



(12) 发明专利

(10) 授权公告号 CN 101159078 B

(45) 授权公告日 2013.03.27

(21) 申请号 200710146477.0

WO 2006033930 A1, 2006.03.30, 全文.

(22) 申请日 2007.07.10

审查员 宋丽

(30) 优先权数据

11/456541 2006.07.10 US

(73) 专利权人 百利国际游戏有限公司

地址 美国内华达州

(72) 发明人 J·W·莫罗 L·麦卡利斯特

M·A·海恩 W·R·怀特

R·A·小卢奇亚诺

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

代理人 刘杰 陈景峻

(51) Int. Cl.

G07F 17/32(2006.01)

(56) 对比文件

WO 2006033986 A1, 2006.03.30, 说明书第 1-99 段、图 1-10.

US 20030054878 A1, 2003.03.20, 全文.

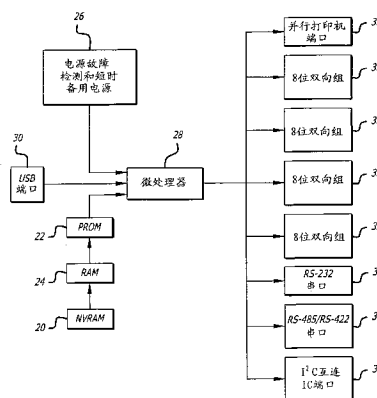
权利要求书 2 页 说明书 32 页 附图 22 页

(54) 发明名称

通用游戏监控单元和系统

(57) 摘要

结合在游戏设备中的嵌入式用户接口, 该游戏设备包括基本游戏的游戏显示和控制基本游戏的游戏处理器。嵌入式用户接口包括玩家跟踪接口和嵌入式处理器。玩家跟踪接口包括显示屏, 允许向用户显示系统游戏, 向用户显示信息, 从用户接收信息。嵌入式处理器使用内部操作系统, 与游戏处理器通信, 实现系统游戏控制, 玩家跟踪信息控制, 非游戏信息控制。



1. 一种结合在游戏设备中的嵌入式用户接口系统, 游戏设备包括基本游戏的游戏显示和用于控制基本游戏的游戏处理器, 其中游戏设备的游戏处理器使基本游戏能够进行, 而无需使用嵌入式用户接口系统, 该嵌入式用户接口系统包括:

包括显示屏的玩家跟踪接口, 其中玩家跟踪接口允许向用户显示系统游戏, 其中玩家跟踪接口允许向用户显示信息, 并且其中玩家跟踪接口允许从用户接收信息; 和

嵌入式处理器, 其中嵌入式处理器使用内部操作系统, 并与游戏处理器通信, 其中嵌入式处理器实现系统游戏和玩家跟踪信息的控制;

其中所述嵌入式用户接口系统的嵌入式用户接口实现系统游戏的控制, 其中将至少一部分系统游戏显示到物理上的显示屏外部; 或

其中所述嵌入式用户接口系统的嵌入式用户接口实现系统游戏的控制, 其中将至少一部分系统游戏显示到物理上的显示屏外部, 且嵌入式用户接口实现系统游戏指示器的控制, 该系统游戏指示器物理上位于显示屏的外部。

2. 权利要求 1 的嵌入式用户接口系统, 其中显示到物理上的嵌入式用户接口外部的至少一部分系统游戏, 或物理上位于显示屏外部的系统游戏指示器包括: 额外的游戏显示。

3. 权利要求 1 的嵌入式用户接口系统, 其中显示到物理上的嵌入式用户接口外部的至少一部分系统游戏, 或物理上位于显示屏外部的系统游戏指示器包括: 额外的显示屏。

4. 权利要求 1 的嵌入式用户接口系统, 其中显示到物理上的嵌入式用户接口外部的至少一部分系统游戏, 或物理上位于显示屏外部的系统游戏指示器包括: 轮盘。

5. 权利要求 1 的嵌入式用户接口系统, 其中显示到物理上的嵌入式用户接口外部的至少一部分系统游戏, 或物理上位于显示屏外部的系统游戏指示器包括: 一个或多个卷轴。

6. 权利要求 1 的嵌入式用户接口系统, 其中显示到物理上的嵌入式用户接口外部的至少一部分系统游戏, 或物理上位于显示屏外部的系统游戏指示器包括: 声音发生机构。

7. 权利要求 1 的嵌入式用户接口系统, 其中显示到物理上的嵌入式用户接口外部的至少一部分系统游戏, 或物理上位于显示屏外部的系统游戏指示器包括: 一个或多个灯。

8. 一种游戏系统, 包括:

连接到网络的玩家跟踪系统服务器, 其中玩家跟踪系统服务器包括应用程序接口;

连接到网络的统计和控制系统服务器, 其中玩家跟踪系统服务器包括应用程序接口, 通过该应用程序接口实现经由玩家跟踪系统服务器的应用程序接口的、与统计和控制系统服务器的通信;

一个或多个连接到网络的游戏设备, 其中每个游戏设备都包括基本游戏的游戏显示和用于控制基本游戏的主处理单元, 并且其中主处理单元实现与统计和控制系统服务器的通信; 和

结合在每个游戏设备中的嵌入式用户接口系统, 其中嵌入式用户接口系统实现与玩家跟踪系统服务器的通信, 每个嵌入式用户接口系统都包括:

包括显示屏的玩家跟踪接口, 其中玩家跟踪接口允许向用户显示系统游戏, 其中玩家跟踪接口允许向用户显示信息, 并且其中玩家跟踪接口允许从用户接收信息; 和

嵌入式处理器, 其中嵌入式处理器使用内部操作系统, 并与游戏处理器通信, 其中嵌入式处理器实现系统游戏和玩家跟踪信息的控制;

其中在游戏设备的主处理单元与结合在所述游戏设备中的嵌入式用户接口系统之间

通信仅通过网络实现,不是通过直接连接来直接实现的,

其中所述嵌入式用户接口系统的嵌入式用户接口实现系统游戏的控制,其中将至少一部分系统游戏显示到物理上的显示屏外部;或

其中所述嵌入式用户接口系统的嵌入式用户接口实现系统游戏的控制,其中将至少一部分系统游戏显示到物理上的显示屏外部,且嵌入式用户接口实现系统游戏指示器的控制,该系统游戏指示器物理上位于显示屏的外部。

通用游戏监控单元和系统

[0001] 版权公告

[0002] 【0001】本专利文献公开的一部分包含受到版权保护的资料。版权所有者不反对本专利文献或专利公开任何一个的复制,如同其出现在专利和商标局文件或记录中一样,但除此之外保留全部任何的版权权利。

[0003] 相关申请的交叉参考

[0004] 【0002】本申请是于 2004 年 9 月 16 日提交的,题为 USERINTERFACE SYSTEM AND METHOD FOR A GAMING MACHINE “用于游戏机的用户接口系统和方法”的,序列号为 10/943,771 的美国专利申请的延续部分,其合并于此作为参考。本申请还是于 2000 年 12 月 22 日提交的,题为 GENERIC DEVICE CONTROLLER UNIT AND METHOD “通用设备控制器单元和方法”的,序列号为 09/746,854 的美国专利申请的延续部分,其合并于此作为参考。本申请要求于 2005 年 9 月 7 日提交的,题为 SYSTEM GAMING APPARATUS AND METHOD“系统游戏设备和方法”的,序列号为 60/714,754 的美国临时专利申请的优先权,其合并于此作为参考。

技术领域

[0005] 【0003】本发明总体上涉及游戏系统,更具体的,涉及一种系统和方法,其将嵌入式用户接口集成到游戏机中。

背景技术

[0006] 【0004】当前一段时间以来,越来越需要能够廉价并容易地将多个任意的设备连接到运行了标准操作系统,如微软®WINDOWS®的计算机。然而,将设备连接到运行了此类复杂操作系统的计算机至少向系统设计者提出了两个令其苦恼的问题。

