(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 107466400 B
(45)授权公告日 2020.12.04

(21)申请号 201680021722.2

(22)申请日 2016.04.07

(65)同一申请的已公布的文献号
申请公布号 CN 107466400 A

(43)申请公布日 2017.12.12

(30)优先权数据
1553369 2015.04.16 FR

(85)PCT国际申请进入国家阶段日
2017.10.13

(86)PCT国际申请的申请数据
PCT/FR2016/050801 2016.04.07

(87)PCT国际申请的公布数据
WO2016/166450 FR 2016.10.20

(73)专利权人 拉姆伯斯公司
地址 美国加利福尼亚州

(72)发明人 V·杜帕奎斯 S·戈津斯基

(74)专利代理机构 北京市金杜律师事务所 11256
代理人 王茂华

(51)Int.Cl.
G06F 12/14(2006.01)
G06F 21/64(2013.01)
G06F 21/74(2013.01)
G06F 21/79(2013.01)

(56)对比文件
US 7555617 B2,2009.06.30
US 7616206 B1,2009.11.10
JP 特开平6-30245 A,1994.02.04
US 2010/0106758 A1,2010.04.29
US 2007/0157030 A1,2007.07.05
CN 103597456 A,2014.02.19
CN 104484284 A,2015.04.01
CN 104221028 A,2014.12.17
CN 1802813 A,2006.07.12
CN 1445680 A,2003.10.01
CN 1550995 A,2004.12.01
FR 2989801 A1,2013.10.25

审查员 刘畅

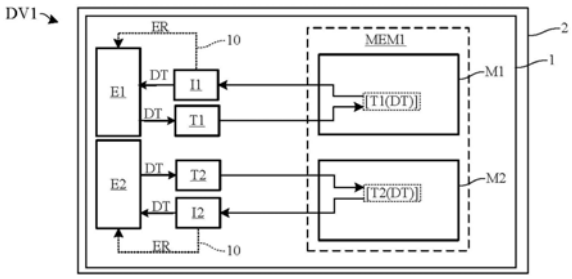
权利要求书4页 说明书11页 附图5页

(54)发明名称

用于在至少两个功能实体之间共享存储器的方法

(57)摘要

本发明涉及一种用于在两个功能实体(E1,E2)之间共享存储器(MEM1)的方法,包括以下步骤:向第一功能实体分配第一数据变换函数(T1)和第一逆变换函数(I1),并向第二功能实体(E2)分配第二数据变换函数(T2)和第二逆变换函数(I2),第二逆变换函数(I2)与第一变换函数(T1)不兼容,并且第一逆变换函数(I1)与第二变换函数(T2)不兼容。



CN 107466400 B

1.一种方法，用于共享装置中的存储器，所述装置包括在同一支架上或在同一壳体中布置的至少一个存储器和至少两个功能实体，所述至少两个功能实体包括第一功能实体和第二功能实体，所述方法包括步骤：

配置第一功能实体，使得其在所述存储器的至少第一存储器区域中写入和读取数据，

配置第二功能实体，使得其在所述存储器的至少第二存储器区域中写入和读取数据，所述第二存储器区域与所述第一存储器区域不相交，

所述方法的特征在于其包括步骤：

向所述第一功能实体分配第一变换函数和被配置为恢复和/或验证由所述第一变换函数变换的数据的有效性的第一逆变换函数，

向所述第二功能实体分配第二变换函数和被配置为恢复和/或验证由所述第二变换函数变换的数据的有效性的第二逆变换函数，其中所述第二逆变换函数与所述第一变换函数不兼容，并且所述第一逆变换函数与所述第二变换函数不兼容，

配置所述第一功能实体，使得其在将数据字写入所述第一存储器区域之前将所述第一变换函数应用到所述数据字，以及将所述第一逆变换函数应用到在所述第一存储器区域中读取的数据字，以及

配置所述第二功能实体，使得其在将数据字写入所述第二存储器区域之前将所述第二变换函数应用到所述数据字，并将所述第二逆变换函数应用到由所述第二功能实体在所述第二存储器区域中读取的数据字。

2.根据权利要求1所述的方法，包括以下步骤：配置至少所述第二逆变换函数，以当将所述第二逆变换函数应用到借助于所述第一变换函数变换的数据时提供错误状态。

3.根据权利要求2所述的方法，包括以下步骤：当在读取所述存储器中的数据字之后所述第二逆变换函数提供所述错误状态时，执行免受所述第二功能实体在所述第一存储器区域中读取数据的尝试的保护动作。

4.根据权利要求1至3中任一项所述的方法，包括以下步骤：

配置所述第一功能实体，使得其还将数据写入第三存储器区域，所述第三存储器区域与所述第一存储器区域和所述第二存储器区域不相交，

配置所述第二功能实体，使得其还读取所述第三存储器区域中的数据，

除了所述第一变换函数之外，向所述第一功能实体分配所述第二变换函数，而不向所述第一功能实体分配所述第二逆变换函数，以及

配置所述第一功能实体以将所述第二变换函数应用到被写入所述第三存储器区域并期望用于所述第二功能实体的数据。

5.根据权利要求4所述的方法，包括以下步骤：

配置所述第一功能实体，使得其还读取所述第三存储器区域中的数据，

配置所述第二功能实体，使得其还将数据写入所述第三存储器区域，

除了所述第二变换函数之外，向所述第二功能实体分配所述第一变换函数，而不向所述第二功能实体分配所述第一逆变换函数，以及

配置所述第二功能实体，使得其将所述第一变换函数应用到期望用于被写入所述第三存储器区域中的所述第一功能实体的数据。

6.根据权利要求1至3中任一项所述的方法，包括以下步骤：

配置所述第一功能实体和第二功能实体，使得它们在第三存储器区域中写入和读取数据，所述第三存储器区域与所述第一存储器区域和所述第二存储器区域不相交，

除了所述第一变换函数和所述第一逆变换函数之外，向所述第一功能实体分配所述第二变换函数和所述第二逆变换函数，

不向所述第二功能实体分配所述第一变换函数或所述第一逆变换函数，

配置所述第一功能实体以将所述第二变换函数应用到被写入所述第三存储器区域并期望用于所述第二功能实体的数据。

7.根据权利要求1至3或5中任一项所述的方法，包括以下步骤：

配置所述第一功能实体，使得所述第一变换函数包括向要被写入的数据字添加校验位，并且所述第一逆变换函数验证从所述存储器读取的数据字具有有效的校验位，以及

配置所述第二功能实体，使得所述第二变换函数包括向要被写入的数据字添加逆校验位，并且所述第二逆变换函数验证从所述存储器读取的数据字具有有效的逆校验位。

8.根据权利要求1至3或5中任一项所述的方法，包括以下步骤：配置至少一个功能实体使得其变换函数包括编码函数。

9.根据权利要求8所述的方法，包括以下步骤：使用用于将数据写入所述存储器的地址作为数据编码变量，来提供编码函数。

10.根据权利要求1至3、5或9中任一项所述的方法，包括以下步骤：配置至少一个功能实体，使得其变换函数包括签名函数，并且其逆变换函数包括在读取尚未借助于所述签名函数变换的数据时提供错误状态的签名验证函数。

11.根据权利要求1至3、5或9中任一项所述的方法，其中所述功能实体中的至少一个功能实体从由直接存储器访问控制器、信号处理器或由一个或多个处理器执行的软件功能组成的组中选择。

12.一种装置，包括在同一支架上或在同一壳体中布置的至少一个存储器和至少两个功能实体，所述至少两个功能实体包括第一功能实体和第二功能实体，其中：

第一功能实体被配置为在所述存储器的至少第一存储器区域中写入或读取数据，

第二功能实体被配置为在所述存储器的至少第二存储器区域中写入或读取数据，所述第二存储器区域与所述第一存储器区域不相交，

所述装置的特征在于：

所述第一功能实体包括第一变换函数和用于恢复和/或验证由所述第一变换函数变换的数据的有效性的第一逆变换函数，

所述第二功能实体包括第二变换函数和用于恢复和/或验证由所述第二变换函数变换的数据的有效性的第二逆变换函数，其中所述第二逆变换函数与所述第一变换函数不兼容，并且所述第一逆变换函数与所述第二变换函数不兼容，

所述第一功能实体被配置为在将数据字写入所述第一存储器区域之前将所述第一变换函数应用到所述数据字，并将所述第一逆变换函数应用到在所述第一存储器区域中读取的数据，以及

