



[12] 发明专利申请公开说明书

[21] 申请号 89108771.0

[51] Int.Cl⁵

G06F 11/22

[43] 公开日 1990年6月13日

[22] 申请日 89.11.21

[30] 优先权

[32] 88.11.23 [33] US [31] 275,185

[71] 申请人 约翰弗兰克制造公司

地址 美国华盛顿州

[72] 发明人 布鲁斯·T·怀特

约翰·D·波尔斯特拉

克雷格·V·约翰逊

[74] 专利代理机构 中国专利代理有限公司

代理人 何耀煌 李先春

说明书页数: 15 附图页数: 4

[54] 发明名称 验证内核数据总线的设备、方法和数据结构

[57] 摘要

用于验证基于微处理器的被测部件的数据总线的设备、方法和数据结构,其中,将其位数为数据总线宽度一半的位模式与另一位模式(或与所述位模式互补或与所述位模式相同)一起施加到数据总线上。对数据总线上导出的位模式的评价使得对数据总线全部宽度的验证成为可能,如果报告无故障的话,这不仅省去了由操作者对数据总线的探测,而且也省去了数据总线诊断。最佳位模式序列的特定的数据结构避免了将任何数据线上的故障报告为合格。

<39>

权 利 要 求 书

1. 用存储器仿真对基于微处理器被测部件(UUT)的数据总部进行验证的方法, 其特征在于该方法包含以下步骤:

a) 执行装入操作, 将第一位模式加到所述数据总线的第一部分线路上, 所述第一位模式包含若干位, 该位数等于所述数据总线至少部分的线路数的一半,

b) 执行装入操作, 将第二位模式加到所述数据总线第二部分上, 该第二部分与总线线路的第一部分线路互补, 所述第二位模式与所述第一位模式互补,

c) 比较所述第一和第二位模式, 以确定所述第二位模式是否与所述第一位模式互补, 或者所述第二位模式不与所述第一位模式互补,

d) 执行装入操作, 将第三位模式加到所述数据总线第三部分线路上, 所述第三位包含若干位, 该位数等于所述数据总线至少一部分线路数的一半,

e) 执行装入操作, 将第四位模式加到所述数据总线第四部分线路上, 该第四部分是总线线路的第三部分的互补, 所述第四位模式不与所述第三位模式互补,

f) 比较所述第三和第四位模式, 确定所述第四位模式不与所述第三位模式互补, 以及

g) 根据步骤c)和f)的结果验证所述数据总线。

2. 根据权利要求1的方法，其特征在于：步骤 a) — c) 及步骤 d) — f) 为重复多次的序列。

3. 根据权利要求2的方法，其特征在于，在所述序列中各重复的第一位模式为：

\$FFFF0000	\$00FFFF00
\$0F0FF0F0	\$3333CCCC
\$5555AAAA	\$AAAA3333
\$CCCC3333	\$F0F00F0F
\$FF0000FF	\$0000FFFF

4. 根据权利要求2的方法，其特征在于，所述序列中各重复的第三位模式为：

\$FFFFFFFF	\$00FF00FF
\$0F0F0F0F	\$33333333
\$55555555	\$AAAAAAAA
\$CCCCCCCC	\$F0F0F0F0
\$FF00FF00	\$00000000

5. 如权利要求1所述的方法，其特征在于，所述第四位模式和所述第三位模式相同。

6. 用存储器仿真对基于微处理器的被测部件(UUT)的数据总线进行测试的设备，其特征在于包括：

第一装置，该装置用于将第一位模式通过所述数据总线的第一部分线路加到所述数据总线上，以及将第二位模式通过与所述数据总线线路的第一部分互补的第二部分总线线路加到所述数据总线上，

所述第一位模式包含若干位，该位数等于所述数据总线至少一部分线路数的一半，所述第二位模式与所述第一位模式互补，

第二装置，该装置用于将第三位模式通过所述数据总线的第三部分线路加到所述数据总线上，以及将第四位模式通过与所述数据总线线路的第三部分互补的第四部分总线线路加到所述数据总线上，所述第三位模式包含若干位，该位数等于所述数据总线至少一部分线路数的一半，所述第四位模式不与所述第三位模式互补，以及

用于比较所述第一和第二位模式以及比较所述第三和第四位模式的装置，如果所述第二位模式不与所述第一位模式互补以及如果所述第四位模式不等于所述第三位模式之反，则该装置产生一信号，以及

用于对由所述比较装置产生的信号进行评价以验证所述数据总线的装置。

7. 根据权利要求 6 的设备，其特征在于：所述施加装置将第一位模式序列中的每一个以及第三位模式序列中的每一个置于所述数据总线上。

8. 根据权利要求 7 的设备，其特征在于：所述序列中的第一位模式为：

\$FFFF0000	\$00FFFF00
\$0F0FF0F0	\$3333CCCC
\$5555AAAA	\$AAAA5555
\$CCCC3333	\$F0F00F0F
\$FF0000FF	\$0000FFFF

9. 根据权利要求 7 的方法，其特征在于，所述序列中的第三位模式为：

\$FFFFFFFF

\$00FF00FF

\$0F0F0F0F

\$33333333

\$55555555

\$AAAAAAAA

\$CCCCCCCC

\$F0F0F0F0

\$FF00FF00

\$00000000

10. 根据权利要求6的设备，其特征在于：所述第四位模式和所述第三位模式相同。

验证内核数据总线的设备、方法 和数据结构

本发明一般涉及基于微处理器的电子系统的测试和故障查找，更准确地说，涉及使用存储器仿真技术对基于微处理器的电子系统的内核进行的测试和故障查找。

本申请与以下和本申请同时提交的申请相关，即，J. Polstra, M. Scott 和 B. White 的“系统自动诊断的内核测试接口和方法”，T. Locke 的“提高存储器仿真法性能的增强型硬件”，以及 J. Polstra 的“基于存储器存取分析的内核电路自动验证”，这些都已转让给本发明的受让人。

