证书号第8117191号

专利公告信息

发明专利证书

发明名称：一种硬件支持的观察点调试方法

专利权人：无锡先进技术研究院

地　　址：214122 江苏省无锡市滨湖区绣溪路50号KPARK商务中心2号楼

发 明 人：谢汶兵;刘汉旭;张艺鸣;关睿雪;李佳梅

专　利　号：ZL 2023 1 0127399.9　　授权公告号：CN 116069653 B

专利申请日：2023年02月17日　　授权公告日：2025年07月29日

申请日时申请人：无锡先进技术研究院

申请日时发明人：谢汶兵;刘汉旭;张艺鸣;关睿雪;李佳梅

国家知识产权局依照中华人民共和国专利法进行审查，决定授予专利权，并予以公告。

专利权自授权公告之日起生效。专利权有效性及专利权人变更等法律信息以专利登记簿记载为准。

局长
申长雨

国家知识产权局
2025年07月29日

国 家 知 识 产 权 局

210061

[redacted]

发文日：

2025年07月01日

申请号：202310127399.9　　**发文序号：**2025070100498740

申请人：无锡先进技术研究院

发明创造名称：一种硬件支持的观察点调试方法

授 予 发 明 专 利 权 通 知 书

1.根据专利法第 39 条及实施细则第 60 条的规定，上述发明专利申请经实质审查，没有发现驳回理由，现作出授予专利权的通知。

申请人收到本通知书后，还应当依照办理登记手续通知书的内容办理登记手续。

申请人按期办理登记手续后，国家知识产权局将作出授予专利权的决定，颁发发明专利证书，并予以登记和公告。

期满未办理登记手续的，视为放弃取得专利权的权利。

法律、行政法规规定相应技术的实施应当办理批准、登记等手续的，应依照其规定办理。

2.授予专利权的上述发明专利申请是以下列申请文件为基础的：

☒原始申请文件。☐分案申请递交日提交的文件。☐下列申请文件：

3.授予专利权的上述发明专利申请的名称：

☒未变更。

☐由_____变更为上述名称。

4. ☐申请人于_____年_____月_____日提交专利号为_____的“放弃专利权声明”，经审查：

☐进入放弃专利权的程序。

☐未进入放弃专利权的程序。理由是：申请人声明放弃的专利与本发明专利申请不属于相同的发明创造。

5. ☐审查员依职权对申请文件修改如下：

6. ☐申请人在申请日后补交了实验数据，该数据未包含在授权公告文本中。

注：在本通知书发出后收到的申请人主动修改的申请文件，不予考虑。

国家知识产权局 专利审查业务章 1108135934

审查员：彭凤鸣　　审查部门：专利审查协作四川中心

联系电话：028-62967882

纸件申请，回函请寄：100088 北京市海淀区蓟门桥西土城路 6 号 国家知识产权局专利局受理处收

电子申请，应当通过电子专利申请系统以电子文件形式提交相关文件。除另有规定外，以纸件等其他形式提交的文件视为未提交。

(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 116069653 A
(43)申请公布日 2023.05.05

(21)申请号 202310127399.9

(22)申请日 2023.02.17

(71)申请人 无锡先进技术研究院
地址 214122 江苏省无锡市滨湖区绣溪路50号KPARK商务中心2号楼

(72)发明人 谢汶兵 刘汉旭 张艺鸣 关睿雪 李佳梅

(74)专利代理机构 南京纵横知识产权代理有限公司 32224
专利代理师 董建林

(51)Int.Cl.
G06F 11/36(2006.01)