所述第二功能实体被配置为在将数据字写入所述第二存储器区域之前将所述第二变换函数应用到所述数据字，并将所述第二逆变换函数应用到在所述第二存储器区域中读取的数据。

13.根据权利要求12所述的装置，其中所述第二逆变换函数被配置为在由所述第一变换函数变换的数据的逆变换期间提供错误状态。

14.根据权利要求13所述的装置，其中所述第二功能实体被配置为当所述第二逆变换函数在由所述第二功能实体读取所述存储器中的数据之后提供错误状态时执行免受由所述第二功能实体读取所述第一存储器区域中的数据的尝试的保护动作。

15.根据权利要求12至14中任一项所述的装置，其中：

所述第一功能实体被配置为还将数据写入第三存储器区域，所述第三存储器区域与所述第一存储器区域和所述第二存储器区域不相交，

所述第二功能实体被配置为还读取所述第三存储器区域中的数据，

所述第一功能实体除所述第一变换函数之外还包括所述第二变换函数，但不包括所述第二逆变换函数，以及

所述第一功能实体被配置为将所述第二变换函数应用到被写入所述第三存储器区域并期望用于所述第二功能实体的数据。

16.根据权利要求15所述的装置，其中：

所述第一功能实体被配置为还读取所述第三存储器区域中的数据，

所述第二功能实体被配置为还将数据写入所述第三存储器区域，

所述第二功能实体除所述第二变换函数之外还包括所述第一变换函数，但不包括所述第一逆变换函数，以及

所述第二功能实体被配置为将所述第一变换函数应用到期望被写入所述第三存储器区域中的所述第一功能实体的数据。

17.根据权利要求12至14中任一项所述的装置，其中：

所述第一功能实体和所述第二功能实体被配置为在第三存储器区域中写入和读取数据，所述第三存储器区域与所述第一存储器区域和所述第二存储器区域不相交，

所述第一功能实体除所述第一变换函数和所述第一逆变换函数之外还包括所述第二变换函数和所述第二逆变换函数，

所述第二功能实体既不包括所述第一变换函数也不包括所述第一逆变换函数，以及

所述第一功能实体被配置为将所述第二变换函数应用到被写入所述第三存储器区域并期望用于所述第二功能实体的数据。

18.根据权利要求12至14或16中任一项所述的装置，其中：

所述第一变换函数包括向要被写入的数据字添加校验位，并且所述第一逆变换函数包括验证从所述存储器读取的数据字具有有效的校验位，以及

所述第二变换函数包括向要被写入的数据字添加逆校验位，并且所述第二逆变换函数包括验证在所述存储器中读取的数据字具有有效的逆校验位。

19.根据权利要求12至14或16中任一项所述的装置，其中至少一个功能实体的变换函数包括编码函数。

20.根据权利要求19所述的装置，其中所述编码函数使用用于将数据写入所述存储器的地址作为用于编码数据的变量。

21.根据权利要求12至14、16或20中任一项所述的装置，其中至少一个功能实体的变换函数包括签名函数，并且所述至少一个功能实体的逆变换函数包括当读取尚未由签名函数

变换的数据时提供错误状态的签名验证函数。

22.根据权利要求12至14、16或20中任一项所述的装置，其中所述功能实体中的至少一个功能实体从由直接存储器访问控制器、信号处理器或由一个或多个处理器执行的软件功能组成的组中选择。

用于在至少两个功能实体之间共享存储器的方法

技术领域

[0001] 本发明涉及一种用于共享装置中的存储器的方法，该装置包括在同一支架上或同一壳体中布置的至少一个存储器和至少两个功能实体，第一功能实体被配置为在存储器的至少第一区域中写入和读取数据，并且第二功能实体被配置为在存储器的至少第二区域中写入和读取数据。

背景技术

[0002] 上述类型的装置中的存储器的共享在寻求一定程度的安全性时引起关于存储在存储器中的数据的各种问题。由于它们的组件被布置在同一支架或者至少在同一壳体中并且因此是物理上可访问的，所以这种装置易于遭受欺诈者的攻击，该欺诈者可能尝试控制功能实体来访问被分配给另一个功能实体的存储器区域。除了控制风险之外，另一种已知类型的攻击是错误注入，例如借助于激光束，使欺诈者能够读取特定存储器区域中的数据或损坏数据。

[0003] 为了克服这种类型的攻击，已经开发出所谓的存储器保护单元(MPU)，其执行一定数量的检查并且验证特定实体被授权在存储器的给定区域中写入或读取数据。例如，ARM®CortexTM-M3处理器被配备有可编程MPU，其中声明了存储器空间的区域，以及其属性(只读或读/写访问、权限、存储器区域的类型：可共享，缓冲区，高速缓存等)和与其相关联的访问权限。

[0004] 然而，MPU不允许抵消所有已知的攻击，并且特别是地址总线上的错误注入。此外，可以期望提供具有提供令人满意的安全程度的多个功能实体的装置，而不需要存储器保护单元。

[0005] 因此，可以期望提供一种用于改进共享存储器中的数据的安全性的方法，以替换或补充由MPU提供的保护。

发明内容

[0006] 因此，本发明的实施例涉及用于在包括在同一支架上或同一壳体中布置的至少一个存储器和至少两个功能实体的装置中共享存储器的方法，包括以下步骤：配置第一功能实体，使得其在存储器的至少第一区域中写入和读取数据，配置第二功能实体，使得其在与第一存储器区域不相交的存储器的至少第二区域中写入和读取数据，向第一功能实体分配第一数据变换函数和被配置为恢复和/或验证由第一变换函数变换的数据的有效性的第一逆数据变换函数，向第二功能实体分配第二数据变换函数和被配置为恢复和/或验证由第二变换函数变换的数据的有效性的第二逆数据变换函数，其中第二逆变换函数与第一变换函数不兼容，并且第一逆变换函数与第二变换函数不兼容，配置第一功能实体，使得其在将数据字写入第一存储器区域之前将第一变换函数应用到数据字，以及将第一逆变换函数应用到在第一存储器区域中读取的数据字，以及配置第二功能实体，使得其在将数据字写入第二存储器区域之前将第二变换函数应用到数据字，并将第二逆变换函数应用到由第二功

能实体在第二存储器区域中读取的数据字。

[0007] 根据实施例，该方法包括步骤：配置至少第二逆变换函数，以在将其应用到借助于第一变换函数变换的数据时提供错误状态。

[0008] 根据实施例，该方法包括步骤：当在读取存储器中的数据字之后第二逆变换函数提供错误状态时，执行防止由第二功能实体读取第一存储器区域中的数据的尝试的保护动作。

[0009] 根据实施例，该方法包括步骤：配置第一功能实体，使得其还将数据写入第三存储器区域，配置第二功能实体，使得其还读取第三存储器区域中的数据，除了第一变换函数之外，向第一功能单元分配第二变换函数，而不向第一功能单元分配第二逆变换函数，并且配置第一功能实体以将第二变换函数应用到写入第三存储器区域并期望用于第二功能实体的数据。

[0010] 根据实施例，该方法包括步骤：配置第一功能实体，使得其还读取第三存储器区域中的数据，配置第二功能实体，使得其还将数据写入第三存储器区域，除了第二变换函数之外，向第二功能元件分配第一变换函数，而不向第二功能元件分配第一逆变换函数，并且配置第二功能实体，使得其将第一变换函数应用到期望用于被写入第三存储器区域中的第一功能实体的数据。

[0011] 根据实施例，该方法包括步骤：配置第一和第二功能实体，使得它们在第三存储器区域中写入和读取数据，除了第一变换函数和第一逆变换函数之外，向第一功能实体分配第二变换函数和第二逆变换函数，而不向第二功能实体分配第一变换函数或第一逆变换函数，配置第一功能实体以将第二变换函数应用到被写入第三存储器区域以及期望用于第二功能实体的数据。

[0012] 根据实施例，该方法包括步骤：配置第一功能实体，使得第一变换函数包括向要被写入的数据字添加校验位，并且第一逆变换函数验证从存储器读取的数据字具有有效的校验位，并且配置第二功能实体，使得第二变换函数包括向要被写入的数据字添加逆校验位，并且第二逆变换函数验证从存储器读取的数据字具有有效的逆校验位。

[0013] 根据实施例，该方法包括步骤：配置至少一个功能实体使得其变换函数包括数据编码函数。

[0014] 根据实施例，该方法包括步骤：使用用于将数据写入存储器的地址作为数据编码变量，来提供编码函数。

[0015] 根据实施例，该方法包括步骤：配置至少一个功能实体，使得其变换函数包括数据签名函数，并且其逆变换函数包括在读取尚未借助于签名函数变换的数据时提供错误状态的签名验证函数。

