 的值还可用于控制一个或多个

PMON/计数器 254 的操作。

图 7 描述了图示计算机系统 350 示例的框图，其可实现一个或多个逻辑分析器系统，诸如包括本文（例如图 1-6 和图 8）示出和描述的系统 and 部件的逻辑分析器系统。虽然单处理器系统也可使用逻辑分析器，但图 7 的计算机系统 350 示为分布式存储器多处理器系统。系统 350 包括分别在单元 1、单元 2 到单元 M 所示的多个单元 352，其中 M 是大于或等于 1 的整数，表示单元的数目。可实现为单元板的每一个单元 352 经由互连 354（诸如背面或纵横结构）通信地耦合到其它单元。互连 354 可实现为专用集成电路（ASIC）。

在图 7 示例中，在互连 354 内实现一个或多个逻辑分析器 356；即，第一互连中的一个逻辑分析器和另一互连中的两个逻辑分析器。本领域技术人员将了解和理解，可在互连 354 内以及包括集成电路上单元 352 或 I/O 子系统 358 中的其它电路中实现一个或多个逻辑分析器中的任意个。通过示例，每个逻辑分析器 356 耦合到各个互连 354 内的总线结构（例如可观测性总线），可用来自一个或多个单元 352 内部件的数据驱动该互连。此外，如本文所述，在系统 350 内每个逻辑分析器 356 可包括可寻址存储器，该存储器可由任一个相关单元 352 上的部件读取或写入。

通过另一个示例，I/O（输入/输出）子系统 358 与每一个单元 352 联系在一起。I/O 子系统 358 可提供接口或路径，用于诸如通过相应的适配器（未示出）来存取相关总线结构（例如 PCI 总线结构）或耦合到相应总线结构的其它装置。本领域技术人员会了解和理解可被存取或可经由 I/O 子系统 358 存取存储器的各种类型 I/O 装置 358。

此外，包含一个逻辑分析器 356 的互连 354 可耦合到包含两个逻辑分析器的另一个互连，用以存取包括一个或多个其它单元（未示出）的另一个基于单元的结构。可与图 8 中示出和描述的相类似地配置另一个基于单元的结构（未示出）。本领域技术人员会了解和理解，可用任何数量的单元、用正在实现的一个或多个逻辑分析

器中的任意个来实现系统 350。

虽然本领域技术人员会了解和理解能以类似方式实现其它各个单元 352 中的每一个，但出于简洁目的，仅示出了单元 1 的内部内容。备选地，也可相对于不同单元 352 实现不同的配置。

5            转到单元 1 的内容，单元 1 包括单元控制器 360，其通过相关缓冲网络 364 耦合到单元存储器子系统 362。缓冲网络 364 可包括队列（例如输入队列和输出队列），以提供存储器子系统 362 和控制器 360 之间请求和响应的智能缓冲。一个或多个中央处理器（CPU）366 也连接到控制器 360，用于存取存储器子系统 362。每一个 CPU 366 可  
10           包括相关高速缓存（未示出），该高速缓存用于在不需存取存储器子系统 362 的情况下存储由 CPU 局部存取的数据。在图 8 所示的配置中，CPU 366 和 I/O 子系统 358 中的每个都可看作是操作上经由控制器 360 存取存储器子系统 362 中数据的存储器存取装置。控制器 360 可包括固件、配置和状态寄存器（CSR）以及用于存取存储器  
15           子系统 362 中数据的有序存取队列。存储器子系统 362 可包括一个或多个存储器模块（包括一个或多个 DIMM 或 SIMM 存储器件）中的任意个。

          当 CPU 366 和/或 I/O 子系统 358 存取数据时，控制器或其它结构可将这种数据中所选部分或全部驱动到与一个或多个逻辑分析器  
20           356 相关联的可观测性总线。逻辑分析器 356 又能监控相关可观测性总线上的数据，基于该监控来限定数据，并基于数据的限定来捕获数据。逻辑分析器还可实现状态机，该状态机包括一个或多个条件，该条件控制状态转换以及给定的数据捕获会话如何进行，诸如本文所述的。在一个示例中，状态机可实现为硬件（例如门或其它电路  
25           的配置），其能以与驱动包括逻辑分析器的集成电路的系统时钟基本相当的速率在多个状态之间转换。