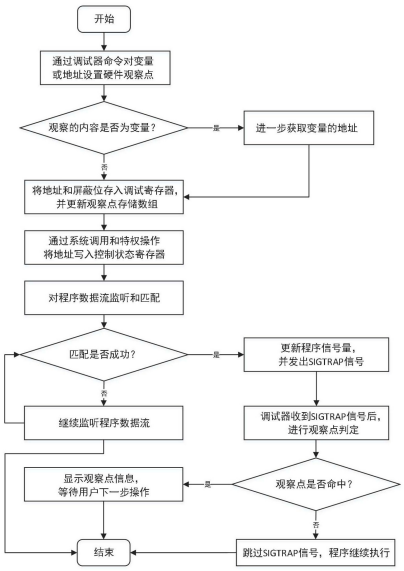
权利要求书1页 说明书6页 附图2页

(54)发明名称

一种硬件支持的观察点调试方法

(57)摘要

本发明公开了一种硬件支持的观察点调试方法,包括步骤1:在处理器中设置一组控制状态寄存器用于数据流调试;步骤2:在调试器中添加硬件观察点功能接口进行观察;步骤3:将观察内容映射到处理器的控制状态寄存器,开始对程序数据流进行监听和匹配;步骤4:当处理器监控到控制状态寄存器的内容被修改后,更新程序的信号量,并发出SIGTRAP信号给程序;步骤5:调试器获取SIGTRAP信号,并进行观察点判定,若判定成功,程序停止运行,等待用户下一步操作,本发明能够及时准确地发现由于数据被非法修改而导致的程序错误,提高程序调试效率。



CN 116069653 A

1.一种硬件支持的观察点调试方法,其特征在于,包括:

步骤1:在处理器中设置一组控制状态寄存器用于数据流调试;

步骤2:在调试器中添加硬件观察点功能接口进行观察;

步骤3:将观察内容映射到处理器的控制状态寄存器,开始对程序数据流进行监听和匹配;

步骤4:当处理器监控到控制状态寄存器的内容被修改后,更新程序的信号量,并发出SIGTRAP信号给程序;

步骤5:调试器获取SIGTRAP信号,并进行观察点判定,若判定成功,程序停止运行,等待用户下一步操作。

2.根据权利要求1所述的硬件支持的观察点调试方法,其特征在于,所述步骤1中,控制状态寄存器包括数据流地址匹配寄存器和数据流地址屏蔽寄存器。

3.根据权利要求2所述的硬件支持的观察点调试方法,其特征在于,所述步骤1中,对控制状态寄存器的设置,包括将数据流地址匹配寄存器设置为可读写,复位时清零,当数据流访问地址与该寄存器内容匹配时产生数据流故障;将数据流地址屏蔽寄存器设置为可读写,复位时将低53位全部置为1,屏蔽进行数据流地址比较的位数,可选择地址匹配的长度。

4.根据权利要求1所述的硬件支持的观察点调试方法,其特征在于,所述步骤2中,调试器中添加的观察点功能接口包括观察点存储数组定义接口、观察点设置接口、观察点删除接口、观察点启用接口。

5.根据权利要求4所述的硬件支持的观察点调试方法,其特征在于,所述观察点存储数组定义接口包括观察点地址长度判断、观察点索引判断、观察点存储数组更新,所述观察点启用接口通过观察点生效标志来判断当前的观察点状态;所述观察点删除接口通过往控制状态寄存器写入默认值来取消数据流匹配功能。

6.根据权利要求1所述的硬件支持的观察点调试方法,其特征在于,所述步骤2中,还包括在调试器中定义一组调试寄存器,所述调试寄存器与处理器的控制状态寄存器相互对应。

7.根据权利要求6所述的硬件支持的观察点调试方法,其特征在于,所述步骤3中,将观察内容映射到处理器的控制状态寄存器,开始对程序数据流进行监听和匹配,包括:

在程序调试阶段,当对某个变量或地址设置观察点后,调试器将观察内容写入调试寄存器,再利用系统调用将观察内容映射到处理器的控制状态寄存器,开始对程序数据流进行监听和匹配。

8.根据权利要求6所述的硬件支持的观察点调试方法,其特征在于,所述步骤5中,若观察点判定失败,则跳过当前SIGTRAP信号。

一种硬件支持的观察点调试方法

技术领域

[0001] 本发明涉及一种硬件支持的观察点调试方法,属于计算机程序的调试技术领域。

背景技术

[0002] 程序调试是软件开发过程中必不可少的阶段,在该阶段中程序员需要耗费大量精力和时间并借助各种调试方法查找程序BUG并进行修正。近年来伴随着应用软件的规模和复杂度提升,导致程序中的BUG变得愈发隐蔽和难以定位,从而对程序调试技术的依赖程度越来越高,迫切需要实现一些高效便捷的程序数据流调试方法。