[0007] 【0005】第一个问题涉及物理互连的问题,就是说,要将一些类型的定制设备插入到计算机中。随着时间流逝,通用“IBM 兼容”计算机功能变得越来越强大,且越来越便宜,但市场被少数程度不同的通用需求,例如打印机、监视器、键盘、鼠标、调制解调器和硬盘所驱动。现代硬件平台被优化以容纳这些部件。

[0008] 【0006】同时,另外的定制设备通常试图或者构建一个为专门连接到该设备而设计的扩展板,或者购买适合于该目的通用板卡。这些选择中最便宜的是通过构建或购买工业标准结构 (ISA) 板卡来增加扩展板。然而,随着时间的继续,正在用越来越少的 ISA 插槽来构建现代中央处理单元 (CPU) 板。许多中央处理单元板现在只具有一个 ISA 插槽。这迫使设计者不得不开发更复杂及昂贵的外围部件接口 (PCI) 板。PCI 总线在系统板部件如 CPU, 与设备如硬盘和视频适配器之间提供了高带宽数据通道。当前所经历的另一个问题是大多数中央处理单元板具有有限数量的串行通讯端口 (com port)。这产生了对能够使用的设备数量的限制。

[0009] 【0007】想要将定制硬件结合到 WINDOWS®环境中的系统设计者所面临的第二个问题是软件开发的问题。通过定义,操作系统负责资源管理。为此,操作系统将连接到系统的

任何及全部硬件都认作属于操作系统。结果,访问该硬件的用户被假定为由操作系统居间协调。

[0010] 【0008】例如作为安全的操作系统环境的 WINDOWS®NT 严格执行该规则。因此,访问硬件的用户被 NT 操作系统居间协调的结果是直接访问硬件的申请的任何努力被操作系统拦截并使其失效。因此,只能通过设备驱动器来实现对硬件的访问,其中该设备驱动器被假定为值得信赖的,这是因为在引导时间将它们加载到操作系统中。

[0011] 【0009】而且,设备驱动器编程是现有的最困难的软件开发范例之一。编程错误往往会使得计算机崩溃,经常没有出现什么出错误的任何指示。调试工具原始且难以使用,并受限于其传送的信息。每个编译加载测试周期要求关闭目标机器并重新启动,这会耗费几分钟。这样,调试过程常常是缓慢的,并阻碍工作。另外,许多设计者都避免实施 WINDOWS®驱动器开发。因此,就希望去除开发者不得不执行此类工作的要求。

[0012] 【0010】在将多个任意的设备连接到计算机时所经历的另一个主要问题是实时设备控制的问题,其中计算机还是运行诸如微软®WINDOWS®的标准操作系统。本质上,真正实时基本上依赖于应用程序。标准 WINDOWS®环境,例如 WINDOWS®98 或 WINDOWS®2000,实际上没有由操作系统对资源管理的真正实时设备控制要求。一旦可以使用安排好的功能,操作系统就仅仅对其进行执行,这常常是在低于 200 毫秒的时间帧内。该时间帧足够小,以使得大多数人将该响应时间视为“实时”,但实际上其不是“真正实时”。

[0013] 【0011】然而,许多外围设备实际上具有真正实时设备控制的要求,其比上述时间间隔更精确。例如,面包块可以以每小时指定英里数在传送带上传送。这些面包块必须在面包块经过喷涂器的准确时间间隔由黄油喷涂器喷涂。如果不能保持这些真正实时设备控制要求,黄油喷涂器就会在面包块经过喷涂器时漏掉它们。不幸的是,以前使标准 WINDOWS®操作系统以真正实时设备控制(例如以分层实时系统或实时核心程序)运行的尝试已经证实是不合需要的昂贵、复杂且不灵活,要求增加更多的串行通讯端口。此外,这些端口是低速的(通常为 9600 波特),且没有着手解决将高速数据(视频)与低速数据(鼠标点击)通信混合的要求。

[0014] 【0012】传统上,游戏机是只为游戏目的而设计的。在这点上,游戏机被构建为只包括游戏功能。然而近来,游乐场所有者已经意识到通过给游戏机增加额外的特色,它们能够将玩家的注意力更长时间的保留在游戏机上。这又导致玩家在游戏机上游戏更长的时间,从而增加游乐场的利润。

[0015] 【0013】已经使用的、将玩家的注意力保持在游戏机上的一种技术是将对与游戏有关的信息的访问提供给玩家。通过将小型电子显示器连接到游戏设备上,可将与游戏有关的信息,以及新闻和广告都发送给玩家。与游戏有关的信息可以包括,例如与体育游戏有关的信息和这些体育赛事的游戏选项。另外,与游戏有关的信息还可以包括诸如赛马和场外游戏之类的信息。通过提供给玩家对信息的访问,信息的范围从放映时间,到餐馆和宾馆特刊,到国际事件,新闻和广告也能够保持玩家的注意力,这样就减小了玩家离开游戏机的需要和/或愿望。

[0016] 【0014】而且,希望提供给玩家对上述信息的交互式访问。此类交互作用允许玩家有更大程度的灵活性来使用上述信息。玩家还可以以更有效的方式来使用与游戏有关的信息。在这点上,更大程度的灵活性和访问可能使玩家停留并在游戏机游戏相当长的时期。

不幸的是,当前使用的,用于显示并访问此类信息的系统部件,例如外部辅助键盘和显示模块,在其所提供的功能和性能上极为有限,因此限制了其保留玩家注意力能力的成效。

[0017] 【0015】如上所述,将与游戏有关的信息和广告发送给玩家的尝试通常要求额外的系统部件单独连接到游戏设备,并脱离于游戏机自身的结构之外。尤其是,因为这些部件固有的性能缺陷,用于访问并显示来自游戏机信息的这些部件在其实用性上极为受限。这种部件通常包括辅助键盘、读卡器和显示设备,例如 2 行 (2-line)LED 显示器。人们希望将这些部件以更统一的方式集成到游戏机自身内,以提供比以前可得到的实质上更强大的功能。

[0018] 【0016】此外,与游戏有关信息的数据收集和各个玩家的游戏习惯对于游戏设备的经营者具有重要的市场价值。通过直邮,以游戏为前提的接触,以更好的顾客为目标的特定事件的组织,及一系列对于在游戏环境中实施营销手段的人来说是公知的技术,这些信息使游戏经营者能够更有效将其营销精力聚焦在顾客身上。

[0019] 【0017】营销专家通常喜欢得到与顾客游戏习惯的集合有关的较宽范围的信息,以及每个顾客的特定习惯。此类营销目的所想要得到的数据类型包括游戏的频率、游戏的持续时间、游戏的钱数、获胜量、及所玩游戏的类型。此类数据的收集最通常的是借助于使用“玩家跟踪系统”来完成。这些计算机系统通常通过使用磁卡(即,玩家跟踪卡)来确认玩家,磁卡是玩家在开始玩之前插入到连接到游戏设备的读卡器中的。在这种系统中,游戏设备通常配备有玩家跟踪部件,其包括磁卡读卡器、显示设备及常用的几个按钮,其向玩家提供了至少一些与玩家跟踪系统通信的能力。

[0020] 【0018】在一些系统中,微处理器(或计算机)位于游戏设备中,游戏设备控制玩家跟踪设备。该微处理器另外的责任是与游戏机自身通信,以监控正进行的游戏。以这个方式,玩家跟踪系统通过读取玩家的磁编码卡来确认正在玩的玩家。玩家跟踪系统还觉察到玩家正在游戏机上玩的游戏活动。玩家跟踪系统计算机通常从游戏机网络收集所有游戏数据,并将其累积到数据库系统的大数据存储库中。在数据库中游戏数据的累积实现了许多类型的分析,以及市场和销售策略的公式化,以改进游戏经营者的商业运作。