随着在消费和工业产品中对复杂的基于微处理器的系统的广泛使用，电路故障的测试和诊断和自动化，特别是对这种系统的内核进行电路故障的测试和诊断的自动化，变得极为需要，在本领域中众所周知，这种系统的内核是指微处理器(μP)本身以及相关元件，为了能够正常运行，微处理器与这些元件（具体是指存储器、时钟、地址总线 and 数据总线）的正确配合是必需的。其中用测试设备仿真内核元件的所谓仿真测试器，由于即使在该内核连最低限度运转也达不到的情况下，也能够对所述内核作较详细的诊断，所以已越来越普及地用于功能测试。

一种类型的仿真测试器就是微处理器仿真器，作为授给 K. S.

Bhaskar等人的美国专利4,455,654中所描述的测试器的实例, (该专利已转让给John Fluke制造有限公司); 在该系统中, 通过拆去被测部件(UUT)的微处理器(μ P)并通过UUT的 μ P插孔连接该测试系统而实现其与UUT的连接。

另一种类型的仿真式测试器是ROM(或存储器)仿真器。由于ROM可与UUT数据和地址总线直接通信, 而且ROM插座管脚结构比较简单, 所以认为ROM仿真是合乎要求的。ROM仿真器用于软件设计和 μ P的操作验证是众所周知的, 只是最近才应用于故障检测和诊断, 因为一般没有同步信号可运用于使该测试设备与它所接收到的测试结果相同步。对该问题的一个解决办法, 公开于M. H. Scott等人于1988年2月19日提交的美国专利申请07/158,223, 即“基于微处理器的电子系统测试和故障查找的存储器仿真方法和系统”, 特此引入作为对比文件。该测试系统包含一个基于微处理器的主机和一个接口箱, 后者还包含一个与所述 μ P和UUT存储器插座相连接的基于 μ P的系统。该接口箱包含与UUT μ P相连专门逻辑电路, 用以在所考虑的总线周期内提供高分辨同步信号, 以便产生完全的故障查找故障隔离, 也就是说, 由 μ P导出的高分辨同步脉冲可用来隔离和评价在存储器插座上通过地址和数据总线观测到的信号。这样做和通过 μ P接点来隔离和评价同样的方便。还有, 如该申请所公开内, ROM仿真可归纳为存储器仿真(例如, 任何存储器或部分存储器的仿真)。因为基于 μ P系统的趋势是增加RAM而减少ROM并有可能用RAM取代而全部放弃ROM。所以测试系统必须恰当地归纳为尚未产生的但从基于 μ P的电子系统结构的发展趋势来看还是可以预见与测试系统。

重要的是理解用于指出由不同测试过程所提供不同程度的功能

确定性的各术语之间的差别。术语“验证”在这里用于指出对至少具有足以进行后继过程的最低级功能性的断定。术语“证实”(validation)指出如未发现故障，那么所证实的整个结构可认为是完全起作用的。术语“测试”用来指出一个过程，即可以发现所有存在的故障但不必隔离或识别。“诊断”在这里指出发现并识别出所有故障。

如上面提及的共同未决的、polstra 等人的申请“系统自动诊断的内核的内核测试接口和方法”中所公开的(特此引入作为对比文件)，提供了一种高度自动化的测试和诊断的系统和方法，这种系统和方法可在地址总线功能测试和诊断之前仅仅验证数据总线。这种验证提供了对数据总线完整功能性的较少确定性，因为，这种验证对数据总线的运用和测试只达到确定足以执行地址总线测试和诊断所必须的程度。具体地说，只需一小部分数据总线线路正常运行。对数据总线证而不是测试的原因，如共同未决的polstra 等人申请中所公开的，是为了减少执行 UUT 内核全部诊断所需的时间。单个元件最慢的测试是需要由操作者手工探测的对测试系统进行数据总线测试和诊断的较高位数据总线的测试。在用存储器仿真测试期间，由仿真存储器从地址上收集能反映内核状态的信号。所以，除非数据总线功能正常，这些信号是否发现数据总线或地址总线上有故障是不明确的。地址总线的测试无需探测并几乎不化时间；如果没发现故障，则可发现数据总线和地址总线都是完全起作用的。因此，最好在地址总线测试之后并仅当在内核某点上已发现故障时再进行数据总线测试。此外，还要求能取消探测，因为探测费时而且要求对测试系统部分较为熟练的操作者。

因此，本发明的目的是为基于微处理器的系统提供测试设备，

用以进行数据总线验证而无需探测数据总线的线路。

本发明的另一目的是提供一种用于验证基于微处理器系统全部数据总线而无需探测的测试方法。

本发明的又一目的是在用于测试基于微处理器系统的设备中，提供一种测试激励例程，该例程起数据总线的合格/不合格测试的作用，从而如果报告数据总线无故障的话，则允许在数据总线诊断之前执行地址总线测试和诊断。