          还将理解到，可由在一个或多个 CPU 366 中运行的计算机可执行指令来以程序方式发起和控制一个或多个逻辑分析器 356 的数据

捕获会话。备选或附加地，可由实用程序或诊断工具发起和控制捕获会话。例如，实用程序或诊断工具可内部运行在 CPU 366 内或作为一个 I/O 子系统 358 的一部分在外部运行。本领域技术人员将了解和理解基于本文包含的教学在计算机系统 350 中可采用的逻辑分析器的各种实现。

考虑到上述的前述结构和功能特征，参考图 8 将更好地理解某一方法。要了解和理解到，在其它实施例 10 中，图示的动作可以不同的顺序发生，和/或其它动作同时发生。此外，可不要求所有图示特征都实现方法。还应理解到，可用硬件（例如包括诸如晶体管、数字信号处理器或专用集成电路的逻辑门）、软件（例如运行在一个或多个处理器上的可执行指令）或硬件和软件的任意组合来实现以下方法。

图 8 示出了可通过集成电路实现的方法 400 的示例。方法 400 包括用状态数据和条件数据对部分集成电路进行编程，该状态数据定义状态机的多个可能状态，而条件数据定义用于控制多个可能状态之间转换的一组条件，如 410 所示。相对于输入数据应用该组条件中的至少一个条件（420 处），所应用的至少一个条件与当前状态相关联。在 430 处，基于与当前状态相关联的至少一个条件的应用来选择多个可能状态中的下一个状态。在 440 处，基于对应于多个可能状态中预定状态的下一个状态来产生触发信号。

以上描述的是本发明的示例。当然，不可能为了描述本发明而描述部件或方法的每个可想到的组合，但是本领域的一般技术人员将认识到，本发明的许多其它组合和变更也是可能的。例如，可在给定的 ASIC 中实现用于控制数据捕获的一个或多个系统中的任意个，并可 25 将任意个这种 ASIC 集成到计算机系统、路由器或其它类型的电子电路系统中。因此，本发明意图包含落入所附权利要求书的精神和范围内的所有这种更改、修改和改变。