[0021] 【0019】一个有助于数据收集的事件是出现几个常用通信协议,其允许在游戏设备与玩家跟踪系统之间准备就绪的通信。许多协议是公知的,并被各种玩家跟踪系统供应商所使用。通常在安装在游戏地点的游戏设备中可以利用这些协议,尤其是传统的国内游乐场,并使得玩家跟踪系统的实施成为相当直接的过程。

[0022] 【0020】然而有时,问题出现在主要不是为传统国内游乐场使用而生产的游戏设备中。这可以是由于许多种原因而发生。这种有潜在问题的系统的一个实例是“视频抽奖”。视频抽奖系统是将在多个物理位置的机器互连的网络。控制计算机系统从网络中的游戏机收集统计数据,并具有各种其它控制和监控功能。通常,在这种系统中,将游戏设备互连到广域网,并使用与在基于主流游乐场的系统中所得到的不同的软件结构和通信机制。通常没有将它们配置为支持与传统玩家跟踪系统的通信。具体的,它们会不支持适当的协议或配置有适当的协议。而且,它们不会被适当的管理机构批准。

[0023] 【0021】在这点上,游戏和制造商的多样性,以及规章批准的成本使得附加的玩家跟踪能力成为非常耗时且昂贵的过程。由于需要在可能是很多的各种制造商之间协调软件与安装,这会更加复杂,这些制造商每一个都可能具有不同的优越性、能力及动机。实际上,

与这种玩家跟踪系统有关的成本会抵消安装玩家跟踪系统的收益。该问题超出了视频抽奖,扩展到许多类型的非传统系统,包括基于宾果 (bingo) 游戏的游戏 (印第安博彩管理法第 2 类 Indian Gaming Regulatory Act Class 2), 欧洲的“街机 (street machine)” (也被称为有奖娱乐游戏 (Amusement With Prizes)), 及各类国际系统。

[0024] 【0022】玩家跟踪系统还被长时间的归类到小型显示器及相当普通的声音性能。另外,还希望得到更多种的输出 (及可能的输入) 设备。而且,晋级的和 / 或基于系统的游戏是新的,并因此在相当小的屏幕上的视频显示受到很大的限制。

[0025] 【0023】因此,本领域技术人员已经认识到设备控制者需要克服以前的,与在硬件、软件和操作系统之间的物理互连;软件开发问题;及晋级的游戏和系统游戏设备控制有关的困难。

发明内容

[0026] 【0024】根据实施例,一种与游戏机有关的嵌入式用户接口系统,其中游戏机包括游戏屏幕和游戏处理器。更具体的,嵌入式用户接口系统包括 web (万维网) 内容显示屏幕、嵌入式处理器和字典扩展 (dictionary extension)。web 内容显示屏幕通过显示屏幕向用户显示信息。嵌入式处理器使用内部操作系统。字典扩展接收输入的文本串,解析文本串以识别导航命令,并从文本串采集统一资源定位符,将从文本串采集的统一资源定位符载入一个变量,并间接将 web 内容显示屏幕导航到变量中的统一资源定位符。以这种方式,借助于提供更丰富的游戏体验,web 内容显示屏幕增加了用户的兴奋性。

[0027] 【0025】根据优选实施例的另一方面,由嵌入式附加用户接口接收的输入数据是 I²C 消息 (或其它串行通信)。优选地,嵌入式处理器通过 I²C 总线 (或其它串行通信总线) 与游戏处理器和 / 或其它相连的设备通信。嵌入式附加用户接口的 web 内容显示屏幕优选地是彩色图形触摸屏显示器。优选地,嵌入式处理器至少是 32 位处理器。此外,优选地对嵌入式附加用户接口的内部操作系统进行定制,以与内部操作系统所依附的特定硬件相匹配。

[0028] 【0026】根据优选实施例的另一方面,嵌入式处理器使用加密技术。在一个优选实施例中,为 web 内容的认证和不可否认性提供一个认证过程。特别的,该认证过程为游戏管理者提供充分的安全性,以允许经营者设计他们自己的内容。

[0029] 【0027】根据优选实施例的另一方面,HTML 是 web 协议,输入的数据在嵌入式附加用户接口中被转换成 web 协议。在另一个优选实施例中,DHTML 是 web 协议,输入的数据在嵌入式附加用户接口中被转换成 web 协议。在再另一个优选实施例中,XML 是 web 协议,输入的数据在嵌入式附加用户接口中被转换成 web 协议。再又一个优选实施例中,MACROMEDIA FLASH 动画技术是 web 协议,输入的数据在嵌入式附加用户接口中被转换成 web 协议。在一个优选实施例中,嵌入式附加用户接口连接到以太网架构。此外在一个优选实施例中,嵌入式附加用户接口通过以太网架构连接到 web 服务器。

[0030] 【0028】根据另一个优选实施例,与游戏机相关联使用的嵌入式用户接口系统还包括如上所述的 web 内容显示屏幕和嵌入式处理器。在该实施例中,字典扩展接收输入的文本串,解析文本串,响应于解析的文本串中的信息发出导航命令,并将显示屏幕导航到由字典扩展选择的统一资源定位符。

[0031] 【0029】如上所述,再根据另一个优选实施例,与游戏机相关联使用的嵌入式用户接口系统包括 web 页显示屏幕和嵌入式处理器。优选地,web 页显示屏幕通过显示屏向用户显示信息。在该实施例中,web 页显示屏幕被分割为多个帧,其中每一个帧都能够显示不同的统一资源定位符。此外,在该实施例中,字典扩展接收输入的文本串,解析文本串,响应于解析的文本串中的信息发出导航命令,并将显示屏幕的一个帧导航到由字典扩展选择的统一资源定位符。

[0032] 【0030】又根据另一个优选实施例,与游戏机相关联使用的嵌入式用户接口系统还包括如上所述的 web 内容显示屏幕和嵌入式处理器。在该实施例中,字典扩展接收输入的文本串,解析文本串,响应于解析的文本串中的信息,发出命令,所述命令在显示屏幕的统一资源定位符上弹出对话框,而不改变显示在显示屏上的统一资源定位符。

[0033] 【0031】一个优选实施例是针对具有游戏显示的游戏机。所述游戏机还包括用户接口,该用户接口具有 web 页显示屏幕、用于控制游戏的处理器、及字典扩展。在该实施例中,字典扩展接收输入的文本串,解析文本串,响应于解析的文本串中的信息发出导航命令,并将显示屏幕导航到由字典扩展选择的统一资源定位符。

[0034] 【0032】根据另一个优选实施例,所要求保护的发明是针对一种方法,用于借助于通过包含在游戏机中的嵌入式用户接口系统提供更丰富的游戏体验,来增加与游戏机有关的用户兴奋性。优选地,嵌入式用户接口系统包括嵌入式处理器、web 页显示屏幕、及字典扩展。该方法优选地包括:接收输入的文本串,解析文本串以识别导航命令,并从文本串采集统一资源定位符,将从文本串采集的统一资源定位符载入一个变量,并间接地将 web 页显示屏幕导航到变量中的统一资源定位符。

[0035] 【0033】在一个实施例中,web 内容受到使用了 DSA(数字签名算法)或 RSA(Rivest-Shamir-Adleman)加密技术的数字签名验证的保护。在这点上,优选地用数字签名验证来保护内容,以便易于识别任何未授权的改变。当然,在其他实施例中也可以使用其它适当的保护技术。

[0036] 【0034】更进一步,一个优选实施例使用消息认证码(MAC),其可以用于验证内容完整性和消息的真实性。消息认证码的产生比使用数字签名验证技术要快,尽管它不太健壮(robust)。在一个优选实施例中,所用的认证技术是 BKEY(电子密钥)设备。BKEY 是电子标识符,其与特定的个人相关联。