本发明的还有一目的是在用于测试基于微处理器系统的设备中提供一种验证例程，如果在验证例程处理中未发现故障的话，则该例程取消对数据总线诊断的要求。

本发明旨在利用存储器仿真对基于微处理器系统内核的数据总线进行测试和验证的设备，包括数据总线的验证方法。首先，执行装入操作，将包含许多位（其位数等于数据总线线路数的一半）的第一位模式加到数据总线的第一部分的线路上。其次，执行装入操作，将与所述第一位模式互补的第二位模式加到由数据总线线路的剩余部分构成的数据总线的第二部分的线路上。随后，比较第一和第二位模式，如果第二位模式不与第一位模式互补则产生一信号。这些步骤最好重复多次。然后，用类似方法，执行装入操作，将包含等于数据总线线路数一半的位数的第三位模式加到数据总线线路的第三部分上，对不与第三位模式互补的第四模式执行装入操作加到由剩余数据总线线路构成的第四部分总线线路上。然后对所述第三和第四位模式进行比较，如果第四位模式与第三位模式不互补，便产生一信号。这些步骤出最好重复多次。根据这些比较的结果便可验证数据总线。

显然，第一和第二模式能同时装入，也许是单个操作，这样做

方便，而且第三和第四模式也可如此。但是，如果第二和第四模式分别由第一和第三模式产生而不是由存贮的的产生，那么，这样做可能是不方便的。

在用于基于微处理器系统的测试的设备中，通过执行该例程，可全部验证数据总线，并且，由于如果报告无故障的话，该验证例程就排除了探测数据总线线路的必要性，所以，加快了整个测试的进程。

本发明的上述及其他目的通过参考附图和以下关于本发明的详细描述，对本领域的技术人员而言是显而易见的。

图1是与本发明结合的基于微处理器测试设备的说明图，

图2是图1所示系统的简化方框图，

图3是图2所示整个系统的详细方框图，

图4是说明本发明功能的流程图。

参考图1 本发明的概略，与被测部件(UUT)14连接的测试设备包括：主机处理器10（该主机安置于小型外壳内，并包括：键盘20、探针32、显示器 22）、接口箱12、包括同步模块适配器150的同步模块150以及随 UUT14的存储器配置而定的至少 1个存储器模块100（图中示出2个）。存储器模块通过多导线电缆92和与UUT存储器插座72对应的插头与UUT相连接。图2原理上说明图1所示系统的互连，展示了该设备在多个外壳中的最佳安排。可以理解，所示最佳的系统元件的连接为操作者方便起见可安装比所示较多或较少的元件。例如，该接口箱可全部包括在主机的同一外壳内。也可注意到，图 2中，虽然以存储器模块在电气上取代 UUT存储器，要么是物理取代要么是并行连接而禁止 UUT存储器，但是，所述同步模块与留在UUT电路中的 μP 相连接。

本发明的验证例程包含激励例程，该激励例程包括这些步骤：将一个其宽度为待验证数据总线或部分数据总线一半的位模式送至所述数据总线的低位线路或其中部分线路；将同一位模式或与该位模式互补的位模式传给该数据总线或其一部分的高位线路上；对各高位和低位线路上所接收的数据模式进行评价，确定这些位模式是否互补。由于这些模式为原模式或与之互补的位模式；评价的不同结果已足以验证数据总线的全部宽度。

根据以上本发明的简短概述，下面将把上述的在与本发明共同未决的申请中所描述的系统的操作作为了解构成本发明的增强措施的操作的背景来回顾。

测试系统包括许多新颖的过程，这些过程包括总线测试原语，数据激励原语和地址激励原语，这些原语将在下文分别归纳。每条这种原语具有用来测试基于 μP 系统的内核的特定部分的实用性，并且，当按照本发明以序列的形式使用时，允许较高级别的自动测试和诊断，这些测试和诊断具有较高速度，并且，和迄今为止能利用的测试和诊断相比，对操作者更为便利。

总线测试原语由测试设备主机中的程序执行。总线测试原语的主要功能是确定 μP 能否在内核执行基本读和写操作，并能由少至单一读与写组成。如果成功，便可知 μP 至少能访问存储器，这种情形下，仿真存储器通过数据总线接收一个位模式并将该位模式置于地址总线上，在该处，位模式可由接口接收并监测。然而，在最佳实施例中，由于测试设备通过产生特征标记来执行数据和地址总线的诊断，所以，总线测试原语以程序的形式实现，该程序预定利用对应于引导存储器的部分数据和地址总线。无论以单一操作实现还是以操作序列来实现，重要的是总线测试原语并不用尽包含数据

和地址的线路或甚至并不用尽可放置于这些线路或部分线路上的位组合，因此可快速地起合格/不合格测试的作用。

实现数据激励原语的 μP 可操作性层次比总线测试原语要低，尤其是通过重复地复位 μP 来进行的数据激励原语；该功能可在总线测试原语之前已被测试。一旦复位， μP 存取引导存储器的第一单元并检索所存贮的位模式。数据激励不是程序而是在每次复位时通过改变引导存储器中第一单元中的位模式来实现的。注意到该功能对所谓向量复位和执行复位两种类型的微处理器是公用的，这是很有意义的。无论用哪一类型的 μP ，从存储器检索到的位模式将通过数据总线进行通信，并且，出现于地址总线上。在复位期间，在初始读取引导存储器第一单元期间由 μP 产生同步脉冲，该脉冲由同步模块捕获，与接口箱通信并用于评价数据总线上所出现的信号，通过探测或非探测的测试来收集数据总线线路的特征标记。后者由一个类似总线测试的过程完成，该过程使用位模式的完备的数据激励序列，同时监测芯片选择线路。数据激励序列以这种意义上是完备的，即，该序列由一连串基本为任意的模式所组成，尽管如此，选择该序列使得在数据总线的每一线路上产生唯一的特征标记。如在总线测试原语的描述中所指出的，存储器引导空间中情况一样，仅当一个或多个的高位在循环通过引导空间单元时如预期那样为非零时，芯片选择线路（地址总线高位若干位的逻辑函数）才会反应出错误，并且，即使1条或多条高位总线线路与地粘连，该合格/不合格测试也能通过。同样，在执行数据激励原语时，如预期那样芯片选择信号的有无反映高位线路中的一根有无被闭锁（例如，与地短接）。如果该测试通过，只剩下将数据的一条线路与另一数据总线线路粘连的粘连线故障。这可在以后用探测来诊断。但是，根据