[0016] 根据实施例，功能实体中的至少一个功能实体选自由直接存储器存取控制器、信号处理器或由一个或多个处理器执行的软件功能组成的组。

[0017] 本发明的实施例同样涉及包括在同一支架上或同一壳体中布置的至少一个存储器和至少两个功能实体的装置，其中第一功能实体被配置为在存储器的至少第一区域中写入或读取数据，第二功能实体被配置为在与第一存储器区域不相交的存储器的至少第二区域中写入或读取数据，第一功能实体包括第一数据变换函数和用于恢复和/或验证由第一变换函数变换的数据的有效性的第一逆数据变换函数，第二功能实体包括第二数据变换函

数和用于恢复和/或验证由第二变换函数变换的数据的有效性的第二逆数据变换函数，其中第二逆变换函数与第一变换函数不兼容，并且第一逆变换函数与第二变换函数不兼容，第一功能实体被配置为在将数据字写入第一存储器区域之前将第一变换函数应用于数据字，并且将第一逆变换函数应用到在第一存储器区域中读取的数据，并且第二功能实体被配置为在将数据字写入第二存储器区域之前将第二变换函数应用到数据字，并且将第二逆变换函数应用到在第二存储器区域中读取的数据。

[0018] 根据实施例，第二逆变换函数被配置为在由第一变换函数变换的数据的逆变换期间提供错误状态。

[0019] 根据实施例，第二功能实体被配置为当第二逆变换函数在由第二功能实体读取在存储器中的数据之后提供错误状态时，执行防止由第二功能实体读取第一存储器区域中的数据的尝试的保护动作。

[0020] 根据实施例，第一功能实体被配置为还将数据写入第三存储器区域中，第二功能实体被配置为还读取第三存储器区域中的数据，第一功能实体除第一变换函数之外还包括第二变换函数，但不包括第二逆变换函数，并且第一功能实体被配置为将第二变换函数应用到被写入第三存储器区域并期望用于第二功能实体的数据。

[0021] 根据实施例，第一功能实体被配置为还读取第三存储器区域中的数据，第二功能实体被配置为还将数据写入第三存储器区域，第二功能实体除第二变换函数之外还包括第一变换函数，但不包括第一逆变换函数，并且第二功能实体被配置为将第一变换函数应用到期望被写入第三存储器区域中的第一功能实体的数据。

[0022] 根据实施例，第一和第二功能实体被配置为在第三存储器区域中写入和读取数据，第一功能实体除第一变换函数和第一逆变换函数之外还包括第二变换函数和第二逆变换函数，第二功能实体既不包括第一变换函数也不包括第一逆变换函数，并且第一功能实体被配置为将第二变换函数应用到写入第三存储器区域并期望用于第二功能实体的数据。

[0023] 根据实施例，第一变换函数包括向要被写入的数据字添加校验位，并且第一逆变换函数包括验证从存储器读取的数据字具有有效的校验位，并且第二变换函数包括向要被写入的数据字添加逆校验位，并且第二逆变换函数包括验证在存储器中读取的数据字具有有效的逆校验位。

[0024] 根据实施例，至少一个功能实体的变换函数包括数据编码函数。

[0025] 根据实施例，编码函数使用将数据写入存储器的地址作为用于编码数据的变量。

[0026] 根据实施例，至少一个功能实体的变换函数包括数据签名函数，并且逆变换函数包括当读取尚未由签名函数变换的数据时提供错误状态的签名验证函数。

[0027] 根据实施例，功能实体中的至少一个功能实体选自由直接存储器存取控制器、信号处理器或由一个或多个处理器执行的软件功能组成的组。

附图说明

[0028] 从以下仅为了示例性目的提供并在附图中表示的本发明的特定实施例的描述中，其它优点和特征将变得更加清楚，在附图中：

[0029] 图1是根据本发明的装置的第一实施例的框图，

[0030] 图2示出了在图1中以框图示出的数据变换函数的实施例，

[0031] 图3A和图3B示出了通过根据本发明的方法抵消的对图1的装置的攻击的示例，

[0032] 图4是根据本发明的装置的第二实施例的框图，

[0033] 图5示出了图4的装置的功能实体的示例性架构，

[0034] 图6示出了图5的功能实体的示例性实施方式，

[0035] 图7是根据本发明的装置的第三实施例的框图，

[0036] 图8示出了图7的装置的示例性架构。

具体实施方式

[0037] 图1是根据本发明的装置DV1的框图，该装置DV1包括两个功能实体E1、E2和存储器MEM1，例如随机存取存储器(RAM)。存储器包括分别专用于功能实体E1和E2的两个不同且不相交的存储器区域M1和M2，每一个功能实体被配置为在分配给它的存储器区域中写入和读取数据DT。

[0038] 功能实体E1、E2和存储器MEM1被布置在相同的互连支架1上，或在装置的壳体2中布置并互连的不同的互连支架上。如果功能实体和存储器被集成在同一半导体芯片上，则互连支架1可以是半导体芯片，或者如果功能实体和存储器被集成在不同的半导体芯片上，则互连支架1可以是印刷电路板。功能实体和/或存储器同样可以被布置在不同的印刷电路板上。功能实体E1、E2可以是两个不同的处理器，例如主处理器和外围处理器，具有并行操作的若干物理核的多核处理器的两个CPU，由同一处理器或不同的处理器执行的程序(服务、应用程序)，不具有与存储器相同的访问权限的同一处理器的不同用户，例如管理员用户和非管理员用户等，或这些各种类型的功能实体的组合。存储器共享允许每个功能实体具有可以存储应用程序数据、变量、代码等的专用空间。

[0039] 为了保护每一个存储器区域M1、M2中存在的数据和/或由该存储器区域尚未被分配到的实体检测每一个存储器区域中的未经授权的读取，向每一个功能实体E1、E2提供数据变换函数(分别为T1、T2)和逆变换函数(分别为I1、I2)。在图1和下面的附图中，函数T1、I1、T2、I2被表示在指定功能实体E1、E2的块之外，但也可以被认为是功能实体E1、E2的组成部分。

[0040] 每一个功能实体E1、E2被配置为经由被分配给它们的变换函数T1、T2向保留到其中的存储器区域写入数据，并且经由被分配给它们的逆变换函数I1、I2读取保留在其中的存储器区域中的数据。

[0041] 每一个变换函数T1、T2被设计成分别从要写入存储器的数据DT提供变换数据T1(DT)、T2(DT)。因此，功能实体E1将数据T1(DT)写入存储器区域M1，并且功能实体E2将数据T2(DT)写入存储器区域M2。经变换的数据可以是通过编码变换的初始数据DT或初始签名的数据，或通过编码变换并且然后签名的初始数据，或签名并且然后通过编码变换的初始数据。

[0042] 每一个逆变换函数I1、I2被设计为从变换的数据T1(DT)、T2(DT)恢复和/或验证初始数据DT的有效性，因此，在初始数据的恢复的情况下：

[0043] I1(T1(DT))＝DT

[0044] I2(T2(DT))＝DT

[0045] 或者，在对数据T1(DT)、T2(DT)的有效性进行验证的情况下：

[0046] I1(T1(DT))＝OK(有效性被确认)

[0047] I2(T2(DT))＝OK(有效性被确认)

[0048] 或者,在对存储器中读取的数据的有效性进行恢复和验证的情况下:

[0049] I1(T1(DT))＝DT并且I1(T1(DT))＝OK

[0050] I2(T2(DT))＝DT并且I2(T2(DT))＝OK

[0051] 此外,功能实体E1、E2的变换和逆变换函数被设计成从一个实体到另一个实体不兼容,逆变换函数I1与变换函数T2不兼容,并且逆变换函数I2与变换函数T1不兼容。因此,逆变换函数I1不允许恢复和/或验证由变换函数T2变换的数据DT的有效性,并且逆变换函数I2不允许恢复和/或验证由变换函数T1变换的数据的有效性。因此:

[0052] -在初始数据的恢复的情况下,由逆变换函数(表示为DT*)产生的数据是错误的并且不同于初始数据DT:

[0053] I1(T2(DT))＝DT*

[0054] I2(T1(DT))＝DT*

[0055] -在对与所使用的逆变换函数不对应的变换函数提供的数据的有效性的验证的情况下,产生错误状态“ER”:

[0056] I1(T2(DT))＝ER

[0057] I2(T1(DT))＝ER

[0058] -在恢复初始数据和检查由与所使用的逆变换函数不对应的变换函数产生的数据的有效性的情况下,提供错误数据DT*以及错误状态:

[0059] I1(T2(DT))＝DT*并且I1(T2(DT))＝ER

[0060] I2(T1(DT))＝DT*并且I2(T1(DT))＝ER

[0061] 函数T1、T2、I1、I2可以被实现为在将功能实体E1、E2连接到存储器MEM1(例如硬连线逻辑电路)的数据路径中布置的硬件电路。在函数I1、I2验证在存储器区域中读取的数据的有效性(也就是说,确保数据已经被相应的变换函数变换)的情况下,每一个提供以错误信号ER的形式的错误状态,其经由硬件链路10路由到相关联的实体,函数I1向功能实体E1提供错误信号ER,并且函数I2向功能实体E2提供错误信号ER。如果数据无效,则错误信号例如被设定为1。错误信号的发生引起免受所关注实体读取尚未分配给它的存储器区域中的数据的尝试的保护动作,例如硬件或软件块或所关注实体的复位,或者提供任意数据代替错误数据,因为尽管有错误,后者可能包含可用的信息。

[0062] 函数T1、T2、I1、I2也可以在软件中实现,例如,当在存储器MEM1中写入和读取数据时以由功能实体E1、E2调用的子例程的形式。用于保护程序的已知技术使得有可能防止功能实体E1、E2被欺诈者转移,以将数据写入存储器而不首先使用函数T1、T2进行变换,或者读取数据而不经过逆变换函数I1、I2。例如,实现函数T1、T2、I1、I2的子例程可以提供允许对存储器的写入或读取访问的签名,使得它们的执行需要访问存储器。在这种情况下,错误状态ER是软件,并且可以以类似于硬件错误信号的方式进行处理,以便执行针对由存储器区域尚未分配的功能实体读取存储器区域的保护动作,例如阻止检测读取错误的存在的实体的程序。例如,错误可能导致通过逆变换例程直接调用设计为执行此动作的子例程。

[0063] 图2示出了装置DV1的示例性实施例,其中每一个变换函数T1、T2包括编码函数(分别为Fc1、Fc2)和签名函数(分别为Fs1、Fs2)。相反,每一个逆变换函数I1、I2包括解码函数

(分别为Fd1、Fd2)和签名验证函数(分别为Fv1、Fv2)。

[0064] 在图2中,签名函数Fs1、Fs2位于编码函数Fc1、Fc2之后,使得变换函数T1提供被签名和编码的数据Fc1(DT)//S1(其中//是指示级联的符号,并且S1是由函数Fs1提供的签名)。类似地,变换函数T2提供被存储在存储器区域M2中的被签名和编码的数据Fc2(DT)//S2(S2是由函数Fs2提供的签名)。

[0065] 因此,在解码函数Fc1、Fc2之前执行签名检查函数Fv1、Fv2,使得验证函数Fv1在解码函数Fd1恢复数据DT之前验证编码数据Fc1(DT)的签名S1的有效性,Fd1使得:

[0066] Fd1(Fc1(DT))=DT

[0067] 类似地,验证函数Fv2在解码函数Fd2恢复数据DT之前验证编码数据Fc2(DT)的签名S2的有效性,Fd2使得:

[0068] Fd2(Fc2(DT))=DT

[0069] 根据替代方案,在编码函数Fc1、Fc2之前执行签名函数Fs1、Fs2,使得变换函数T1提供被编码的签名数据Fc1(DT//S1),并且变换函数T2提供被编码的签名数据Fc2(DT//S2)。然后在验证其签名之前对数据进行解码。签名和编码操作的执行同样可以在两个变换函数之间反转,一个在签名前对数据进行编码,而另一个在编码之前对其进行签名。

[0070] 函数I1与函数T2的不兼容性以及函数I2与函数T1的不兼容性导致在签名验证函数Fv1和签名函数Fs2之间以及在签名验证函数Fv2和签名函数Fs1之间以及在解码函数Fd1和编码函数Fc2之间以及在解码函数Fd2和编码函数Fc1之间的不兼容性。换句话说:

[0071] -通过签名函数Fs2生成的签名S2的由函数Fv1的验证导致错误状态ER的发送,

[0072] -通过签名函数Fs1生成的签名S1的由函数Fv2的验证导致错误状态ER的发送,

[0073] -通过函数Fc2编码的数据Fc2(DT)的由函数Fd1的解码产生错误数据DT*,以及

[0074] -通过函数Fc1编码的数据Fc1(DT)的由函数Fd2的解码产生错误数据DT*。

[0075] 在一些实施例中,变换函数T1可以仅包括编码函数Fc1或仅包括签名函数Fs1。变换函数T2可以仅包括编码函数Fc2或仅包括签名函数Fs2。逆变换函数I1可以仅包括解码函数Fd1或仅包括签名验证函数Fv1。逆变换函数I2可以仅包括解码函数Fd2或仅包括签名验证函数Fv2。

[0076] 在本发明的实施例中,两个函数Fc1、Fc2使用相同的编码算法,该算法取决于对于每个函数不同的编码密钥来实现代码。该密钥可以仅包括一位或多位。编码可能是复杂的并且使用标准化或“专有”密码算法,或者相反地非常简单,并且包括例如根据作为密钥的函数的加扰规则对数据的位进行加扰。

[0077] 在一个实施例中,编码如下:

[0078] -函数Fc1不反转数据DT的位的极性,并向数据添加等于0的最高有效位,这意味着位的极性不被反转,并且

[0079] -函数Fc2将数据DT的位的极性反转(每个等于1的位变为等于0,并且每个等于0的位变为等于1),并向数据添加等于1的最高有效位,这意味着位的极性已反转。

[0080] 在这种情况下,解码函数Fd1不修改数据,并且可以可选地验证最高有效位等于1,否则,除了由签名验证函数提供的签名错误状态之外还发出编码错误状态。另一方面,解码函数Fd2包括步骤:反转数据DT的所有位的极性,并且可选地验证最高有效位等于0,如果不是,则发送编码错误状态。

[0081] 根据替代方案：

[0082] -编码函数Fc1生成随机极性的位为0或1，如果随机极性的位等于1则反转数据的位，并且如果随机极性的位等于0则不反转数据的位，并将极性的位作为数据中的最高有效位，

[0083] -编码函数Fc2生成随机极性的位为0或1，并且如果随机极性的位等于0(而不是函数Fc1的1)，则通过反转数据的位，并且如果随机极性的位等于1(而不是函数Fc1的0)，则不反转数据的位，来执行与前述编码规则相反的编码规则，并且插入极性位作为数据中的最高有效位，

[0084] -解码函数Fd1评估最高有效位，如果极性位等于1则极性位反转数据的位，并且如果极性位等于0则不反转数据的位，并且从数据中移除极性位，

[0085] -解码函数Fd2评估最高有效位，如果极性位等于0则极性位反转数据的位，并且如果极性位等于1则不反转数据位的值，并从数据中移除极性位。

[0086] 在该替代方案中，解码函数未检测到解码错误。读取由另一实体写入的数据的实体仅仅接收到随机错误的数据。

[0087] 在一些实施例中，编码函数Fc1、Fc2除了使用数据DT外，还使用要写入数据的地址AD作为输入数据。在该情况下，变换数据DT可以表示为：

[0088] Fc1(AD,DT)或

[0089] Fc2(AD,DT)

[0090] 这种编码增强了装置对地址总线上的错误注入攻击的阻力。实际上，在这种情况下，使用读取指令中存在的地址来解码数据，该地址与由于错误注入而读取数据的地址不同。被解码的数据因此无效。

[0091] 本发明的实施例同样可以供已知的签名函数Fs1、Fs2，诸如标准错误检测码(EDC)或更复杂的纠错码(ECC)(诸如汉明码)使用。

[0092] 例如，函数Fs1可以包括向数据添加由CRC(循环冗余校验)码(其是数据的位的函数)形成的签名S1，并且函数Fs2可以包括向数据添加由逆CRC码或与密钥组合的CRC码形成的签名S2。在该情况下，签名验证函数Fv1重新计算数据的CRC码并将其与形成其签名S1的CRC码进行比较，而签名验证函数Fv2重新计算数据的CRC码，反转其位或将其与密钥组合，然后将其与形成签名S2的CRC码进行比较。

[0093] 在由于其简单性而有利的实施例中，函数Fs1包括对数据添加由校验位形成的签名，并且函数Fs2包括对数据添加由逆校验位形成的签名S2。

[0094] 在所有这些示例中，验证函数Fv1与签名函数Fs2不兼容，并且验证函数Fv2与签名函数Fs1不兼容，使得功能实体E1、E2对由其它功能实体签名的数据签名的验证导致错误状态。