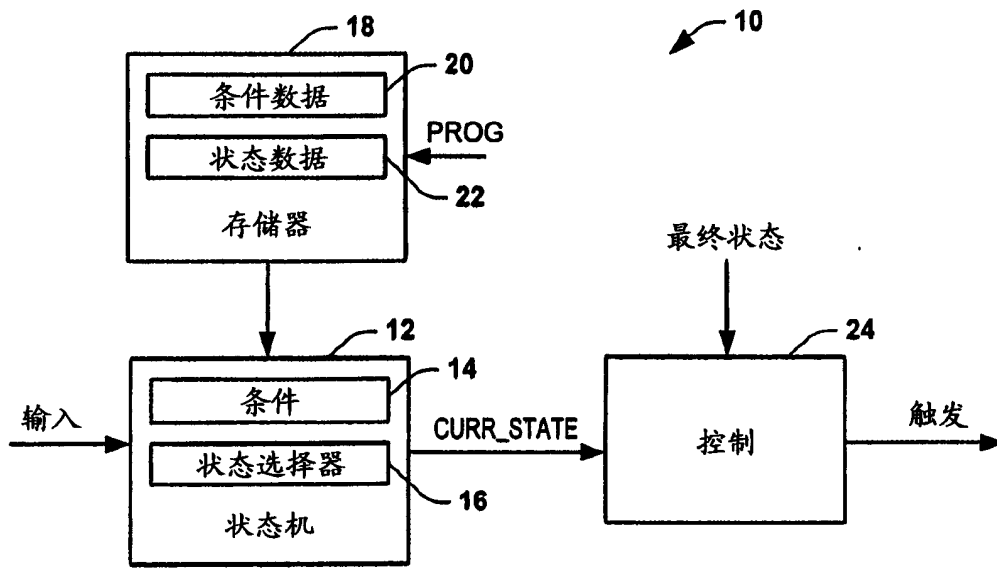


图 1

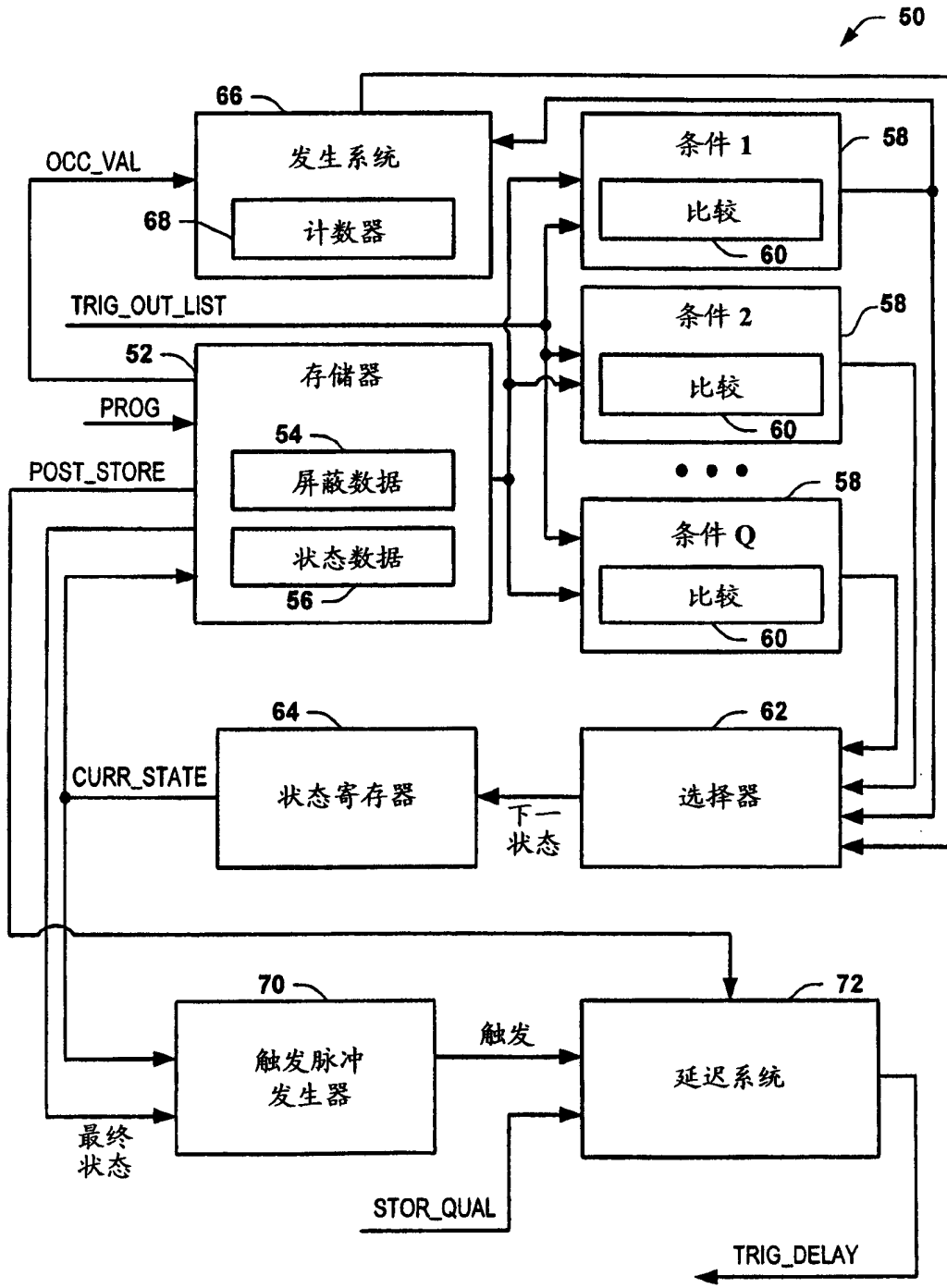