本发明，最好在这时确认数据总线，因而，如果确认例程执行无故障，那么，就可知数据总线是完全起作用的。

数据总线测试和诊断之后，可以用与关于数据激励原语的相同的激励序列进行地址总线的测试。然而，这可通过执行编程的读／写指令序列来完成，这些指令序列使用这些位模式并通过探测、或最好是用分析存储器，从中演绎出闭锁或粘连的线路。如上所指出地址线的完备测试只要用激励序列中有限个数的位模式便可进行。一旦地址总线线路被这样完全诊断，由于任何在数据总线线路上出现的故障也会反映在地址总线线路上，所以，数据线路的完全诊断也是可能的。由于地址总线线路已完全诊断过，所以，所述任何故障都对具体总线隔离。指出以下事实是有益的，即，对执行复位处理器来就，执行地址激励原语的程序可以为单一指令。对于依赖引导单元地址而得到第一指令地址的向量复位处理器来说，地址激励原语通常不需要任何指令，仅仅将所要求的位模式置于仿真存储器的复位向量单元之中。

为了从整个系统和方法角度扼要说明以上归纳的这些原语，当要求执行对基于 μP 系统的内核测试时，将仿真存储器在电气上取代被测部件存储器，并将同步模块通过导线140（图3）与 μP 的时序信号和强制管脚相连接。当启动测试过程时，进行某些规定的检验来断定有无将电源加至如下文更详细枚举的内核元件。然后，执行复位过驱动检验，以确定接口箱事实上能否启动 μP 的复位，并由同步模块对该复位过驱动检验进行监测以确定该复位线是否先变有效后变为无效。 μP 的实际复位并不在这一步里检验，而只是在 μP 的合适管脚上呈现能够进行复位的信号。

下面，同步模块捕获待评价的 μP 时钟信号。如接口箱没有接

收到 μP 时钟信号，便对时钟作额外的检验以确定该时钟信号是否慢、或短接，并且，检验强制线上不期望的信号值。

这时，对 μP 本身未曾进行任何测试，不过已充分验证了信号而确定这些测试可能进行。这些测试中的基本测试，即， μP 复位，由复位线的过驱动和在与引导存储单元对应的芯片选择线上寻找来自地址译码器80的选片信号来完成。如 μP 复位成功，再次对 μP 复位以检验地址总线低位线路上的正确信号以存取引导存储器的第一单元。该过程验证了内核足够的功能性，可进行如上概述的总线测试序列、数据和地址激励原语序列。还应注意到，如果至此实施的任何测试已指明故障，便无需使用 μP 复位以外的任何更复杂或高级的功能，便能清楚地指出特定的内核故障。上面概述的总线测试在其第一循环（可能仅有的）里只证明对内核其余部分的完全测试和诊断所必需的唯一的另一种功能是：读和写操作。还应记住，由于同步模块产生的高分辨的同步脉冲的缘故，可将所考虑的总线周期隔并，并以响应用来运用这些线路的激励模式而产生的特征标记为依据进行总线的评价。例如，如在上文引入的Polstra等人的申请中所公开的那样，由于只有12个位模式是激励原语所必需的，以便运用总线中32条线路，产生每一线路的唯一特征标记、用于对总线的完全诊断，因此，可使测试速度显著提高。

具体参照图 4。数据总线证实例程包含从仿真存储器读取位模式以及将该位模式置于数据总线的一部分(401)。然后，最好由 μP 产生所述位模式之反(402)并置于数据总线另一部分(403)。然后，对数据总线的两个部分进行比较(404)，如果两个位模式不是互补，则产生一种信号。该结果可立即报告，但最好收集一串位模式的预定序列上重复该序列不必在单个操作序列中执行，但可中断或与步

骤406—409的类似序列交叉进行。

数据总线的证实进程可通过将最好来自位模式第二序列的位模式置于数据总线的一部分(406)来继续。在该序列中,明显不同于前面的序列,将同一位模式置于数据总线的第二部分(407)。再作比较(408)并收集比较结果(411)。然后,最好重复在位模式的整个第二序列上这些步骤(409)。

若两个序列已完成,评价所收集的比较结果(410)和(411),如果第一序列报告全部互补而第二序列报告没有互补,则证实了正考虑的数据总线。

按照本发明并再参考图3,导线140连接复位线,并且,与其上面出现反映 UUT μ P的运行状态的信号的其他线路相连接。例如,对80386处理器,这些线路为 HOLD、HLDA、(保持应答)、 $\overline{ADS}$ 、 $\overline{READY}$ 、CLK²和RESET线。这些信号通过线路140由缓冲器152接收,并作为总线周期状态机200的输入通过电缆 90传送到接口箱。总线周期状态机200在主机 10的控制下对这些信号执行逻辑操作以产生用以控制同步脉冲发生状态机 202的控制信号,所述同步脉冲发生状态机响应到该处的信号而产生同步信号,该同步信号送到从中产生控制信号的主机10,而这些控制信号返回到如图示的接口箱内各部件中,特别的是分析器RAM 62。