[0095] 下面的表1提供了包括编码函数Fc1和签名函数Fs1的组合的变换函数T1的示例，编码函数Fc1没有极性变化并且添加等于0的极性位，签名函数Fs1包括添加校验位。以及，表1提供了包括编码函数Fc2和签名函数Fs2的组合的示例，编码函数Fc2改变数据位的极性并且添加等于1的极性位，签名函数Fs2包括添加逆校验位。

[0096]

表1	变换T1		变换T2	
	编码Fc1	签名Fs1	编码Fc2	签名Fs2
初始数据DT	添加等于0的极性位	对编码数据添加校验位	位极性反转并且添加等于1的极性位	对编码数据添加逆校验位
1000	10000	100001	01111	011111

[0097] 下面的表2示出了极性位未添加到数据的实施例。

[0098]

表2	变换T1		变换T2	
	编码Fc1	签名Fs1	编码Fc2	签名Fs2
初始数据DT	没有极性变化	对编码数据添加校验位	位极性反转	对编码数据添加逆校验位
1000	1000	10001	0111	01110

[0099] 在上述示例中，变换函数T1和T2依赖于相同的编码和/或签名算法，并且借助于形成密钥或诸如极性位的密钥的等价物的参数来区分。在其它实施例中，变换函数实现不同的编码算法，一种使用例如位混洗(bit shuffling)函数，而另一种使用极性反转函数。

[0100] 图3A、3B示出了违反存储器空间MEM1的示例，其中根据本发明的方法有助于抵消。在图3A的示例中，功能实体E2合法地将由函数T2变换的数据T2(DT)写入存储器区域M2。该数据然后由功能实体E1非法读取。因此它受到与变换函数T2不兼容的逆变换函数I1。结果，如果函数I1包括签名验证函数Fv1，而无论具有或不具有解码函数Fd1，则发送错误状态ER。如果函数I1包括解码函数Fd1，而无论具有或不具有签名验证函数Fv1，则由函数Fd1提供的读取数据DT*是错误的(或随机错误的)。如果错误状态ER包括错误检测(例如，当密钥或密钥的一部分被包括在数据中时，并且数据的位与密钥的位之间的关系的验证是可能的)，则同样可以由解码函数Fd1发送错误状态ER。如上所述，错误状态可以用于阻止功能实体E1，并且从而停止对无效数据的读取。

[0101] 在图3B的示例中，功能实体E2将由函数T2变换的数据T2(DT)非法写入功能实体E1的存储器区域M1中。然后，该数据由功能实体E1合法读取。因此，如前所述，其经受与变换函数T2不兼容的逆变换函数I1。后果类似于关于图3A描述的那些。

[0102] 图4是示出根据本发明的装置的另一实施例DV2的框图，除了区域M1、M2之外，在存储器MEM1中还提供了由两个功能实体E1，E2共享的存储器区域M3。这里的目的是将存储器区域M3用作从功能实体E1到功能实体E2以及从功能实体E2到功能实体E1的数据交换区域。

[0103] 为此，功能实体E1被分配功能实体E2的变换函数T2，而没有逆变换函数I2，并且功能实体E2被分配功能实体E1的变换函数T1而没有逆变换函数I1。功能实体E1使用变换函数T2将期望用于功能实体E2的数据写入存储器区域M3，并且继续使用变换函数T1将数据写入存储器区域M1。一旦写入存储器区域M3，由于功能实体E1不具有逆变换函数I2，所以数据T2(DT)不能被功能实体E1读取，并且只能由功能实体E2读取。相反，功能实体E2使用变换函数T1将期望用于功能实体E1的数据写入存储器区域M3，并且继续使用变换函数T2将数据写入存储器区域M2。一旦写入存储器区域M3，由于功能实体E2不具有逆变换函数I1，所以数据T1(DT)不能再被功能实体E2读取，并且只能由功能实体E1读取。

[0104] 通过选择信号SEL确定由功能实体E1对变换函数T1或T2的选择以及由功能实体E2

对变换函数T2或T1的选择，该选择信号SEL是要写入的数据的地址的函数。如果数据的地址位于存储器区域M3中，则该信号例如等于1，否则为0。在该情况下，如果信号SEL等于1，则功能实体E1选择函数T2，否则默认选择函数T1。如果信号SEL等于1，则功能实体E2自动选择函数T1，否则默认选择函数T2。

[0105] 在替代方案中，功能实体E2不具有变换函数T1，并且仅被授权读取存储器区域M3，或者相反地，功能实体E1不具有变换函数T2，并且仅被授权读取存储器区域M3。

[0106] 图5示出了装置DV2的示例性架构。如前所述，功能实体E1、E2和存储器MEM1被布置在相同的互连支架1上或在装置的壳体2中布置的不同的互连支架上。仅表示功能实体E1，功能实体E2具有相同的架构并且具有相同的布置，采用参考“I2”代替参考“I1”。装置DV2包括数据总线DTB、地址总线ADB和由控制信号CT控制的地址多路复用器15，地址多路复用器15的输出端连接到地址总线ADB。控制信号CT由总线仲裁器(未示出)提供，根据来自它们的请求，授权每一个实体对存储器MEM1的访问时段，并且管理访问优先级。

[0107] 存储器MEM1包括连接到数据总线DTB的数据输入/输出DIO和连接到地址总线ADB的地址输入AIN。功能实体E1包括连接到地址多路复用器15的输入的地址输出，地址多路复用器15的另一输入接收从功能实体E2(未示出)输出的地址。由SEL信号选择的功能实体E1的变换函数T1、T2中的每者包括连接到数据总线DTB的输出，连接到功能实体E1的数据输出的输入，以向存储器MEM1提供要写入的数据DT。逆变换函数I1包括连接到数据总线DTB以接收变换数据T1(DT)的数据输入和连接到功能实体E1的数据输入的输出，以便向功能实体E1提供存在于变换数据T1(DT)中的数据DT。可选地，函数T1、T2和函数I1还包括连接到功能实体E1的地址输出的输入，以便使用数据地址AD作为将数据DT变换成数据T1(DT)或T2(DT)的输入数据，及其解码作为地址AD的函数。

[0108] 装置DV2同样包括地址检测器ADT，地址检测器ADT具有连接到地址总线ADB的地址输入并将选择信号SEL提供给功能实体E1和E2。该地址检测器例如是具有安全访问的可编程电路，其包括经由数据总线DTB可配置的地址寄存器(未示出)，如图所示，或经由安全专用链路。这些寄存器可以定义共享区域M3的开始和结束地址。

[0109] 尽管在此描述的示例仅提供共享区域M3，但是地址检测器ADT可能更复杂，并允许对若干地址空间进行编程。检测器同样可以提供专用于功能实体E1的选择信号SEL1和专用于功能实体E2的选择信号SEL2。因此，地址检测器类似于简单类型的存储器保护单元，其没有监测指令和控制功能实体E1、E2的访问权限的通常功能，在此通过提供变换和逆变换函数而提供存储器MEM1的安全性。在替代实施例中，根据本发明的存储器共享方法与使用实际存储器保护单元(MPU)组合，然后可以将其配置为提供SEL信号，并且从而替换地址检测器ADT。

[0110] 图6示出了图5的功能实体E1的实施例，其中变换函数T1包括编码函数Fc1和签名函数Fs1，变换函数T2包括编码函数Fc2和签名函数Fs2，并且逆变换函数I1包括解码函数Fd1和签名验证函数Fv1。

[0111] 编码函数Fc1、Fc2和解码函数Fd1以框图表示，并且可以如上述以对于每个编码函数使用不同的密钥的硬连线逻辑算法的形式来实现。编码函数Fc1、Fc2中的每者具有连接到功能实体E1的数据输出以接收数据DT的输入，以及可选地连接到功能实体E1的地址输出以接收地址AD的输入，以及连接到数据总线DTB的输出。解码函数Fd1具有连接到数据总线

DTB的输入，以及可选地连接到功能实体E1的地址输出以接收地址AD的输入，以及连接到功能实体E1的数据输入的输出。在图中表示为不同的功能实体E1的数据输入和输出可以由同一双向端口形成。