图 2

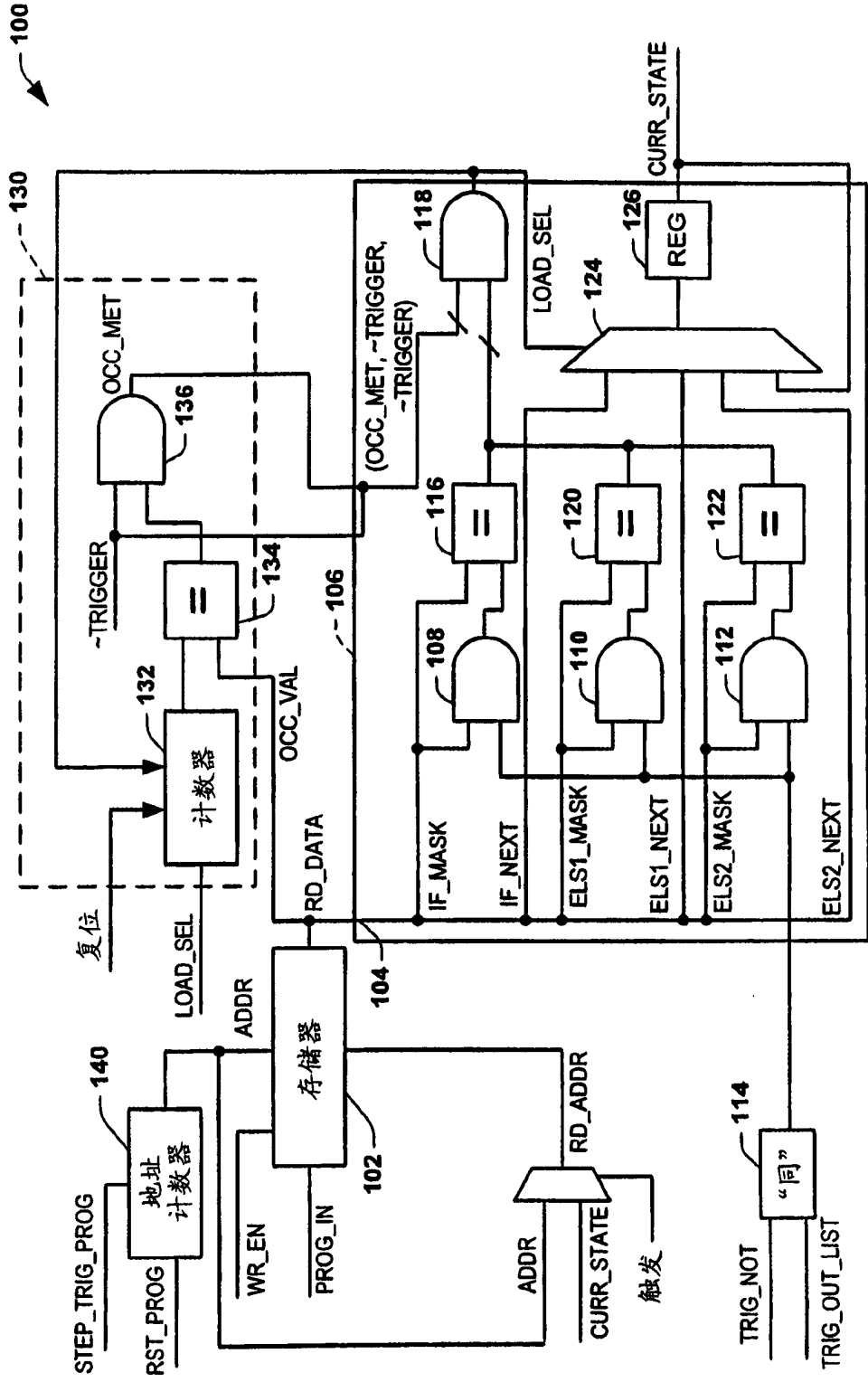


图 3