[0003] 观察点(watchpoint)是程序调试中关键技术之一,当对程序中变量设置观察点后,任何试图对该变量的操作都会触发程序停止运行,从而暴露出程序中的错误,比如对程序变量的异常访问和非法篡改。观察点的实现方法通常有两种:软件观察点和硬件观察点。软件观察点通过单步调试让程序停下来并同时检测变量的值,该方法虽然实现过程简单,但“一步一停”会严重降低程序的执行速度,使得调试过程低效且耗时;硬件观察点则借助处理器硬件寄存器对变量进行监控,当匹配到程序数据流试图修改变量值时,就会触发硬件中断并暂停程序运行,该方法在实现监控机制的同时不影响程序执行效率,但硬件观察点的设计和实现与处理器架构紧密相关,需要处理器和操作系统功能同时支持。

[0004] 目前,观察点的实现方法有两种:软件观察点和硬件观察点。软件观察点实现过程简单,但会严重降低程序的执行速度,使得调试过程变得耗时;硬件观察点则借助处理器硬件寄存器对变量进行监控,能够在实现监控机制的同时不影响程序执行效率,但硬件观察点的设计和实现与处理器架构紧密相关,需要处理器和操作系统功能同时支持,整体实现难度和工作量较大。

发明内容

[0005] 本发明的目的在于克服现有技术中的不足,提供一种硬件支持的观察点调试方法,能够及时准确地发现由于数据被非法修改而导致的程序错误,提高程序调试效率。

[0006] 为达到上述目的,本发明是采用下述技术方案实现的:

[0007] 第一方面,本发明提供了一种硬件支持的观察点调试方法,包括:

[0008] 步骤1:在处理器中设置一组控制状态寄存器用于数据流调试;

[0009] 步骤2:在调试器中添加硬件观察点功能接口进行观察;

[0010] 步骤3:将观察内容映射到处理器的控制状态寄存器,开始对程序数据流进行监听和匹配;

[0011] 步骤4:当处理器监控到控制状态寄存器的内容被修改后,更新程序的信号量,并发出SIGTRAP信号给程序;

[0012] 步骤5:调试器获取SIGTRAP信号,并进行观察点判定,若判定成功,程序停止运行,等待用户下一步操作。

[0013] 进一步的,所述步骤1中,控制状态寄存器包括数据流地址匹配寄存器和数据流地

址屏蔽寄存器。

[0014] 进一步的,所述步骤1中,对控制状态寄存器的设置,包括将数据流地址匹配寄存器设置为可读写,复位时清零,当数据流访问地址与该寄存器内容匹配时产生数据流故障;将数据流地址屏蔽寄存器设置为可读写,复位时将低53位全部置为1,屏蔽进行数据流地址比较的位数,可选择地址匹配的长度。

[0015] 进一步的,所述步骤2中,调试器中添加的观察点功能接口包括观察点存储数组定义接口、观察点设置接口、观察点删除接口、观察点启用接口。

[0016] 进一步的,所述观察点存储数组定义接口包括观察点地址长度判断、观察点索引判断、观察点存储数组更新,所述观察点启用接口通过观察点生效标志来判断当前的观察点状态;所述观察点删除接口通过往控制状态寄存器写入默认值来取消数据流匹配功能。

[0017] 进一步的,所述步骤2中,还包括在调试器中定义一组调试寄存器,所述调试寄存器与处理器的控制状态寄存器相互对应。

[0018] 进一步的,所述步骤3中,将观察内容映射到处理器的控制状态寄存器,开始对程序数据流进行监听和匹配,包括:

[0019] 在程序调试阶段,当对某个变量或地址设置观察点后,调试器将观察内容写入调试寄存器,再利用系统调用将观察内容映射到处理器的控制状态寄存器,开始对程序数据流进行监听和匹配。

[0020] 进一步的,所述步骤5中,若观察点判定失败,则跳过当前SIGTRAP信号。

[0021] 与现有技术相比,本发明所达到的有益效果:

[0022] 本发明在程序调试阶段,当对某个变量或地址启动观察调试后,通过系统调用和特权操作将观察的地址写入控制状态寄存器,开始对程序数据流进行监控和匹配,一旦发现地址被修改,更新程序的信号量,并发出SIGTRAP信号,调试器获取到信号后进行观察点判定,程序停止运行,等待用户下一步操作。通过这种基于硬件“实时”的监控机制,能够及时准确地发现由于数据被非法修改而导致的程序错误,提高程序调试效率。

附图说明

[0023] 图1是本发明实施例提供的一种硬件支持的观察点调试方法流程图;

[0024] 图2是本发明实施例提供的一种观察点存储数组结构示意图;

[0025] 图3是发明实施例提供的一种数据流地址匹配寄存器位功能图;

[0026] 图4是发明实施例提供的一种数据流地址屏蔽寄存器位功能图。

具体实施方式

[0027] 下面结合附图对本发明作进一步描述。以下实施例仅用于更加清楚地说明本发明的技术方案,而不能以此来限制本发明的保护范围。

[0028] 实施例1

[0029] 本实施例介绍一种硬件支持的观察点调试方法,包括:

[0030] 步骤1:在处理器中设置一组控制状态寄存器用于数据流调试;

[0031] 步骤2:在调试器中添加硬件观察点功能接口进行观察;

[0032] 步骤3:将观察内容映射到处理器的控制状态寄存器,开始对程序数据流进行监听

和匹配；

[0033] 步骤4：当处理器监控到控制状态寄存器的内容被修改后，更新程序的信号量，并发出SIGTRAP信号给程序；

[0034] 步骤5：调试器获取SIGTRAP信号，并进行观察点判定，若判定成功，程序停止运行，等待用户下一步操作。

[0035] 本实施例提供的硬件支持的观察点调试方法，其应用过程具体涉及如下步骤：

[0036] a.在处理器中设置一组控制状态寄存器，具体需包含数据流地址匹配寄存器和数据流地址屏蔽寄存器；

[0037] b.在调试器软件代码支持观察点包含观察点存储数组定义、观察点设置、观察点删除、观察点启用和调试寄存器定义等；

[0038] c.通过调试器将观察的内容写入调试寄存器，再利用系统调用和特权操作将观察内容映射到处理器的控制状态寄存器，开始对程序数据流进行监听和匹配；

[0039] d.当处理器监控到控制状态寄存器的内容被修改后，更新程序的信号量，并发出SIGTRAP信号，调试器获取到信号后进行观察点判定，程序停止运行，等待用户下一步操作。

[0040] 具体的，所述步骤a中对控制状态寄存器的设置，包括将数据流地址匹配寄存器设置为可读写，复位时清零，当数据流访问地址与该寄存器内容匹配时产生数据流故障；将数据流地址屏蔽寄存器设置为可读写，复位时将低53位全部置为1，屏蔽进行数据流地址比较的位数，可以选择地址匹配的长度。

[0041] 具体的，所述步骤b中在调试器源码添加的观察点功能接口，其中观察点存储数组包含观察点设置接口包含观察点地址长度判断、观察点索引判断、观察点存储数组更新；观察点启用接口通过观察点生效标志来判断当前的观察点状态；观察点删除接口通过往控制状态寄存器写入默认值来取消数据流匹配功能。

[0042] 具体的，所述步骤c中调试器将观察的地址写入控制状态寄存器，具体借助系统调用和特权操作途径实现。首先将地址写入调试寄存器，再进一步写入到控制状态寄存器，最后将观察的地址成功记录到线程状态中，线程在运行时会持续匹配控制状态寄存器的观察内容，开始对程序数据流进行监听。

[0043] 具体的，所述步骤d中当匹配到观察的内容被修改后，操作系统会更新程序的信号量，并发出SIGTRAP信号给程序，调试器中信号处理函数获取到SIGTRAP信号后，通过查询观察点存储数组进行观察点判定；若观察点匹配成功，则控制程序停止运行，等待用户下一步操作；若匹配失败则忽略该SIGTRAP信号。