用于分析地址跟踪的机构和用于分析来自总线测试原语的跟踪的机构是同一个。该机构的更全面的描述公开于 J. Polstra 的并已转让给本发明的受让人的“基于存储器存取分析的内核电路自动验证”,特此引入作为对比文件。

按照本发明的最佳实施例,有10个位模式用于两个序列中的每一个。一个序列可将每一个位模式及其互补位模式加到数据总线的

整个宽度上。这些模式(16进制记法)最好为:

\$FFFF0000	\$00FFFF00
\$0F0FF0F0	\$3333CCCC
\$5555AAAA	\$AAAA5555
\$CCCC3333	\$F0F00F0F
\$FF0000FF	\$0000FFFF

其它序列将每个位模式以及该位模式的真正的重复以同样方式加到数据总线上。这些模式最好为:

\$FFFFFFFF	\$00FF00FF
\$0F0F0F0F	\$33333333
\$55555555	\$AAAAAAAA
\$CCCCCCCC	\$F0F0F0F0
\$FF00FF00	\$00000000

选择这些模式要保证至少可用一个模式检测出所有数据线故障。可检测出的故障是,例如,线路固定为高或低、线路与其它线路粘连。只有当报告一个模式序列的所有模式为高位和低位互补的位模式以及报告另一序列没有高位与低位互补的位模式时,才证实数据总线,所以也保证不会有数据总线故障被不正确地报告为合格。

这个证实过程可等价地应用于4位、8位、16位、32位或任何数据总线宽度的处理器。

下文是80386数据测试程序的说明。

作为按照本发明的证实方法和数据结构的应用的一个实例,下面的例子将详细地将本发明应用于包括80386型处理器的内核的测试。应记住,这种方法和数据结构可应用于许多不同类型的计算机和数字电路。

体现本发明的方法和数据结构的程序如下：

[illegible]

在最左栏上面，列表示出了在该程序中每一步期间32条数据总线上出现的二值电平（好的 UUT）。如所示，列表表示了实际使用80386所有32条数据线的一个UUT。然而，如上所述，该测试对使用较少数据线的UUT也是有效的。

“地址”栏以十六进制给出了在该程序每一步中由UT微处理器所存取地址。每次存取取出 32 位=4 字节的数据线上的数据，这

也是以4字节为间隔分隔地址的原因。

“标号”栏给出了当前指令可选的符号。这是汇编语言程序设计中通常做法。

“操作码”栏给出80386机器指令的符号名称。

“操作数”栏列出了可更完全地说明由微处理器所作的精确动作的操作数（参数）。

每条取出指令通过数据总线传送完整的32位信息。这样的一次传送可能包含一条或多条完全或部分 80386指令。例如，前两条指令“nop”和“movax,xxxxxxxxxxxxxxxx”一起占取数据的前32位。

一条指令也可分为两次32位传送。这可从“Xor ax yyyyyyyyyy yyyyyyyyyy”指令中看出，该指令始于一个 32位集合，但终止于另一个集合。

该程序的逐步的描述如下：

1. nop

这是 80386的空操作指令。微处理器什么也不做，它只用来占用空间使得下面的指令落在数据总线所要求部分。

2. movax,xxxxxxxxxxxxxxxx

这是“装入立即数”指令。在这里，“xxxxxxxxxxxxxxxx”表示由该接口箱所填测试模式的较高的16位。该指令的作用是将这16位传送到处理器的“ax”寄存器中。注意：模式“xxxxxxxxxxxxxxxx”是通过数据线 D16~D31传送的。这是专门由前面的“nop”指令实现的。

3. not ax

这是“位取反”指令。它将“ax”寄存器中每一位取反。这条指令执行之后，“ax”寄存器包含测试模式“xxxxxxxxxxxxxxxx”

的反。

4. nop

这是另一条“空操作”指令，包含把下一指令置于数据总线上所要求部分。

5. xorax, yyyyyyyyyyyyyyyyyy

这是一条“异或立即数”指令。其中，“yyyyyyyyyyyyyyyyyy”表示由接口箱所填入的测试模式的较低的16位。这条指令的作用是将位“yyyyyyyyyyyyyyyyyy”以一定方式与“ax”寄存器的当前内容进行组合。即，“ax”中那些与模式相对应位相等位得到新值0，而那些不等的位得到新值1。执行这条指令之后，当，并且仅当先前含有值“yyyyyyyyyyyyyyyyyy”时，“ax”含有值0000000000000000。

在该指令之前，“ax”包含模式“xxxxxxxxxxxxxxxx”的反。所以，如果“yyyyyyyyyyyyyyyyyy”恰好为“xxxxxxxxxxxxxxxx”的反，那么，该指令可得到结果0。否则，得出某非零值。

注意：模式“yyyyyyyyyyyyyyyyyy”是通过数据线D00至D15进行传送的，数据线的另一半可用来传送步骤2中的其它模式。这是专门由前面的“nop”指令完成的。

6. jzsuccess

这是一条“若零转移”指令。如前面指令结果为0（即，如数据测试通过），则该指令将转移到标号“success”的地址。否则（即，如果数据测试失败），顺序执行下面的指令。

注意：该指令的标号为“success”。这样，如测试通过，处理器会无限期地循环，执行这条“若零转移”指令。

7. jmpfailure

这是一条“无条件转移”指令。它使得继续执行标号“failure”的指令。注意到标号“failure”的指令位于地址 FFFFF800，远离“success”地址（它处于 FFFFFFF8和FFFFFFFC之间）。这是专门用来由接口箱通过考察所引用地址踪迹来判定测试是通过还是未通过。