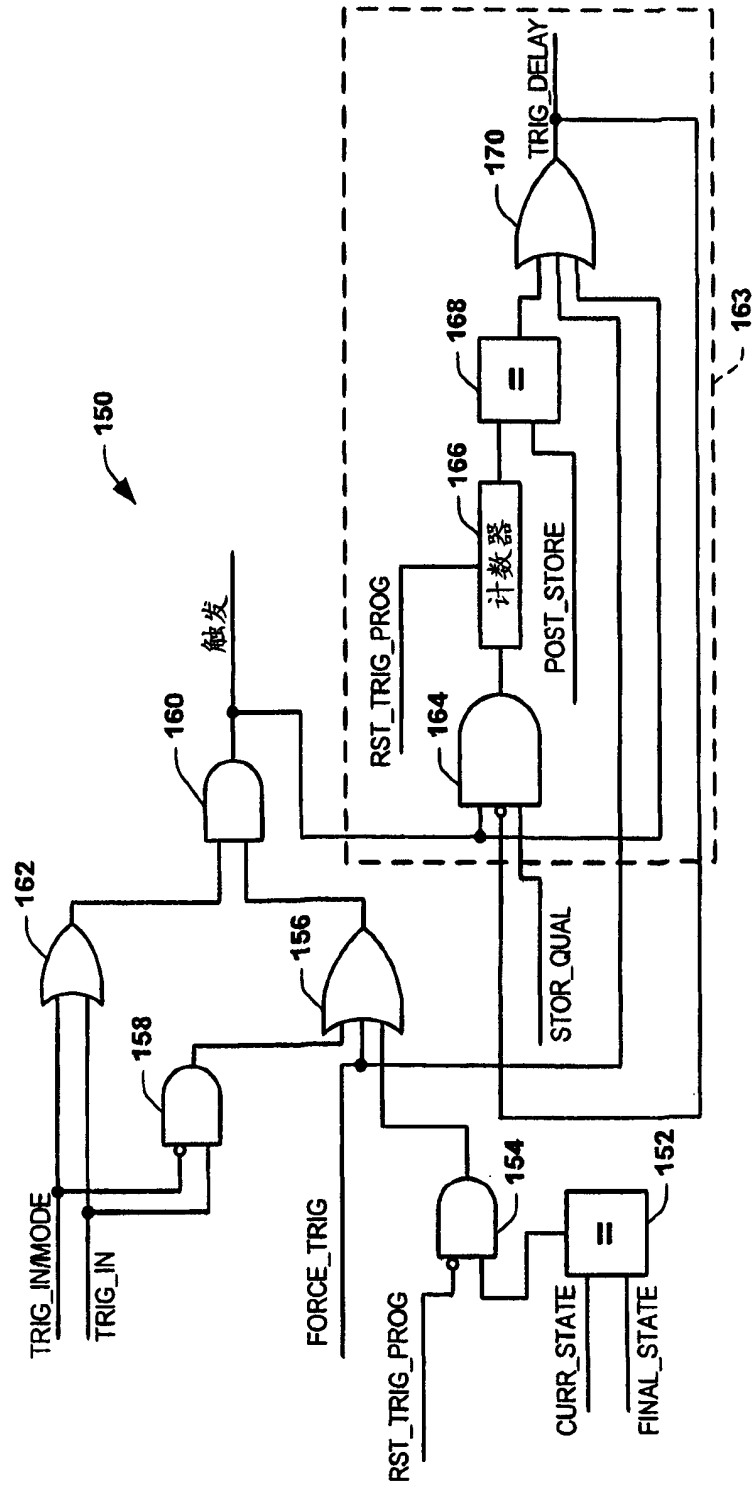


图 4



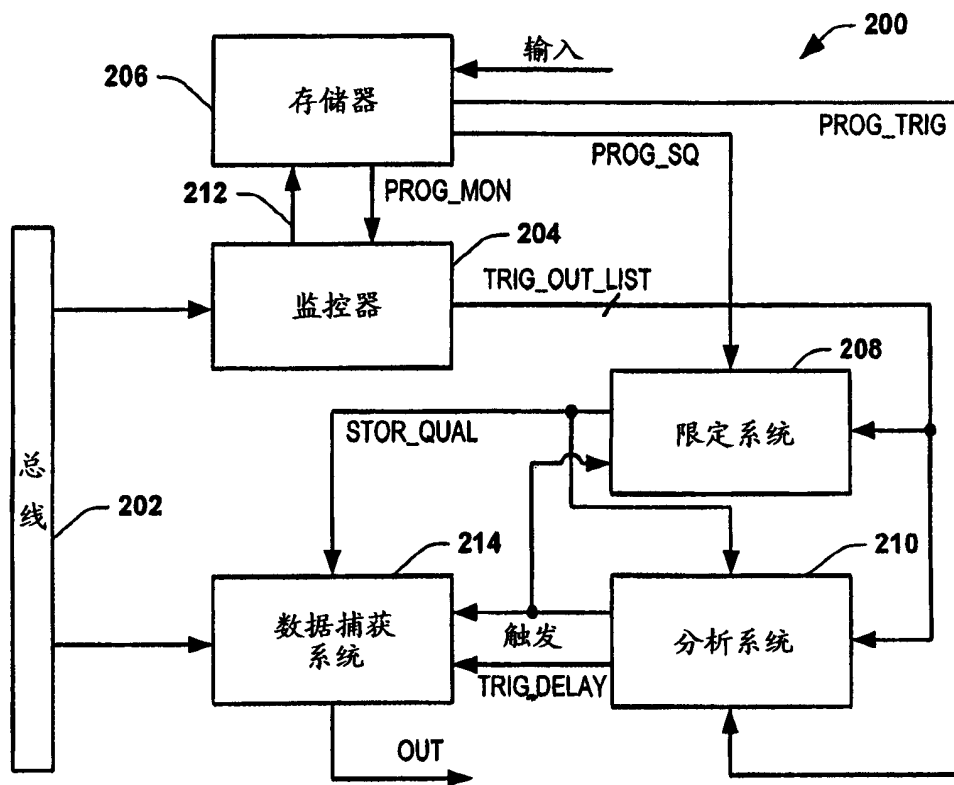