[0037] 【0035】通常,在优选实施例中,数据是可认证的和不可否认的,而不是隐藏或者是混乱的(加密的)。不可否认性是确保消息发送者以后不能否认已经发送的消息,且接收者不能否认已经接收到该消息的方法。

[0038] 【0036】根据一个优选实施例,对一个或多个游戏机系统或嵌入式附加用户接口部件(或内容)分配了识别码。使用加密安全程序和绑定组中的部件的识别码,这些部件被一起分组成一个受保护的部件绑定的组。因此,该绑定避免了绑定组内与部件或内容的任何修改或替换有关的内容条目的伪造或否认。

[0039] 【0037】根据优选实施例的另一方面,每个内容条目都必须借助于用散列的消息授权码进行数字签名来进行认证,散列的消息授权码是基于条目自身的,且基于绑定组中的部件和内容的单独的识别码。以相同的方式,试图替换任何嵌入式附加用户接口部件或内容的每一个条目都必须借助于用散列的消息授权码进行数字签名而被认证,散列的消息授

权码是基于条目自身的,且基于绑定组中的部件和内容的单独识别码。

[0040] 【0038】优选地,随机地或伪随机地产生嵌入式附加用户接口部件的识别码。根据验证系统的另一方面,用于证实访问部件绑定的散列的消息授权码密钥用 SHA-1 散列来产生,SHA-1 散列是用绑定组内部件的单独识别码来产生的。另外,在部件绑定中用 SHA-1 散列来保护嵌入式附加用户接口部件,SHA-1 散列是用绑定组内部件和内容的单独识别码来产生的。

[0041] 【0039】通过以下结合附图的详细说明,所要求保护发明的其它特点和优点将会变得明显,附图借助于实例说明了所要求保护发明的特点。

附图说明

[0042] 【0040】图 1 示出了根据本发明的通用设备控制器单元系统的系统结构的部件示意图;

[0043] 【0041】图 2 示出了本发明的通用设备控制器单元系统的操作流程图,对该系统进行配置以连接处理器和单一外围设备;

[0044] 【0042】图 3 示出了本发明的通用设备控制器单元系统的操作流程图,对该系统进行配置以连接处理器和多个外围设备;

[0045] 【0043】图 4 示出了本发明的混合系统的操作流程图,混合系统具有一个对其进行配置以连接处理器和单一外围设备的通用设备控制器单元系统,并具有对其进行配置以连接相同处理器和各种其它多个外围设备的第二通用设备控制器单元系统;

[0046] 【0044】图 5A 示出了从“灯泡”应用到实际灯泡的逻辑数据流程图;

[0047] 【0045】图 5B 示出了图 5A 的最高逻辑传输层的数据流程图,及在第二逻辑传输层中的从应用程序接口到 GDCU 包解码器的逻辑数据流,以及在最高层与第二层之间的物理数据流;及

[0048] 【0046】图 5C 示出了图 5A 的最高逻辑传输层和图 5B 的第二逻辑传输层的数据流程图,其具有在最高层与第二层之间的物理数据流,从 USB 设备驱动器到第三逻辑传输层中的 GDCU USB 接口固件的逻辑数据流,及从 USB 主驱动器到最低物理传输层中的 GDCU USB 接口硬件的物理数据流,以及在层之间的物理数据流。

[0049] 【0047】图 6 示出了根据所要求保护的发明构建的嵌入式附加用户接口的关系图,使用 web 页显示屏幕和嵌入式处理器,嵌入式处理器从游戏监控单元接收数据消息,该数据消息被转换成 web 页内容并映像到 web 页显示屏幕;

[0050] 【0048】图 7 示出了现有技术的游戏系统的关系图,其使用 2×20VF 显示器和 12 数字 (12-digit) 辅助键盘;

[0051] 【0049】图 8 示出了根据所要求保护的发明构建的嵌入式附加用户接口的关系图,使用 web 页显示屏幕和嵌入式处理器,嵌入式处理器通过网络适配器端口从便携式计算机接收加密认证的 web 页内容;

[0052] 【0050】图 9 示出了根据所要求保护的发明构建的嵌入式附加用户接口的关系图,使用 web 页显示屏幕和嵌入式处理器,嵌入式处理器通过以太网架构从后端服务器接收 web 页内容;

[0053] 【0051】图 10 示出了根据所要求保护的发明构建的嵌入式附加用户接口的关系

图,使用 web 页显示屏幕和嵌入式处理器,嵌入式处理器包括标准游戏处理器的功能;

[0054] 【0052】图 11 示出了根据所要求保护的发明构建的嵌入式附加用户接口的对象交互作用图;

[0055] 【0053】图 12 是显示当数据在嵌入式附加用户接口与游戏监控单元之间进行发送时出现的事件顺序的示意图;

[0056] 【0054】图 13 是显示当在 web 页显示屏幕上的虚拟按键被按压时出现的事件顺序的示意图;

[0057] 【0055】图 14 示出了一种通用游戏监控单元,其包含显示屏和扩展的显示设备控制器,其与游戏处理器,一个或多个外围显示设备,及一个或多个后端系统通信;

[0058] 【0056】图 15 示出了一种通用游戏监控单元,其实现了系统游戏指示器的控制,该指示器物理上位于嵌入式用户接口之外;

[0059] 【0057】图 16 是显示基于系统的游戏的游戏过程的逻辑流程图,其使用了由通用游戏监控单元控制的扩展的付费显示;

[0060] 【0058】图 17 示出了一种游戏系统,其通过网络设备将游戏设备连接到提供控制和统计功能的后端计算机系统;

[0061] 【0059】图 18 示出了传统游戏系统,其包括联网到统计和控制系统服务器的游戏设备,其中游戏系统不包括玩家跟踪系统或游戏监控单元 (GMU);

[0062] 【0060】图 19 示出了传统游戏系统,其包括联网到统计和控制系统服务器及玩家跟踪系统服务器的游戏设备,其中游戏设备使用游戏监控单元从游戏主处理单元收集统计及其它信息,并提供玩家跟踪能力;

[0063] 【0061】图 20 示出了新的游戏系统,其包括联网到统计和控制系统服务器及玩家跟踪系统服务器的游戏设备,其中游戏设备包括游戏监控单元和主处理单元 (MPU),每一个游戏监控单元和主处理单元都独立连接到游戏网络,游戏网络又独立连接到统计和控制系统服务器及玩家跟踪系统服务器;和

[0064] 【0062】图 21 示出了经其各自的 API,在游戏监控单元与玩家跟踪系统服务器之间的,及在主处理单元与统计和控制系统服务器之间的逻辑独立连接,以及在统计和控制系统服务器与玩家跟踪系统服务器之间的连接。

具体实施方式

[0065] 【0063】根据本发明构建的通用设备控制器单元系统和方法的优选实施例提供了数据和协议通信接口,其有利于在处理器与设法要进行控制的各种非特定外围设备的任何一个之间的“真正实时”交互作用。现在参照附图,其中在全部附图中相同的参考数字表示相同或相应的部分,更具体的对于图 1-2,显示了根据本发明构建的通用设备控制器单元系统 10 的一个实施例。

[0066] 【0064】简述之,通用设备控制器单元 (GDCU) 系统 10 包括通用“真正实时”外围设备控制器及数据和协议通信接口。设备控制器单元系统 10 是通用的,这是因为系统 10 能够将处理器 40 连接到多个不同外围设备 50 上,而不是设计为只将处理器互连到一个特定外围设备。通用设备控制器单元系统 10 以使用真正实时外围设备控制的方式,用标准非真正实时操作系统和外围设备 50 来连接处理器 40。系统 10 的“真正实时”设备控制器允许