8.failure:impfailure

这条指令使得处理器执行在 failure（失败）地址上的无限循环，即FFFFF800。

当该程序运行时，接口箱使用地址 RAM(ARAM)聚集 UUT地址引用的踪迹。然后便分析地址的序列。如果有一连串顺序存取继之以转移回“success”地址，那么，接口箱便知测试通过。如果继之以别的（通常为到“failure”地址的转移），那么，接口箱便知测试未通过。

这样全面地描述了本发明的细节后，应该理解。在不偏离本发明精神和范围的情况下许多变型和修改对本领域的技术人员是显而易见的。上面给出的详细描述用作一种实例而不是限制，本发明的范围仅由所附的权利要求所限定。

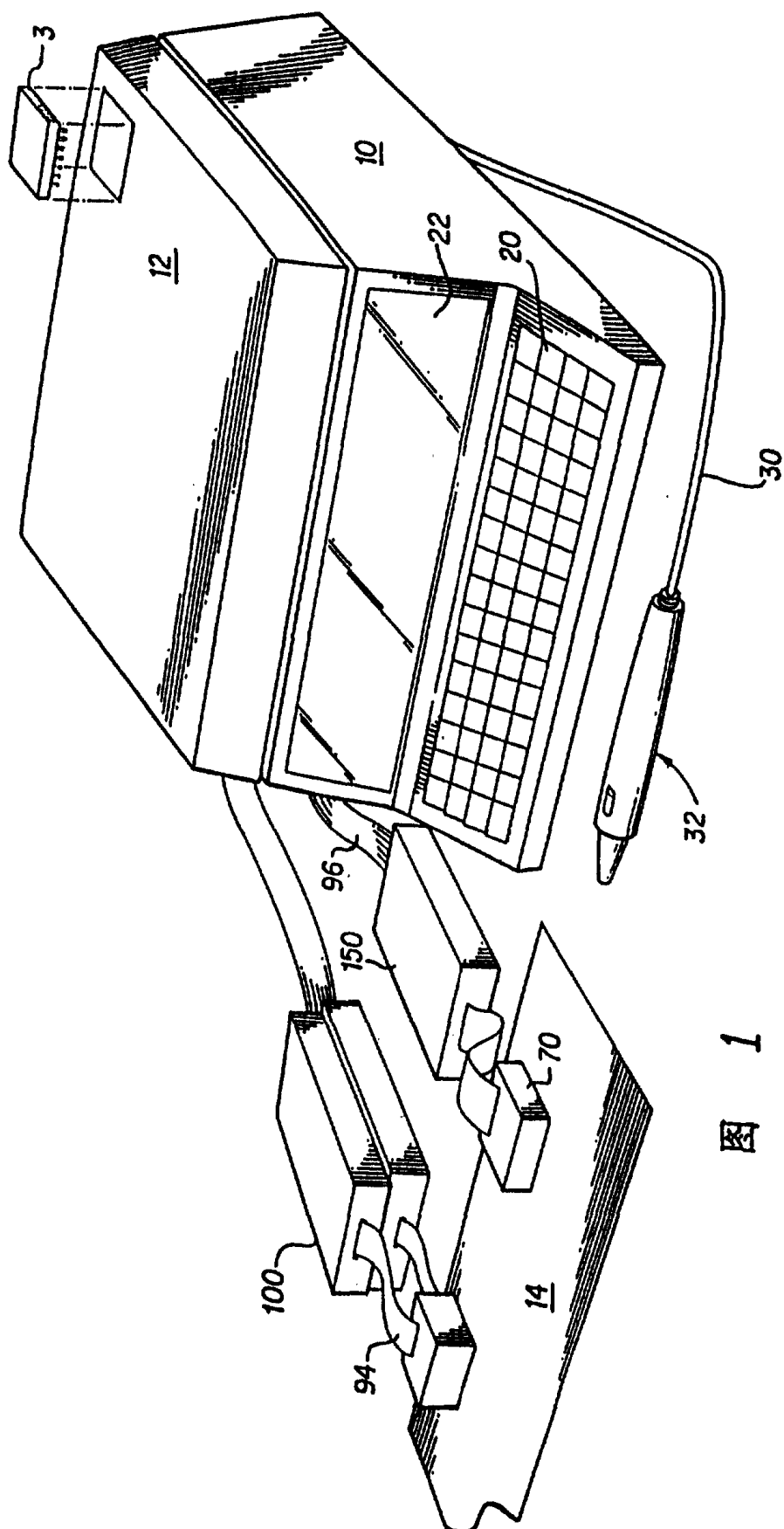


图 1

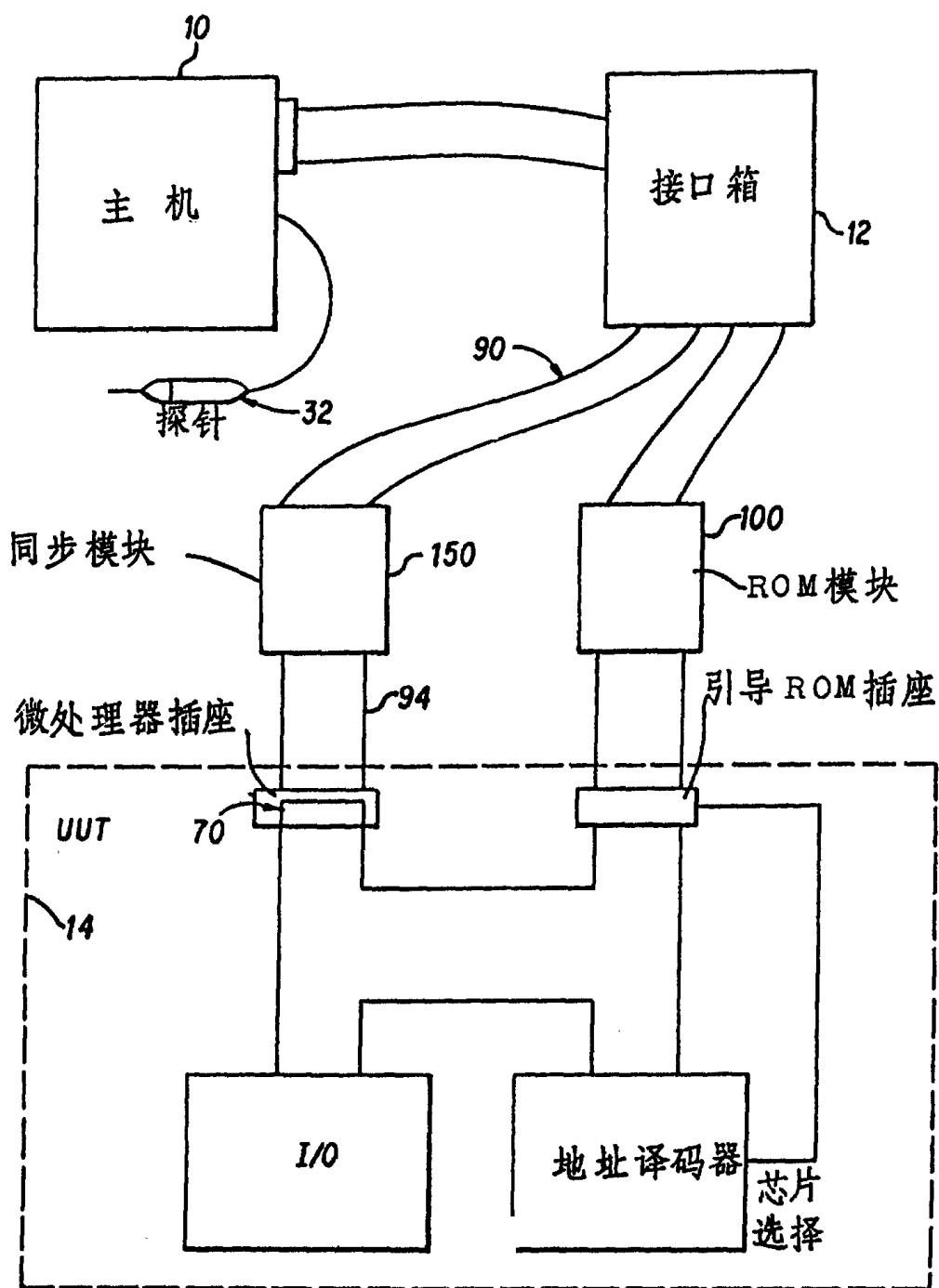
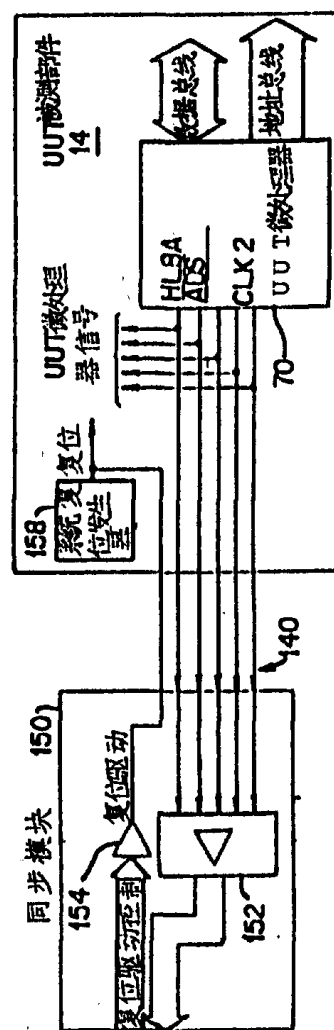
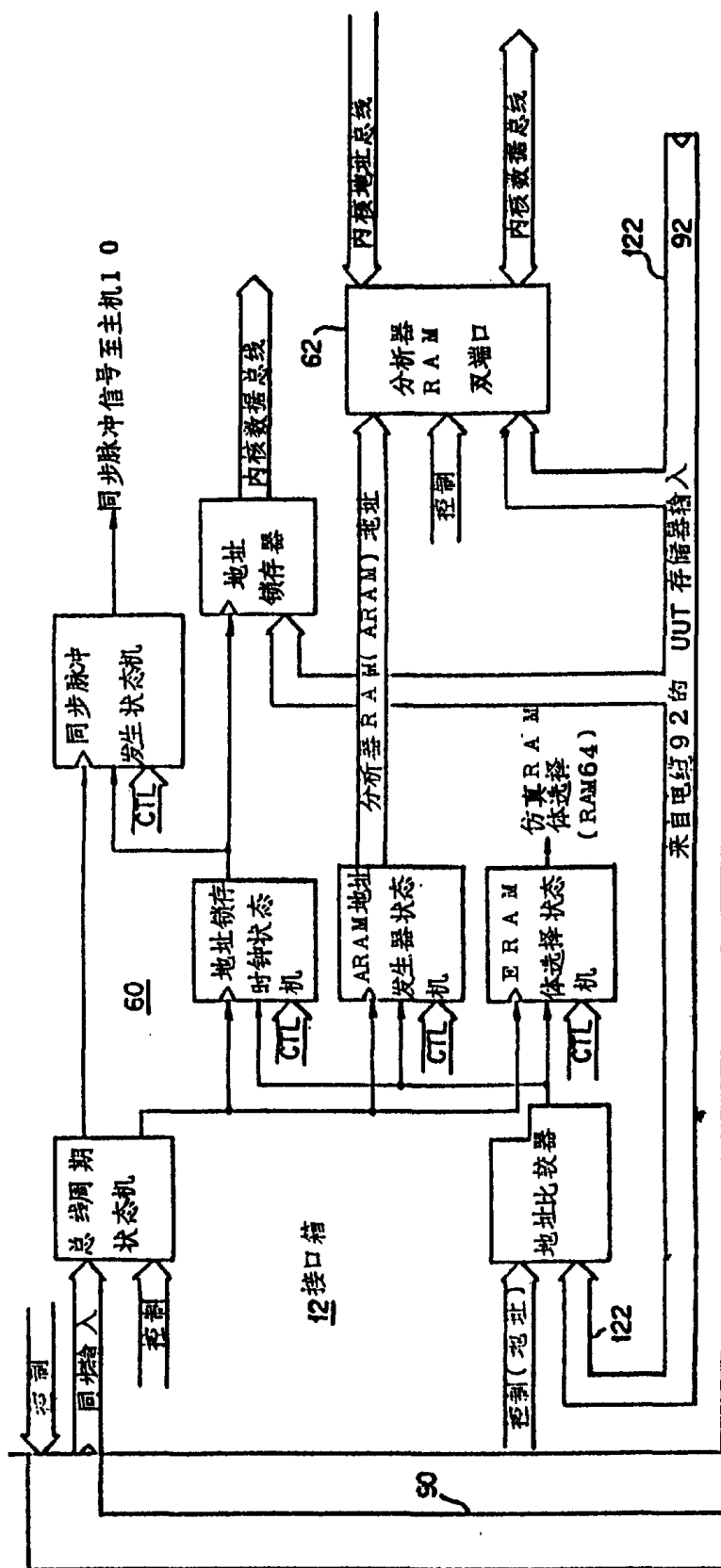