图 5

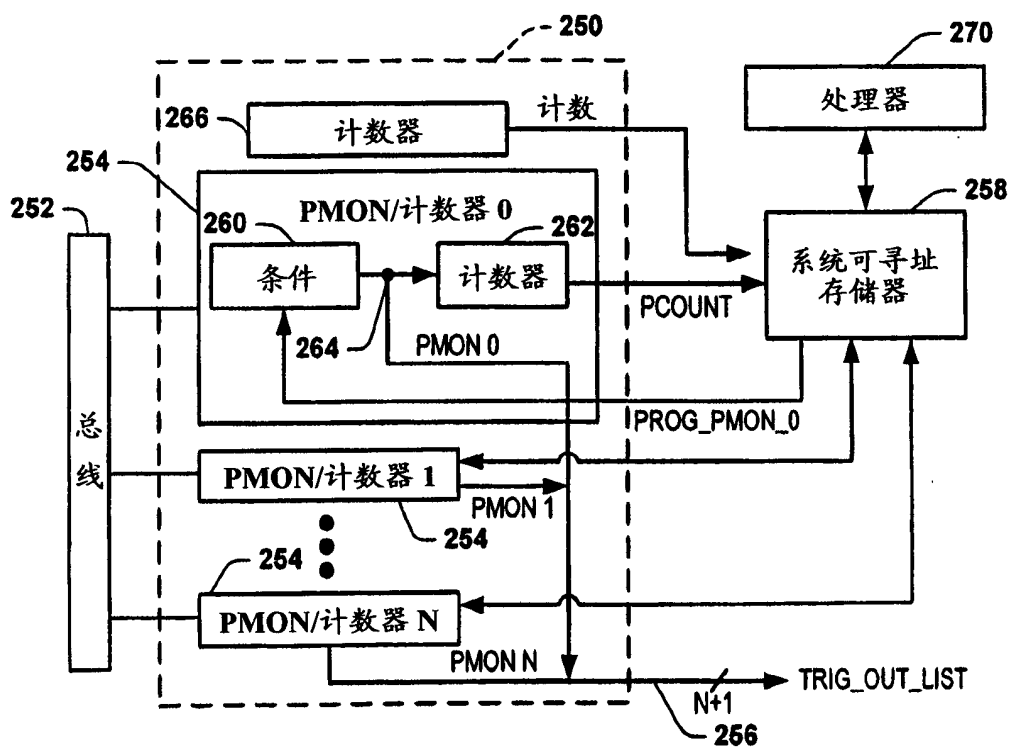


图 6