标准非真正实时操作系统实现外围设备 50 的真正实时控制,而不是要求以处理器 40 使用特定的“真正实时”内核程序或特定的“真正实时”分层操作系统。而且,通用设备控制器单元系统 10 连接在处理器 40 与外围设备 50 之间,以使得系统的数据和协议通信接口允许处理器使用单一类型的协议和相关数据,以便通过 GDCU 系统与外围设备通信,其中外围设备可以使用不同类型的协议和相关数据。

[0067] **【0065】**现在更详细说明,并再次参照图 1-2,根据本发明构建的通用设备控制器单元系统 10 的一个优选实施例优选地提供“真正实时”设备控制器,该“真正实时”设备控制器在与运行标准非真正实时软件的处理器 40 连接时产生真正实时外围设备控制。本发明的一个优选实施例提供了一种方法,允许对任何指定应用程序的真正实时的任意定义,从 1 毫秒到 1 纳秒。以这种方式,系统 10 适合于任何指定应用程序的真正实时要求。优选地,系统 10 的设备控制器允许处理器 40(优选地,但不必运行在 Win32 环境中)使用“真正实时”的外围设备控制。通用设备控制器单元系统 10 向标准非真正实时操作系统的资源管理能力提供该真正实时设备控制。有利的,通用设备控制器单元系统 10 产生真正实时外围设备控制,而无需处理器 40 的更高级别的功能。该更高处理器级别功能是极为复杂和昂贵的,其中在以前特定设备控制器单元需要这样的更高处理器级别功能。本发明从而减少这种复杂性和相关的费用。而且,本发明允许使用来自个人计算机、消费电子产品及工业控制企业的可商业获得的、库存的设备,以便增加产品开发和创新的的速度。这样允许了有效而迅速地引入变化。

[0068] **【0066】**使用系统 10 的数据和协议通信接口,来自所有协议及相关数据的常用接口部件被集成到单一“通用”通信流中,这样实现了由现有数据和协议通信流到任何其它类型数据和协议通信流的转换。就“通用”来说,其意味着 GDCU 系统 10 的数据和协议通信接口接受例如来自处理器 40 的 USB 协议及相关的数,并将该协议及数据流转换为 I²C、RS-232、RS-422/RS-485、并行打印机端口、8 位双向端口、通用数字 I/O 端口接口中的任意一个,或任何其它预期的协议和相关数据。相反地,GDCU 系统 10 的数据和协议通信接口接受这些协议和数据流,并将它们转换为 USB 协议及其相关数据以供处理器 40 使用。GDCU 系统 10 的数据和协议通信接口提供这种通用数据和协议接口,用于将处理器 40 与要由系统来控制的任何预期的处理控制设备 50 相连接。因此,根据本发明,借助于使用 GDCU 系统 10,不管其所选择的协议和数据,任何设备 50 都能够与处理器 40 相关联和连接。

[0069] **【0067】**更具体的,现代软件应用程序和设备 50 由许多内部机电模块组成,其全都需要由更高级别的系统来控制并与之通信。GDCU 系统 10 提供了一种控制器,其具有足够的额外输入/输出能力以控制任何设备。GDCU 系统 10 包含定制设计的系统驱动器,该定制设计的系统驱动器允许 GDCU 系统是简单的控制器,该控制器包括许多设备 50 所共有的部件,由处理器 40 执行设备特定的更高的智能功能。GDCU 系统 10 提供输入/输出功能,同时将主处理器 40 用作在常规 WINDOWS[®]操作系统环境中的更高级别的智能。由于其模块性,GDCU 系统 10 易于修改,其模块性允许改变一个级别,而无需必须改变其它级别。例如,通过改变包编码和解码层,能够增加加密和解密,而无需必须改变物理传输层。类似的,也能简单地改变协议及相关的数。

[0070] **【0068】**如上所述,在本发明的优选实施例中,单一 GDCU 系统 10 使用多个协议及其相关的数。这样,GDCU 系统 10 能够与多个设备进行通信。GDCU 系统 10 允许多个协议和

功能合并到一个系统中,同时允许 GDCU 系统 10 通过相容的接口一直与处理器 40 通信。这样,只需要处理器和操作系统使用单一协议及其相关的数据来通过相容的接口与 GDCU 系统 10 进行通信。GDCU 系统 10 包含唯一分配的处理结构,该处理结构允许与任意设备的多任务。

[0071] 【0069】具体的,本发明优选实施例的通用设备控制器系统 10 以相关的支持硬件连接到处理器 40 (有时被称为主控制单元,或游戏处理器)。处理器 40 可以是任何计算机,但优选地是通用单板机,其包括操作系统、软件及相关元件。单板机适于插入到一个装置,用于控制一个过程。优选地操作系统是 WINDOWS[®] NT 嵌入式系统图像,该嵌入式系统图像被配置以支持一个协议,例如 USB。用于处理器 40 的其它可接受的操作系统包括,仅作为实例而不是限制:WINDOWS[®] NT、WINDOWS[®] 98、WINDOWS[®] 2000、WINDOWS[®] CE、LINUX[®]、QNX[®]、DOS、VXWorks[®]、WHISTLER[®]、及嵌入式 WHISTLER[®]。

[0072] 【0070】而且,开发者能够使用一个开发平台,以便在 GDCU 系统 10 上实现定制的解决方案。围绕处理器 40 和通用设备控制单元系统 10 来构建这种开发平台。开发平台提供与这两个设备一起工作所需的硬件和软件,以便设计并实现复杂的嵌入式控制系统。开发平台与多个外围的和插入式部件在一起。这些部件包括,仅作为实例而不是限制:软盘驱动器、IDE CD-ROM 和硬盘驱动器、AGP 视频板卡、键盘、鼠标、PCI 10/100 以太网接口卡、用于 MCU 板的 32 针插入式芯片的典型种类,包括:SRAM、FLASH 存储器、及 M 系统 DiskOnChip[®],但不限于此。

[0073] 【0071】在本发明的一个优选实施例中,通用设备控制器单元 (GDCU) 系统 10 借助于使用工业标准通用串行总线 (USB) 解决了在过去所经历的硬件互连问题。通用串行总线是由主要硬件和软件制造商联盟设计的,以便解决由“IBM 兼容”计算机体系结构的特性和局限所引起的一组问题,因为这与不断扩大的不具有专业技术的人的用户基础相冲突。最终用户通常想要能够简单的插入一个新设备,并使其适当进行工作,而不用必须打开他们的计算机来安装新硬件。设计了通用串行总线协议标准来满足这个需要。

[0074] 【0072】将通用串行总线设计为将其大部分复杂性集中到主机中,以便单个设备可以是简单和廉价的。该总线规范在每个设备插入时,允许其告知 USB 主机它是何类型,应动态地安装什么设备驱动器以便能够使用该设备。出于这些及其它原因,USB 是用于 GDCU 系统 10 的优选实施例物理传输层。然而,本领域技术人员应意识到,尽管一些 USB 特性对于 GDCU 系统 10 的用途是非常适合的,但 USB 协议标准的使用是适当的,而不是必须的。就是说,能够使用任何适合的协议。基本通用设备控制器单元系统 10 独立于任何特定的物理总线。因此,根据本发明,在通用设备控制器单元系统 10 的可选优选实施例中,也能够以相同的有效性来使用 ATM、以太网、CAN、I²C 或多点串行通信。而且,能够配置系统,以驱动任何网络协议,包括,仅作为实例而不是限制:以太网、ATM、WAN、红外线、串行及光纤。

[0075] 【0073】在本发明的优选实施例中,GDCU 系统 10 被设计为帮助工程师利用通用串行总线技术的优点,同时节省时间和费用。提供设备驱动器和 USB 通信协议,以便工程师能够将注意力集中在开发控制系统应用程序上。优选地,GDCU 系统 10 使用 USB 通信协议来与主机(例如处理器 40)对话,并使用以下一个或多个协议(仅作为实例列出而不是限制)来与所连接的设备 50 进行通信:RS-232 和 RS-422/RS-485 串口、LPT 并行打印机端口、及 32 位(即,四个 8 位)双向数字 I/O。还提供了定制设计的设备驱动器和软件库。优选地,