图 2



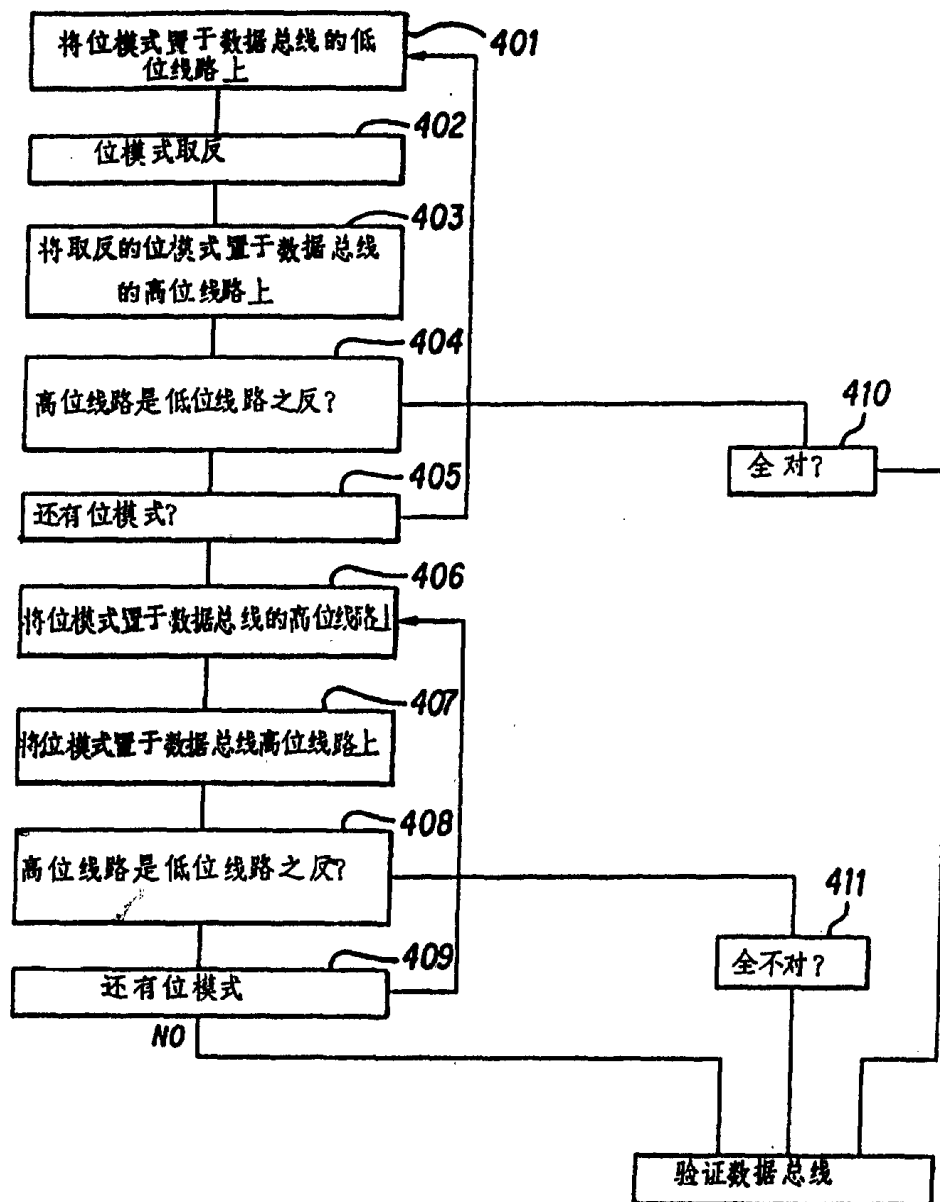


图 4