[0112] 本文中借助于异或类型(“XOR”)的逻辑门G1来执行签名函数Fs1、Fs2，逻辑门G1具有接收信号SEL的输入和接收由函数Fc1、Fc2分别提供的编码数据Fc1(DT)、Fc2(DT)的位b0-bn-1的N个输入。取决于信号SEL的值，即0或1，门G1的输出提供编码数据Fc1(DT)的校验位或编码数据Fc2(DT)的逆校验位。校验位或逆校验位被连接到编码数据的位作为最高有效位“bn”，以形成签名编码数据b0-bn-1//bn，其中位bn形成签名S1或签名S2，即数据字Fc1(DT)//S1或Fc2(DT)//S2。连接可以包括向传送该位的N个磁道(track)添加传导磁道(conductive track)，该组合直接被连接到数据总线DTB，该数据总线DTB包括N+1位，或者经由缓冲电路(未示出)被连接到数据总线。

[0113] 本文中借助于异或类型的两个门G2、G3执行签名验证函数Fv1。门G2具有接收具有固定值0或1的配置位bc的输入，以及在读取存储器MEM1(未示出，参见图5)之后接收由数据总线DTB提供的编码数据字Fc1(DT)或Fc2(DT)的位b0-bn-1的N个输入。位b0-bn-1还被应用于解码函数Fd1的数据输入。配置位bc在这里等于0，并且门G2的输出提供作为数据位b0-bn-1的函数的校验位bn’，其被应用于门G3的输入。后者在另一个输入处接收从数据总线DTB取得的校验位bn，并且当校验位bn和bn’不同时提供等于1的错误信号ER，这表明在存储器中读取的数据在其存储期间已被破坏，或者数据已被功能实体E2写入存储器中，并且因此具有逆校验位。

[0114] 应当注意，在该示例性实施例中，功能实体E2的结构类似于图6的结构，其通过在门G1的输入处反转信号SEL，使得当SEL＝0时提供逆校验位，并且通过将配置位bc设定为1，使得门G2提供逆校验位而不是校验位。

[0115] 图7示出了根据本发明的装置的另一实施例DV3，其中功能实体E1是可信实体，例如被称为“主”实体的安全处理器的中央处理单元CPU。然后，功能实体E2是“从”实体，例如直接存储器访问处理器(DMA)。形成功能实体E2的DMA处理器包括被连接到各种外围处理器的串行输入SIN，其可以直接访问存储器MEM1而不通过功能实体E1。

[0116] 存储器包括与之前类似的可由两个功能实体E1、E2读取和写入的存储器区域M3。如图4的实施例中，本文的目的是使用存储器区域M3作为从功能实体E1到功能实体E2以及从功能实体E2到功能实体E1的数据交换区域。然而，与图4的实施例不同，功能实体E1除了其自身的变换函数T1及其逆变换函数I1之外，还被分配有功能实体E2的变换函数T2及其逆变换函数I2，而功能实体E2仅具有其自身的变换函数T2和逆变换函数I2。

[0117] 如图所示，功能实体E1使用变换函数T2向存储器区域M3写入期望用于功能实体E2的数据字T2(DT)，并且可以使用变换函数T1向存储器区域M1或存储器区域M3写入没有期望用于功能实体E2并且不能被其读取的数据字T1(DT)。一旦写入存储器区域M3，数据T2(DT)同样可以由功能实体E1读回，因为后者在这里是逆变换函数I2。

[0118] 此外，功能实体E2使用其变换函数T2来将期望用于功能实体E1的数据T2(DT)写入存储器区域M3，其中功能实体E1可以使用逆变换函数I2读取或在存储器区域M1中写入其自身的数据T2(DT)，但是功能实体E1同样可以在这里读取存储器的主(master)。

[0119] 如上所述，功能实体E1使用选择信号SEL，一方面在将数据字写入存储器区域M1或

M3时选择变换函数T1或变换函数T2(同样可以将数据写入到功能实体E2的存储器区域M2)，另一方面，当在存储器区域M1或M3中读取数据字时选择逆变换函数I1或逆变换函数I2。该选择信号在这里不是像前述由地址检测器ADT提供，而是由实体E1本身作为存储器的可信赖的主提供。

[0120] 图8示出了装置DV3的架构的示例。如前所述，功能实体E1、E2和存储器MEM1被布置在相同的互连支架1上或在装置的壳体2中布置的不同的互连支架上。如上所述，装置DV3包括数据总线DTB和地址总线ADB以及由控制信号CT控制的地址多路复用器15。控制信号CT在这里由功能实体E1提供，该功能实体E1在这里被指定为地址总线的仲裁器，并且根据功能实体E2或外围处理器的请求向功能实体E2和外围处理器授予对存储器MEM1的访问时段。

[0121] 如前所述，存储器MEM1包括被连接到数据总线DTB的数据输入/输出DIO和被连接到地址总线ADB的地址输入AIN。功能实体E1包括被连接到地址多路复用器15的输入的地址输出，地址多路复用器15另一输入接收功能实体E2的地址输出。功能实体E1的变换函数T1和T2中的每者具有被连接到数据总线DTB的输出和被连接到功能实体E1的数据输出的输入。功能实体E1的逆变换函数I1、I2中的每者包括被连接到数据总线的数据输入和被连接到功能实体E1的数据输入的输出。功能实体E2的变换函数T2包括被连接到数据总线DTB的输出和被连接到功能实体E2的数据输出的输入。功能实体E2的逆变换函数I2包括被连接到数据总线的数据输入和被连接到功能实体E2的数据输入的输出。如同前述实施例，本实施例同样可以以软件形式实现，输入和输出本身是软件。

[0122] 本领域技术人员将清楚，根据本发明的装置可以经受各种其它实施例。虽然已经在前面描述了双实体装置，但是根据本发明的存储器共享方法可以应用于具有大于2的多个实体的装置和若干共享存储器。存储器或多个存储器可以是易失性存储器(RAM)、可编程和电可擦除存储器(EEPROM，闪存)或其它类型的存储器。类似地，尽管已经描述了仅将存储器划分为两个或三个区域，但是根据本发明的装置可以包括多个私有存储器区域和多个共享存储器区域，并且同一实体可被分配若干个私有存储器区域。根据本发明的装置同样可以具有各种应用。根据本发明的装置可以例如形成移动电话、无线电电视解码器、智能卡等的全部或一部分。

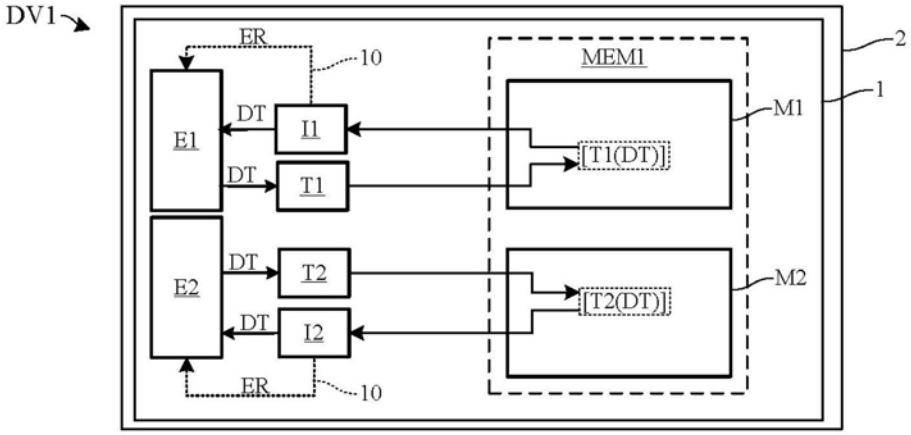

DV1
ER
10
MEM1
2
1
E1
DT
I1
M1
[T1(DT)]
DT
T1
E2
DT
T2
M2
[T2(DT)]
DT
I2
ER
10

图1

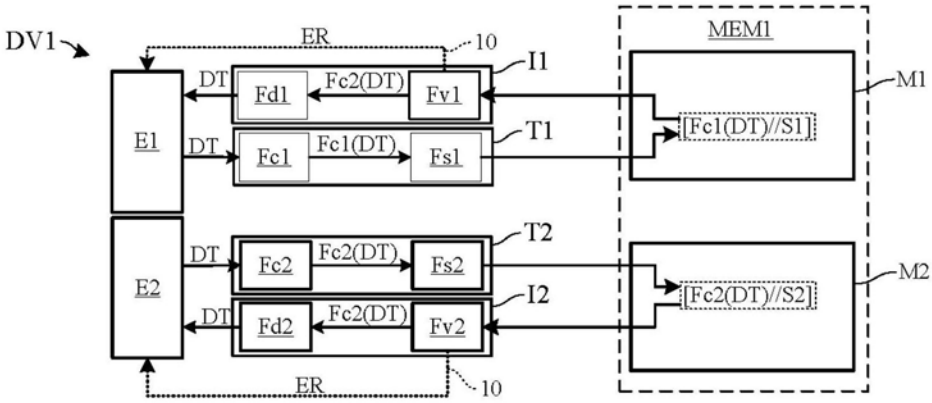

DV1
ER
10
MEM1
I1
M1
E1
DT
Fd1
Fc2(DT)
Fv1
[Fc1(DT)//S1]
T1
DT
Fc1
Fc1(DT)
Fs1
T2
DT
Fc2
Fc2(DT)
Fs2
E2
[Fc2(DT)//S2]
M2
I2
DT
Fd2
Fc2(DT)
Fv2
ER
10