在 GDCU 系统 10 上的数据线是为 I/O 使用这些驱动器而配置的。一旦数据线被配置,就能够写入数据并检查其状态。该应用程序是用子程序调用编写的,指示 GDCU 系统 10 打开或关闭特定的位,并随后检查其它位的状态。

【0076】 【0074】在本发明的一个优选实施例中,处理器 40 运行 WINDOWS®应用程序,该应用程序将信息转换成用于 GDCU 系统 10 的命令。该应用程序使用驱动器通过处理器 40USB 端口来与 GDCU 系统 10 通信。在本发明的 GDCU 系统 10 的一个优选实施例中,数据和协议通信接口是系统 10 的通信部分,其与处理器 40 中的应用程序和不同的外围设备 50 “对话”。当与各种物理设备 50 连接时,GDCU 系统 10 的数据和协议通信接口允许使用“通用”协议及相关数据。GDCU 系统 10 的数据和协议通信接口允许具有变化的输入信号的多个事件由单一通用设备控制器单元系统 10 进行解释,该单一通用设备控制器单元系统 10 用于控制各种外围设备 50。

【0077】 【0075】具体的,根据本发明,图 1 示出了通用设备控制器单元系统 10 的一个优选实施例的系统结构。在该实施例中,GDCU 系统包括具有非易失性存储器 20 的串行 EEPROM、PROM 存储器 22、RAM 外部存储器 24、电源故障检测和短时备用电源电路 26、单板处理器 28、监视计时器(未示出)、软件资源、通用串行总线端口 30、和多个输入/输出功能单元 32。这些多个输入/输出功能单元 32 包括,仅作为实例而不是限制:集成电路互连(I²C)电路、RS-232 串行接口电路、RS-422/RS-485 串行接口电路、32 通用双向 I/O 线路、并行打印机端口(还可以进一步包括光纤、CAN、以太网、及 ATM)。

【0078】 【0076】在提供非易失性存储器的串行 EEPROM 20 中,GDCU 系统 10 为其自己的使用而保留了一些存储器(例如,用于存储设备 ID 代码和序列号),同时用户可以利用这些保留的存储器。在本发明的一个优选实施例中,至少有 512 字节的非易失性串行 EEPROM 存储器 20。要求至少 8K RAM 和 NVRAM 的本发明的一个优选实施例是通过 Dallas Semiconductor 的 32K 的 8NVRAM 得到满足。该存储器由可更换的 10 年锂电池供电。优选地,但不是必须要求的,至少 64K PROM 用于代码和永久数据表。32 针插槽被连线以接受 27C256 或更大的 EPROM 或 FLASH 存储器,该 32 针插槽提供 32 千字节的程序和数据表存储器。另外,优选地至少 32K RAM 用于变量和易失性数据存储器。

【0079】 【0077】电源故障检测电路 26 包括大电解电容器,该大电解电容器缓冲输入的未经调节的 9V 电源(其通过二极管进行隔离),并作为电源故障检测器。该二极管的电源侧被中断电路监控。该结构的实际结果是在电源故障的情况下,在电容器上的电压下降到处理故障点之前的几百毫秒,警告单板处理器电力丧失。这个时间足以在串行 EEPROM 20 中存储至少 128 字节的数据。优选地,短时电源备用电路提供至少足够在电源故障之后 200 毫秒正常操作的备用电源。在电源出现问题的情况下,这提供了对“实时”数据的保护。

【0080】 【0078】优选地,单板处理器是 8051 工业标准 8 位处理器。在一个实施例中,该微控制器是 Philips P80C652。该部件基本上与 8051 相同,除此之外,在标准 UART 之外还包含 I²C 电路。然而,根据本发明,可以使用任何适合的处理器。其它适合的处理器包括 Cypress 和 Microchip 的工业标准 8 位处理器。

【0081】 【0079】当内部程序不再正常运转时,监视计时器复位单板处理器,并包含该监视计时器以提高总体可靠性。监视计时器的操作对于用户是透明的。

【0082】 【0080】对于软件资源,大多数用户应用程序能够用 GDCU 系统 10 的固有特征来实

现,但一些应用程序会要求单板 GDCU 系统处理器 28 的定制编程。在一个优选实施例中, GDCU 系统 10 包含 64Kb 的 PROM22 存储器空间,以及 32Kb 的外部 RAM 24,用于定制应用程序最大程度的灵活性。能够以几种不同的方式来完成定制代码开发,包括对于特定用户规范来约束自定义代码开发,及在编译时间将定制开发者的代码与原始代码相合并。在一个优选实施例中,Philips PDIUSB12 满足了 USB 端口的需求,该 Philips PDIUSB12 是具有并行处理器访问端口的 USB 接口。

[0083] **【0081】**在一个优选实施例的另一方面,RS-232 和 RS422/RS-485 串行接口电路接收器被多路复用到 8051 计算机上相同的 Received Data In(接收数据输入)信号输入。这样,在任一时间只能使用这些串行端口中的一个。由 Maxim 可以获得 MAX202 接口芯片。它由 +5V 供电产生 +/-10 伏,以便应付 RS-232 的电压。MAX 3080 是符合工业标准 75180 管脚引出线,用于 RS-422/485 连接的 Maxim 部件中的一个。这两个接口中的哪一个被连接到 80C652 的 RXD 串行输入线路的选择可由处理器来进行配置。

[0084] **【0082】**在一个优选实施例的再另一方面,将 I²C 端口包含在 80C652 中。优选地,具有用于连接 I²C 端口的四针管座(four-pin header)。优选地,32 通用双向 I/O 线路排列在 8 条线路的四个组中。在每组中的所有 8 条线路在任一时刻或者是输入或者是输出。借助于使用四个 ALS646 锁存收发器和两个 16V8 可编程逻辑器件来对该 32 通用双向 I/O 线路进行寻址,建立 32 个 I/O 信号。该 32 通用双向 I/O 线路可由处理器配置为 8 个一组的输入或输出。这些 I/O 线路的 13 条线路执行双工作状态,作为到并行打印机端口的输出。(从并行打印机端口的四条输入线直接连到 80C652 上的一些另外未使用的针脚。)

[0085] **【0083】**在一个优选实施例的另一方面,并行打印机端口的 8 条数据线共用四个通用组中的一个。还使用了在第二通用组中的四条另外的输出线。这样,当使用并行端口时,两个组专用于输出,16 条线路中的 12 条供并行端口使用。由于 5 条并行端口输入线直接连接到处理器芯片,其它两个通用 I/O 组保留为未定用途的。

[0086] **【0084】**现在参考 GDCU 系统 10 的互连,所有 USB 设备都具有 16 进制 USB 设备供应商 ID 和产品 ID。USB 规范还提供 16 位二进制编码的十进制编码(BCD)的设备 ID,其范围从 0000 到 9999。设备 ID 用于指定系统中的特定 GDCU 板,其中在所述系统中将多个 GDCU 板连接到 USB 总线。

[0087] **【0085】**如上所述,在本发明的一个优选实施例中,GDCU 系统 10 是具有 USB 连接端口的通用 8 位计算机。简而言之,优选地具有足够的 PROM 和 RAM 存储器,这通常对于连接到外部设备的任何合理的连接都是有用的。GDCU 系统 10 具有检测到其将要被关闭,并将关键信息存储到其单板非易失性串行 EEPROM 中的能力。为了控制和其它设备通信,GDCU 系统 10 具有 32 条通用 I/O 线路,一个 I²C 双线连接端口,一个 RS-232 串行端口,及一个并行打印机端口,用于总共 61 个激活的 I/O 信号。在本发明通用设备控制器单元系统 10 的一个优选实施例中使用的硬件运行应用程序指定的固件。固件的主要任务是为了驱动输出设备提供适当的信号。