[0044] 下面结合一个优选实施例，对上述实施例中涉及到的内容进行说明。

[0045] S10：在处理器中设置一组控制状态寄存器用于数据流调试。

[0046] 一组控制状态寄存器具体需包含数据流地址匹配寄存器和数据流地址屏蔽寄存器。

[0047] 该实施例中，对控制状态寄存器的设置，包括将数据流地址匹配寄存器设置为可读写，复位时清零，当数据流访问地址与该寄存器内容匹配时产生数据流故障；将数据流地址屏蔽寄存器设置为可读写，复位时将低53位全部置为1，屏蔽进行数据流地址比较的位数，可以选择地址匹配的长度。

[0048] 例如，如图3所示，依次设定了数据流地址匹配寄存器第0到63位的功能。其中第0

到52位表示与数据流地址进行比较的地址，即观察点能够设置的最大地址长度保持在53位以内；第53到54位表示地址匹配使能位，“00”指示禁止地址比较，“01”指示允许读访问地址比较，“10”指示允许写地址比较，“11”指示允许读写地址比较；第55位表示物理地址标志，为“1”表示物理地址比较，即数据流的物理地址0到47位进行比较。否则进行虚地址比较，即数据流的虚地址0到52位进行比较；第56到63位功能暂时保留。

[0049] 例如，如图4所示，依次设定了数据流地址屏蔽寄存器第0到63位的功能。其中第0到52位表示数据流地址匹配屏蔽位，与数据流地址匹配寄存器中第0到52位中的每一位对应，为“1”表示该位参与比较，初始全为“1”，即将长度为53位地址全部进行比较。

[0050] S20：在调试器中添加硬件观察点功能接口。

[0051] 调试器中添加的观察点功能接口包括观察点存储数组定义、观察点设置、观察点删除、观察点启用等。此外，在调试器中定义一组调试寄存器与处理器的控制状态寄存器相互对应。

[0052] 该实施例中，如图2所示，定义观察点的存储数组结构，数组的大小根据处理器支持的观察点最大数目来决定，数组中每个元素记录一个观察点的所有信息，包含观察地址、生效标志和其他信息；其中，观察地址记录观察的内容，如果观察的内容是变量，则记录变量的地址；生效标志用来记录观察点的禁用/启用状态；其他信息记录着观察点的读/写模式、观察地址的长度等信息。

[0053] 本发明实施例中，关于调试器中添加的硬件观察点功能接口和作用，描述如下表1所示：

[0054] 表1观察点功能接口和作用

[0055]

观察点功能接口	作用
uint32_t NumSupportedHardwareWatchpoints()	获取观察点的最大支持数目
uint32_t SetHardwareWatchpoint()	设置观察点
bool ClearHardwareWatchpoint()	删除观察点
bool WatchpointIsEnabled()	获取观察点生效状态
Status ReadDebugRegisterValue();	从调试寄存器中读取地址
Status WriteDebugRegisterValue();	将地址写入调试寄存器
lldb::addr_t GetWatchpointAddress()	获取观察点具体地址
void NativeProcessLinux::MonitorSIGTRAP()	监控SIGTRAP信号并进行处

[0056]

	理
void NativeProcessLinux::MonitorWatchpoint()	观察点命中后进行处理
void NativeThreadLinux::SetStoppedByWatchpoint()	停止程序运行，并告知原因

[0057] 例如,在调试器中通过观察点命令对变量或地址设置硬件观察点,调试器判断观察的内容是变量还是地址。如果是变量,则需要获取该变量的地址。得到观察的地址后,调试器直接调用SetHardwareWatchpoint()开始设置硬件观察点,其中,需要结合地址的屏蔽位和长度来获取最终的观察地址,进一步调用WriteDebugRegisterValue()将地址写入调试寄存器。同时,调试器更新观察点存储数组,添加当前设置的硬件观察点信息。

[0058] S30:读写控制状态寄存器,开始对程序数据流进行监听和匹配。

[0059] 在程序调试阶段,当对某个变量或地址设置观察点后,调试器将观察地址写入调试寄存器,再利用系统调用和特权操作将观察内容映射到处理器的控制状态寄存器,开始对程序数据流进行监听和匹配。