图2

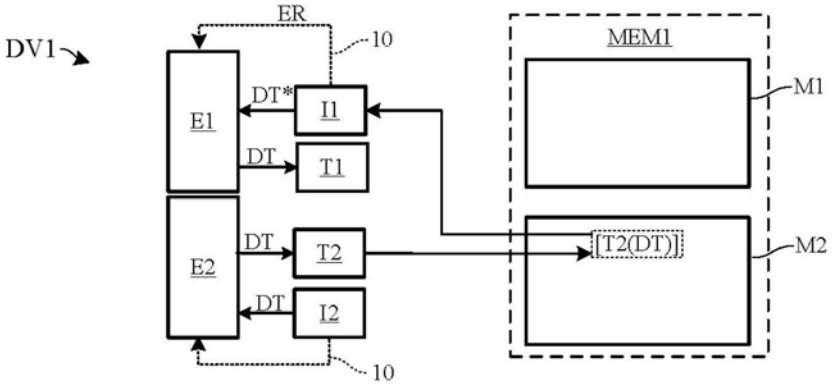

ER
DV1
10
MEM1
E1
DT*
I1
M1
DT
T1
E2
DT
T2
[T2(DT)]
M2
DT
I2
10

图3A

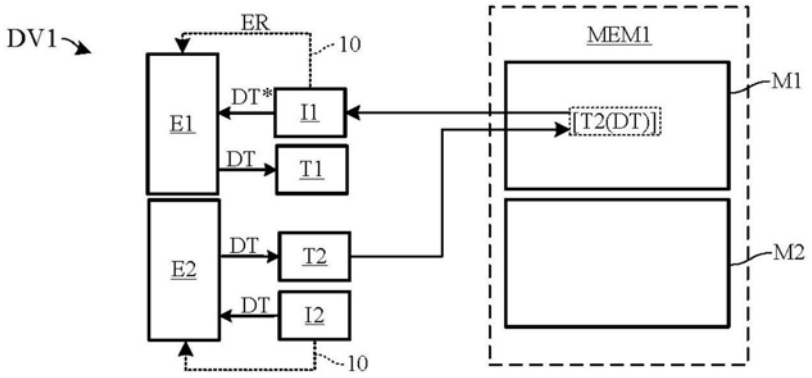

DV1
ER
10
MEM1
M1
E1
DT*
I1
[T2(DT)]
DT
T1
E2
DT
T2
M2
DT
I2
10

图3B

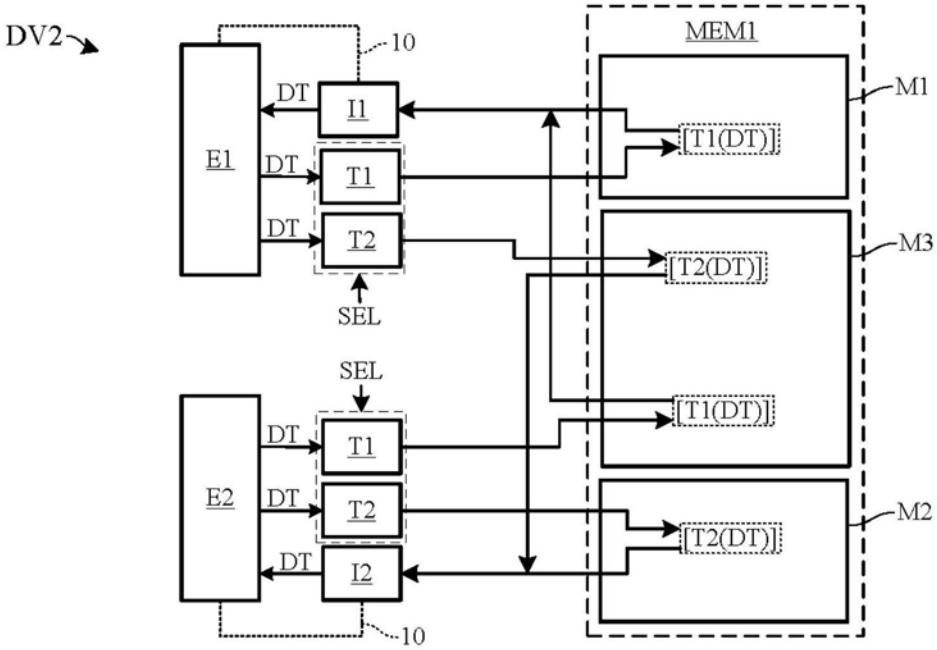

DV2
10
MEM1
M1
DT
I1
E1
DT
T1
[T1(DT)]
DT
T2
[T2(DT)]
M3
SEL
SEL
[T1(DT)]
DT
T1
E2
DT
T2
[T2(DT)]
M2
DT
I2
10

图4

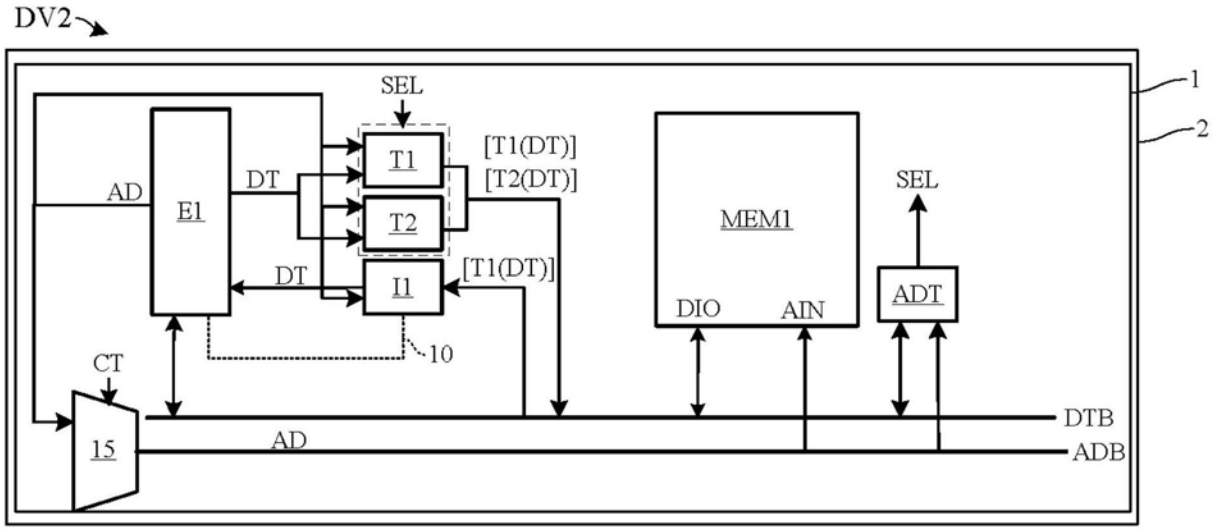

DV2
SEL
1
2
AD
E1
DT
T1
T2
[T1(DT)]
[T2(DT)]
DT
I1
[T1(DT)]
MEM1
SEL
DIO
AIN
ADT
10
CT
15
AD
DTB
ADB

图5