[0088] **【0086】**此外,使用了一个普遍的协议,而不是为 GDCU 系统 10 所连接的每个单独设备产生唯一的固件。该协议具有适当的命令,用于配置 GDCU 系统 10(数据方向、波特率、驱动器启动等)和用于发射并接收数据。用于 GDCU 系统 10 的固件执行该协议。同样的,为来自主机端的与 GDCU 系统 10 的相对低级的通信实现匹配 WINDOWS[®]或 MACINTOSH[®]的设备

驱动器。用这种方式,与任何特定设备相连所需的复杂信息能够被保留在主机的应用程序层中,其中主机将 GDCU 系统 10 作为桥使用。

[0089] **【0087】**现在参考图 2,显示了通用设备控制器单元系统 10,配置该通用设备控制器单元系统 10 以连接处理器 40,用于控制单一外围设备 50(外围设备具有多个要求处理器控制的任任务)。本发明系统 10 的该实施例使用了功能较弱的处理器(例如 8051 处理器),并被设计为“单个的(al a carte)”或“每个设备(per device)”型的通用设备控制器单元系统 10。在这点上,该实施例是本发明系统 10 的更简单、更廉价、更灵活的实施例。该实施例考虑到一个外围设备 50 的控制,而无需昂贵的电路并且不需要即将到来的任务的功能。

[0090] **【0088】**具体的,图 2 示出了游戏装置(仅作为实例),包括连接到第一 GDCU 系统 60 的处理器 40,以及三个通过集线器 100 连接到处理器 40 的另外的 GDCU 系统 70、80 及 90。第一 GDCU 系统 60 连接并控制退币器装置(hopper device)64,同时这三个另外的 GDCU 系统 70、80 和 90 每一个分别控制按钮 74、灯 84 和投币机构 94。按钮 74 和投币机构 94 是向处理器 40 发送信息的输入设备,用于通过其各自的 GDCU 系统 70 和 90(通过集线器 100)进行数据通信和协议转换。处理器 40 随后处理输入的数据,并以适用于 GDCU 系统 60 和 80 的形式返回数据,GDCU 系统 60 和 80 传输该数据并将其转化为要发送到输出设备的命令,该输出设备具体的是退币器(hopper)64 和灯 84。由于每个设备都具有其自身的通用设备控制器单元系统,因此该结构允许方便地增加、移去、或换出额外的设备。

[0091] **【0089】**现在参考图 3,显示了通用设备控制器单元系统 60,配置该通用设备控制器单元系统 60 以连接到用于多个外围设备 50 的控制的单一处理器 40。根据本发明,系统 60 的该实施例使用功能更强大的处理器(例如 Motorola 68332 处理器),如此充当通用设备控制器单元系统 60 的功能更强大的方案。在这个方面,本发明的系统 60 的这个实施例能够处理更大数量的输入/输出设备要求。

[0092] **【0090】**具体的,图 3 示出了一种装置,包括连接到单一 GDCU 系统 60 的处理器 40。单一 GDCU 系统 60 与退币器装置 64、按钮 74、灯 84 及投币机构 94 相连接,并对它们进行控制,单一 GDCU 系统 60 还具有 I²C 端口。在该实施例中,按钮 74 和投币装置 94 还是输入设备,其向处理器 40 发送信息。然而,在此情况下,两个输入设备都使用单一 GDCU 系统 60,用于与处理器 40 的数据通信和协议转化。再一次,处理器 40 用非真正实时操作系统处理输入的数据,并以对于 GDCU 系统 60 适当的形式返回数据,GDCU 系统 60 随后用 GDCU 系统 10 的真正实时操作系统传输该数据,并将其转化为命令,所述命令被适当发送到灯 84 和退币器 64 输出设备。该结构允许单一通用设备控制器单元系统 60 控制多个设备,但还考虑到增加额外的设备,而无需要求 GDCU 系统 60、退币器装置 64、按钮 74、灯 84、或投币机构 94 的移除和/或修改。

[0093] **【0091】**最后,图 4 示出了本发明的混合系统 10,具有连接到多个通用设备控制器单元系统的处理器 40,其中每一个通用控制单元系统都被配置以控制,如图 2 所示的单一外围设备,及被配置以控制多个外围设备的另一个通用设备控制器单元系统,如图 3 所示。

[0094] **【0092】**具体的,图 4 示出了一种装置,包括处理器 40,该处理器 40 连接到第一功能更强大的 GDCU 系统 60,以及两个另外的通过集线器 100 连接到处理器 40 的功能较弱的 GDCU 系统 110 和 120。如图 3 所示,功能更强大的 GDCU 系统 60 与退币器设备 64、按钮 74、灯 84 和投币机构 94 相连接并控制它们,GDCU 系统 60 还具有一个 I²C 端口。再一次,在该

实施例中,按钮 74 和投币机构 94 仍是向处理器 40 发送信息的输入设备,并使用功能更强大的 GDCU 系统 60 来与处理器进行数据通信和协议转换。处理器 40 处理输入的数据,并以对于 GDCU 系统 60 适当的方式返回数据,GDCU 系统 60 随后传输该数据,并将其转化为命令,所述命令被适当地发送到灯 84 和退币器 64(输出设备)。正如可由图可见的,图 4 的该下面部分与图 3 所示的相同。

[0095] **[0093]**然而,在本发明的这个实施例中,处理器 40 还以适当的形式向 GDCU 系统 110 和 120 返回数据(通过集线器 100),GDCU 系统 110 和 120 随后继续通信并将来自处理器 40 的指令转换为适当发送到额外的灯 114 和电子动画设备(animatronics)124(输出设备)的命令。该结构允许单一功能更强大的通用设备控制器单元系统控制多个设备;允许增加额外的设备,而无需 GDCU 系统 60、退币器设备 64、按钮 74、灯 84 或投币机构 94 的移除和/或修改;并且由于每个设备都具有其自己的通用设备控制器单元系统,因此易于增加、移除、或换出具有其自己的通用设备控制器单元系统的设备(例如另外的灯 114 和电子动画设备 124)。

[0096] **[0094]**以前,对于特定设备接口的设备控制器单元系统,现有数据和协议接口与不同的数据和协议接口的转换(例如从 I²C 到 USB)会在为每个过程控制设备开发所需的不同代码和电路中花费相当大的开发时间、精力和费用。与此相反,将本发明的通用设备控制器单元系统 10 进行配置以充当普通设备的、“通用”数据和协议接口。

[0097] **[0095]**在这点上,根据本发明,GDCU 系统 10 能够代替嵌入式控制系统、多任务操作系统、或任何其它现有技术的嵌入式应用程序。工业上对这种嵌入式控制系统有多个名称。此类名称,包括 MPU(主要或主处理单元),都与单一中央嵌入式控制器有关。单一中央嵌入式控制器是复杂的设备,能够包括 GDCU 系统 10 和用于特定应用程序的处理器 40 的功能。单一嵌入式控制系统能够控制外围设备 50(其由 GDCU 系统 10 进行控制)和应用程序软件(其另外由处理器 40 进行控制)。这些类型的单一中央嵌入式控制器通常是不合需要的,其原因在于它们缺乏互换性和费用问题(由于必须满足 GDCU 系统、处理器和实时操作系统的需要)。GDCU 系统 10 还能够消除为每个活动都具有 ISA 插入卡的需要,及实时分层操作系统或昂贵的“特定任务”实时内核的需要。