[0060] S31:操作系统判断观察的地址是否监听和匹配成功。

[0061] 通过系统调用和特权操作将观察内容映射到处理器的控制状态寄存器,开始监听程序执行的数据流,并调用do_match()进行数据流监控和匹配。

[0062] 例如,如果监控到观察的地址被修改,即数据流匹配成功后,操作系统便立即更新程序的信号量siginfo中的si_signo、si_addr、si_code、si_value等信息,并向程序发出SIGTRAP信号。如果数据流匹配失败,则继续监听程序数据流。

[0063] S32:调试器判断观察点是否命中。

[0064] 当监控到观察的内容被修改后,操作系统更新程序的信号量,并发出SIGTRAP信号给程序,调试器获取到信号后进行观察点判定。若判定成功,则程序停止运行,等待用户下一步操作;若判定失败,则跳过当前SIGTRAP信号。

[0065] 例如,调试器通过MonitorSIGTRAP()函数来捕获程序的SIGTRAP信号并进行判断,如果调试器判断结果为命中观察点,则调用MonitorWatchpoint()对当前线程状态进行更新,并暂停程序运行。

[0066] 若调试器判定观察点命中,则调用GetDescription()函数来显示观察点的具体信息,包括观察点的新值和旧值、观察地址、观察变量等信息。若调试器判定观察点未命中,则会跳过当前SIGTRAP信号,即不会把该信号当做观察点来处理。

[0067] 本发明在程序调试阶段,当对某个变量或地址启动观察调试后,通过系统调用和特权操作将观察的地址写入控制状态寄存器,开始对程序数据流进行监控和匹配。一旦发现地址被修改,更新程序的信号量,并发出SIGTRAP信号,调试器获取到信号后进行观察点判定,程序停止运行,等待用户下一步操作。通过这种基于硬件“实时”的监控机制,能够及时准确地发现由于数据被非法修改而导致的程序错误,提高程序调试效率。

[0068] 以上所述仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明技术原理的前提下,还可以做出若干改进和变形,这些改进和变形