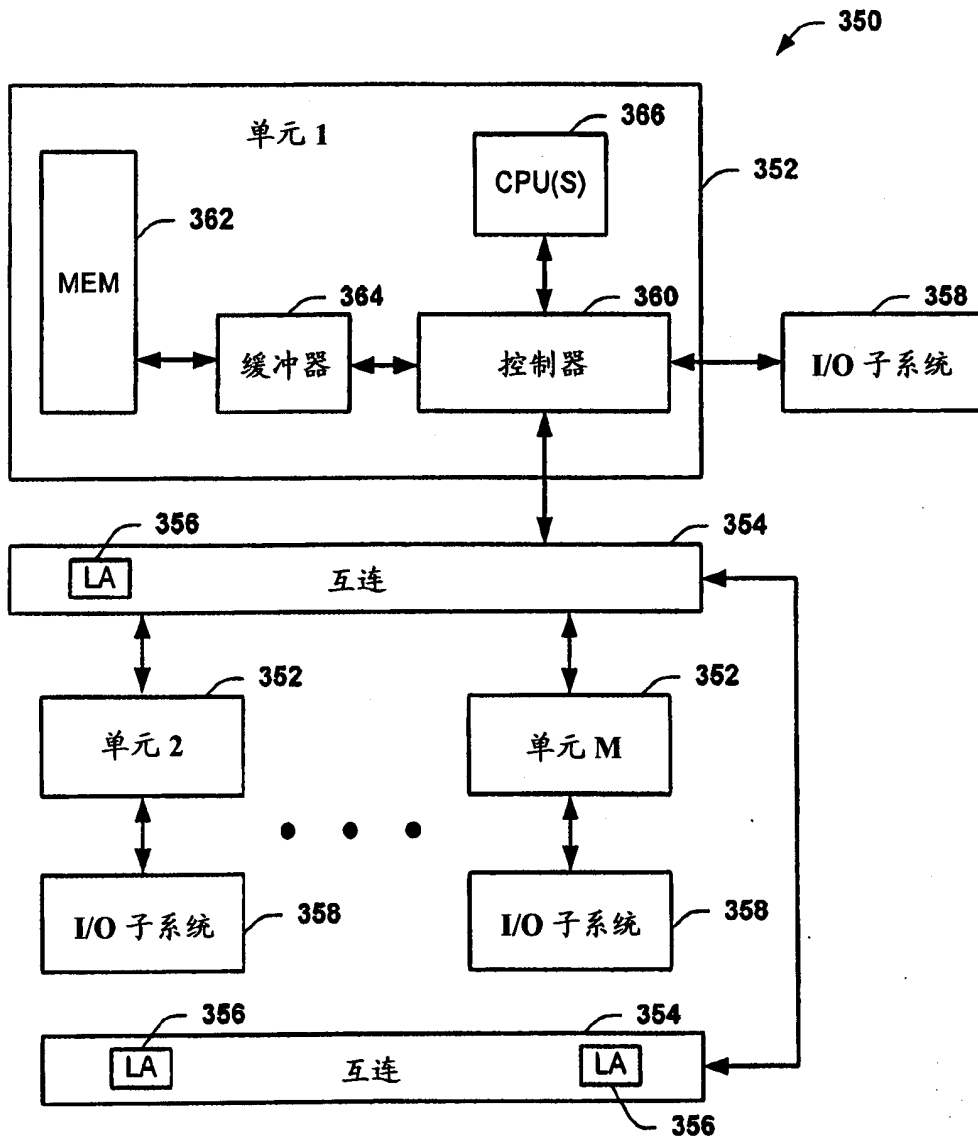


图 7

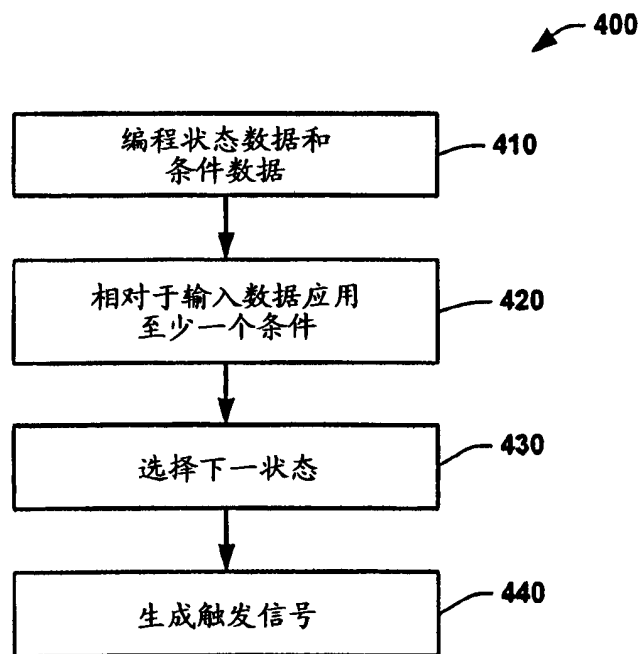


图 8