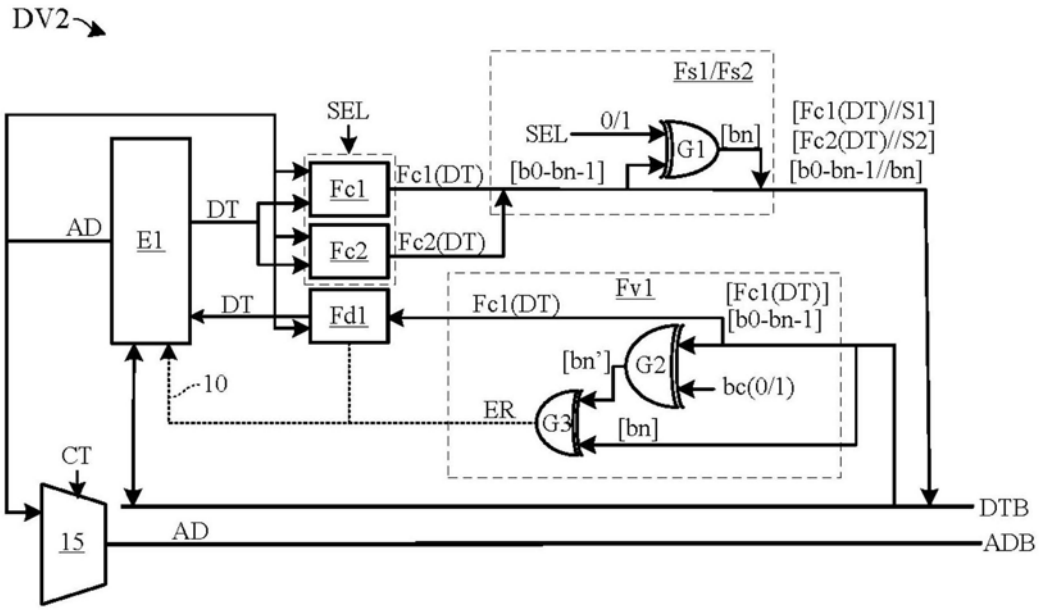

DV2
Fs1/Fs2
SEL
SEL
0/1
G1
[bn]
[Fc1(DT)//S1]
[Fc2(DT)//S2]
[b0-bn-1]
[b0-bn-1//bn]
AD
E1
DT
Fc1
Fc1(DT)
Fc2
Fc2(DT)
DT
Fd1
Fv1
Fc1(DT)
[Fc1(DT)]
[b0-bn-1]
[bn']
G2
bc(0/1)
10
ER
G3
[bn]
CT
15
AD
DTB
ADB

图6

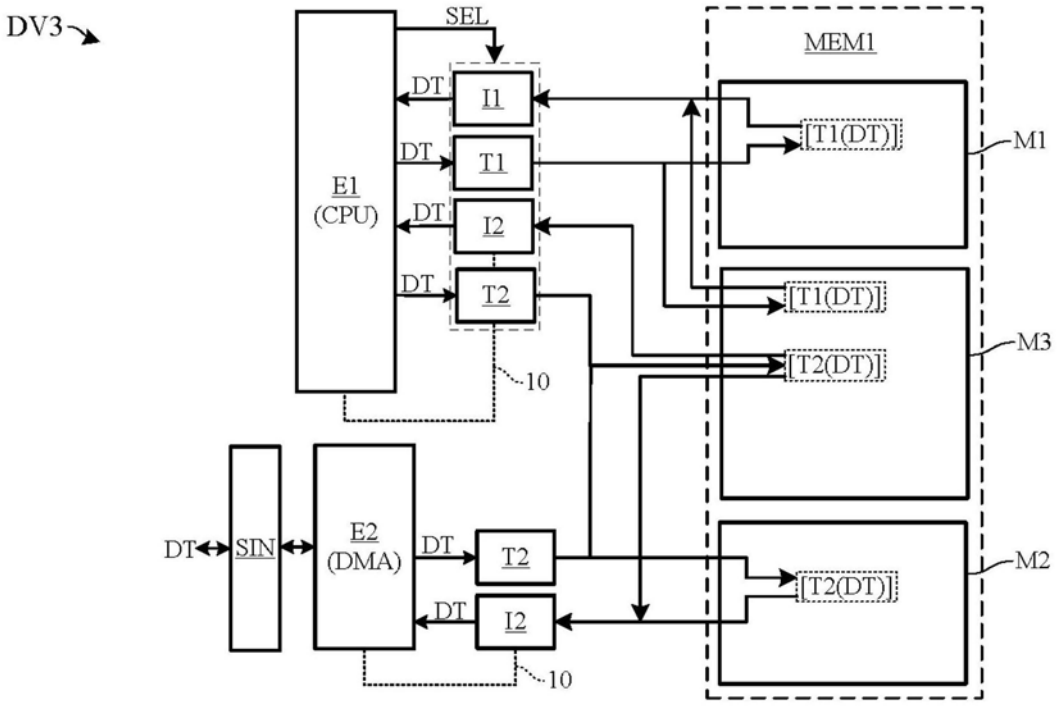

DV3
SEL
MEM1
DT
I1
[T1(DT)]
M1
DT
T1
E1
(CPU)
DT
I2
DT
T2
[T1(DT)]
M3
[T2(DT)]
10
DT
SIN
E2
(DMA)
DT
T2
[T2(DT)]
M2
DT
I2
10

图7

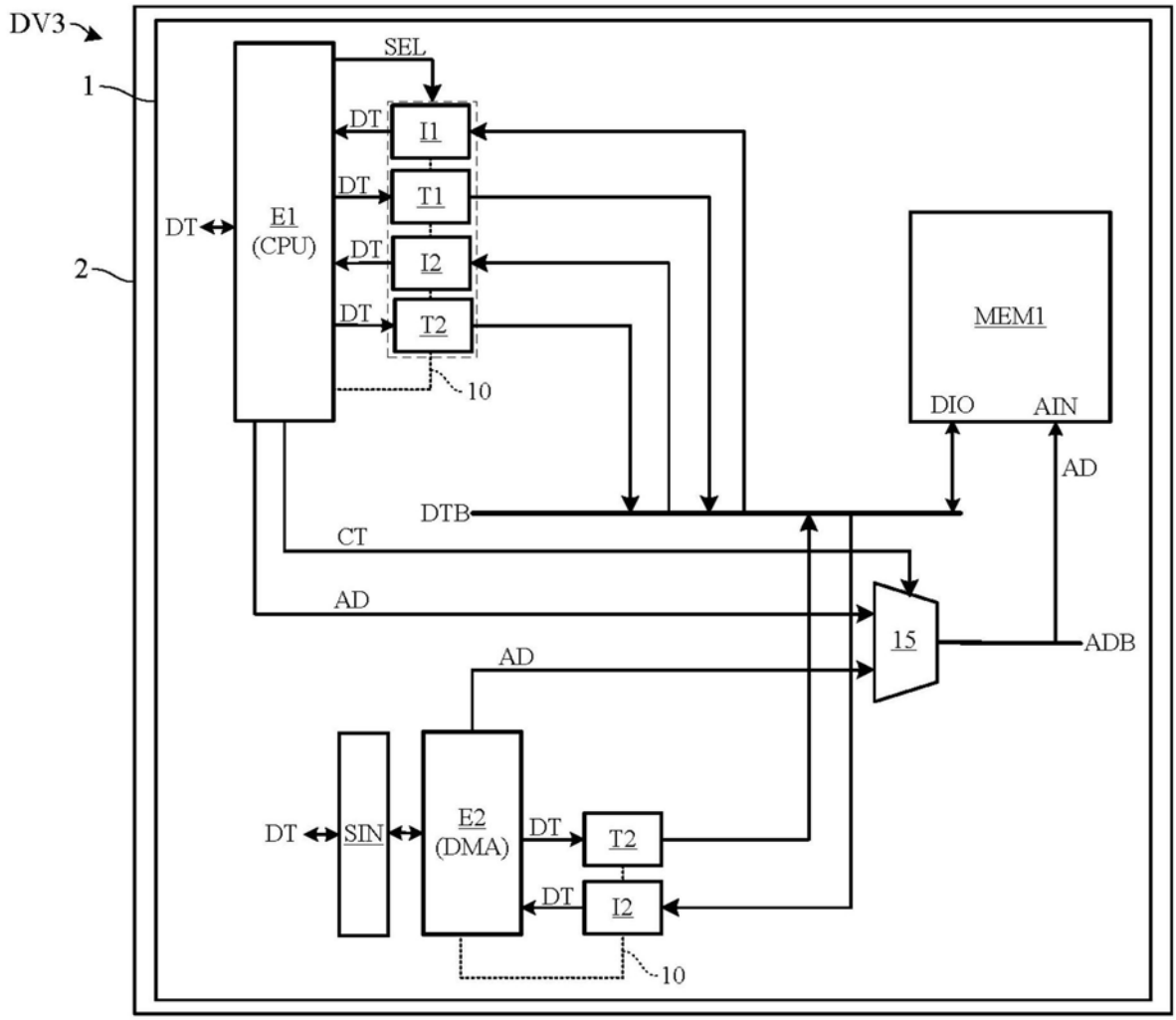

DV3
1
2
SEL
DT
I1
T1
I2
T2
E1
(CPU)
10
MEM1
DIO
AIN
AD
DTB
CT
15
ADB
SIN
E2
(DMA)

图8