也应视为本发明的保护范围。

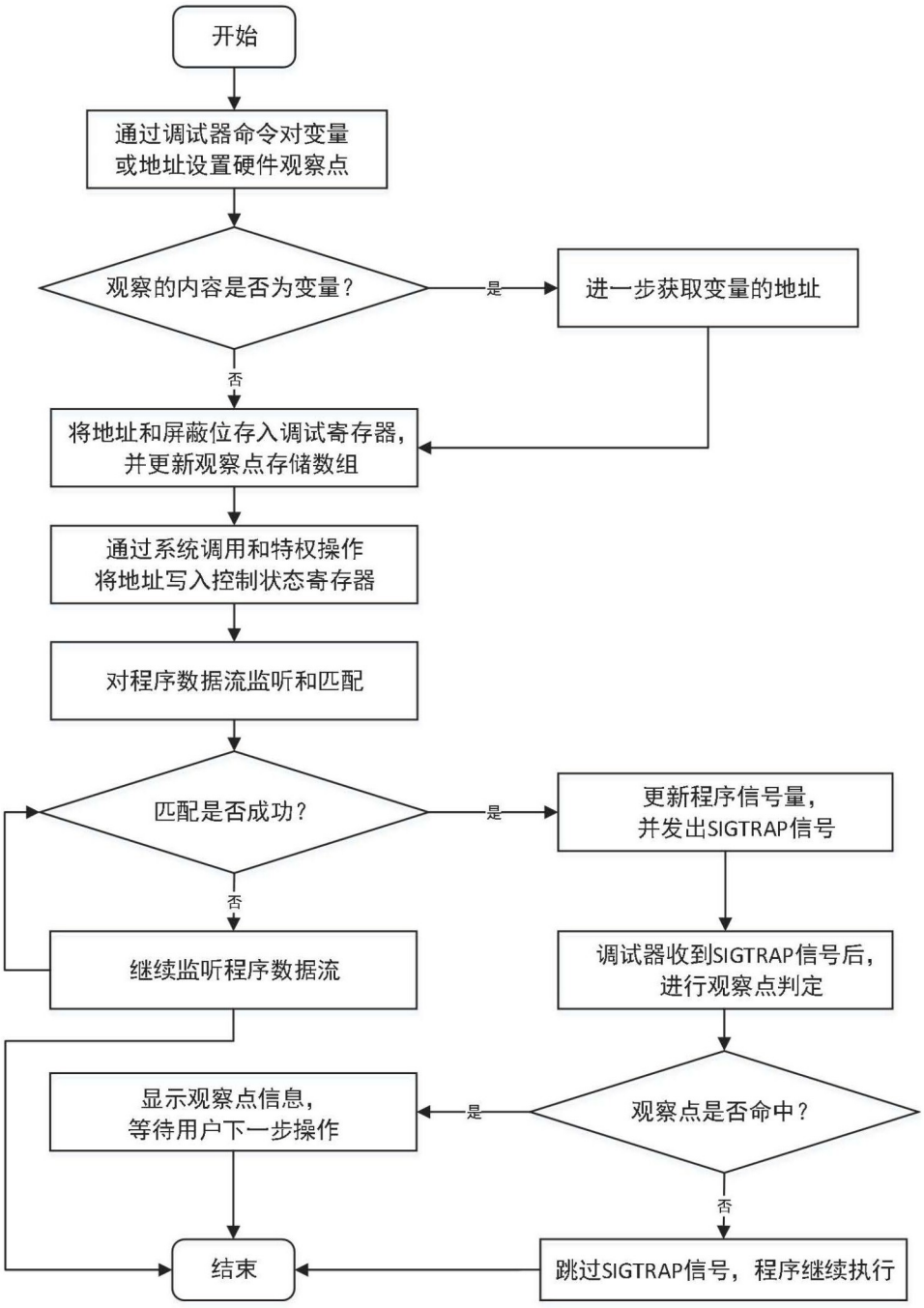

开始
通过调试器命令对变量
或地址设置硬件观察点
观察的内容是否为变量？
是
进一步获取变量的地址
否
将地址和屏蔽位存入调试寄存器，
并更新观察点存储数组
通过系统调用和特权操作
将地址写入控制状态寄存器
对程序数据流监听和匹配
匹配是否成功？
是
更新程序信号量，
并发出SIGTRAP信号
否
继续监听程序数据流
调试器收到SIGTRAP信号后，
进行观察点判定
显示观察点信息，
等待用户下一步操作
是
观察点是否命中？
否
结束
跳过SIGTRAP信号，程序继续执行

图1


观察地址
生效标志
其他信息
观察点1
观察点2
.....
观察点n

图2

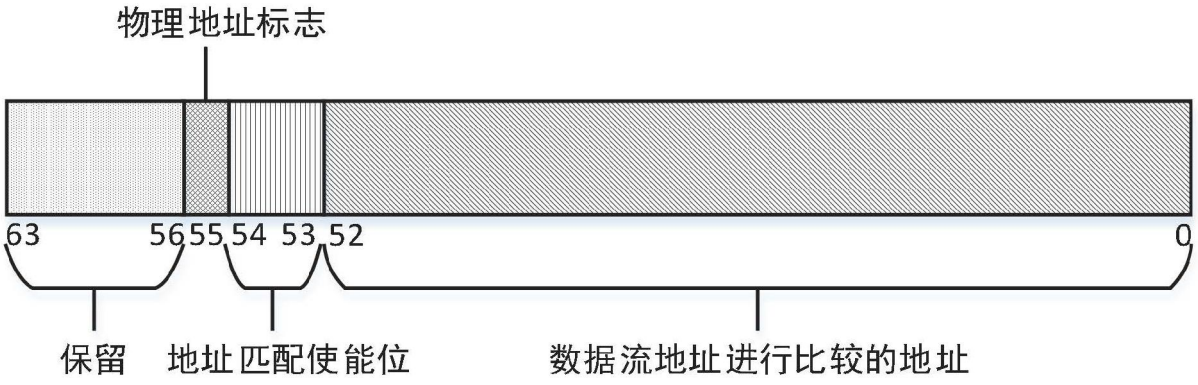

物理地址标志
63
56
55
54
53
52
0
保留
地址匹配使能位
数据流地址进行比较的地址

图3


63
53
52
0
保留
数据流地址匹配屏蔽位

图4