[0098] **[0096]**将本发明的各种实施例的逻辑操作实现为(1)运行在计算机系统上的计算机所执行的步骤或程序模块的序列,和/或(2)在计算机系统内互连的机器逻辑电路或电路模块。该实现是依赖于实施本发明的计算机系统的性能要求进行选择的问题。因此,构成在此所述本发明实施例的逻辑操作以不同的方式被称为操作、结构设备、行为或模块。本领域技术人员会认识到这些操作、结构设备、行为和模块可以在系统 10、固件、专用逻辑电路、模拟电路、或其任何组合中进行实现,而不会脱离本发明的精神和范围,正如在所附权利要求中所述的。换句话说,在本发明的通用设备控制器单元系统 10 的优选实施例中,工业标准物理总线及由不同来源提供的各种元件的使用允许了本发明使用分层的软件接口概念。

[0099] **[0097]**现在参考图 5A、5B 和 5C 以说明上述概念,考虑控制电灯泡的行为。在此情况下,简单的 WINDOWS®应用程序使用单一按钮。如图 5A 所示,根据该应用程序,当用鼠标点击按钮时,灯泡被点亮。当然,在 WINDOWS®灯泡应用程序 200 与灯泡 300 之间没有物理连接,但在逻辑上存在连接。通信和控制结构的最顶层被示为从灯泡应用程序 200 到实际

灯泡 300 的逻辑数据流。

[0100] 【0098】逻辑上,这代表预期的实施。用户的应用程序想要能够开启或关闭灯泡,而无需考虑所有系统级的要求,其中实际上需要所有系统级的要求以用于执行灯泡的开关任务。然而,WINDOWS®应用程序并没有与灯泡对话。如图 5B 所示,应用程序实际所做的是与低于它的另外一层软件进行对话。灯泡应用程序 200 向下将物理数据流发送到应用程序接口 (API) 210,应用程序接口 (API) 210 向包解码器 290 发送逻辑数据流,包解码器 290 又连接到实际灯泡 300 上。

[0101] 【0099】总体系统设计者已告知灯泡软件工程师,其灯泡已连接到,例如 GDCU 板的 I/O 端口 2 上的比特位 3,且当该比特位设定为高时,灯泡开启。因此,当是开启灯泡的时间时,所有“灯泡”应用程序要做的是用指令“将 I/O 端口 2 上的比特位 3 设定为高”来调用适当的 API 库程序。

[0102] 【0100】“灯泡”应用程序 200 既不知道也不关心 API 程序 210 将如何安排以接通该比特位。应用程序 200 并不知道 API 程序 210 是将自己执行该行为,通过互联网向在克利夫兰市的灯泡发送 TCP/IP 包,还是向看门人发送电子邮件。应用程序只是向下发送请求,并希望灯泡会真的开启。

[0103] 【0101】同样,API 程序 210 并不知道为何“灯泡”应用程序 200 要将该比特位设定为高。API 程序 210 所真正知道如何做的是将指令“将 I/O 端口 2 上的比特位 3 设定为高”编码到 GDCU 数据包中,该 GDCU 数据包随后在逻辑意义上发送到匹配 GDCU 数据包解码器 290,GDCU 数据包解码器 290 驻留在 GDCU 板的固件中。当 GDCU 包解码器 290 接收到包时,将其分开,并检查该包。包解码器 290 获知该包是用于控制 GDCU 板上的数字 I/O 数据位的包类型之一,并将 I/O 端口 2 上的比特位 3 设定为高,这样使得灯泡点亮。

[0104] 【0102】再一次,这是逻辑连接。如图 5C 所示,在主机中的 API 包编码器程序 210 不能直接与在 GDCU 固件中的包解码器 290 对话。在实际物理数据流通信路径中,物理数据从灯泡应用程序 200 向下流动到应用程序接口 (API) 210,从 API 210 向下流动到 USB 设备驱动器 220,从 USB 设备驱动器 220 向下流动到 USB 主驱动器 230,从 USB 主驱动器 230 流动到 GDCU USB 接口硬件 270,从 GDCU USB 接口硬件 270 向上流动到 GDCU USB 接口固件 280,从 GDCU USB 接口固件 280 向上流动到 GDCU 包解码器固件 290,该 GDCU 包解码器固件 290 最终连接到灯泡 300 自身。这样,向该结构增加了两个额外的级。

[0105] 【0103】上述实际通信路径中的底层是物理传输层。在本发明的一个优选实施例 GDCU 系统 10 中,通信路径中的底层是通用串行总线的硬件。在底层两侧的接口由 USB 接口硬件的制造商提供。如前所述,由于 USB 是更为经常且更为广泛使用的协议,有多个芯片组可用于主机和设备端接口,该主机和设备端接口遵守为物理和电子互连所公布的 USB 规范。

[0106] 【0104】在连接的主机端,有两个由 USB 用户组为 USB 通信所定义的逻辑协议,一个是通用主控制接口 (UHCI),另一个是开放主控制接口 (OHCI)。在任一情况下,制造商都提供 WINDOWS®设备驱动器,该 WINDOWS®设备驱动器允许下一层来与硬件进行通信。

[0107] 【0105】通用设备控制器单元系统 10 具有的计算能力通常比主机所能利用的小得多,且操作系统要求(如果有的话)要简单得多。这种芯片组的各种制造商有简单的接口,该简单的接口允许一个调用程序来确定 USB 的状态,发送数据块,接收数据块等。

[0108] 【0106】返回到主机端,在应用程序级的 GDCU 软件程序与底层级硬件程序之间的转换工作由 GDCU 设备驱动器来执行。该程序是操作系统的有效部分。以可信赖的内核级特权来操作,能够从上层取得 GDCU 包,并将它们向下发送到硬件,以传输到设备。逻辑上,这些 USB 数据块被横向传输到 GDCU 系统 10 固件的 USB 接口级。USB 接口级的工作是与硬件对话,接收包,并将它们向上传递到包解码器。

[0109] 【0107】为了简化,通信路径被描述为(且在图 5A-5C 中示出)单向流。然而实际上,通信是双向的,通信路径箭头双向流动。尽管看起来复杂,上述分层结构在设计中实际上传达了更大的灵活性。每一层都能够被替换,而不会影响在它上下的层。

[0110] 【0108】例如,可以希望加密 GDCU 数据包,以防止其内容在总线上被探查;或者实施数据压缩以改善数据传输时间。这将只要求改变在主机端的 GDCU 应用程序接口级,并在设备端重写包解码器级。其它的一切都保持相同。

[0111] 【0109】作为另外的实例,物理传输层能够从 USB 变为 ATM。这样,底层将不得不改变。在主机端,将不得不提供不同的 GDCU 设备驱动器,这是因为具有该底层的接口将会不同。然而,在主机端的其它一切都保持相同。相应的,在设备端,与通信硬件相连接的 GDCU USB 接口固件将不得不被重写并改变,这是因为将改变该硬件。然而再一次,向上的接口将保持相同。

[0112] 【0110】从系统设计者和应用程序开发者的观点来看,可以忽略最下面三层的功能。所有最下面三层需要知道的是 GDCU 系统 10 的性能,及如何访问它们。对于应用程序开发者来说,这些问题的答案存在于 GDCU 应用程序接口软件的接口规范之中。GDCU 系统 10 的分层结构意味着通过改变在主机上的 GDCU API 软件和在设备上的包解码器层,可以改变或增加功能。可以改变这种功能,而无需关注下面的传输层,并且可以同样的改变传输层,而无需要求对更高层的任何争论。这导致了更短的开发时间和更快的面世时间。

[0113] 【0111】现在参考软件资源,在本发明的一个实施例中,提供了称为 GDCUCONFIG 的程序,所述程序用于改变在 GDCU 板上的设备 ID。使用 GDCUCONFIG,设计者将唯一的设备 ID 分配给各个 GDCU 板。然后,当使用 GDCU 的应用程序调用各种库程序以执行 I/O 请求时,该应用程序为目标 GDCU 板指定设备 ID。

[0114] 【0112】对于 GDCU 系统 10 库软件,在本发明的优选实施例中,以下五个文件用于编译并链接库软件:ESTGDCU.H- 声名和定义;ESTGDCU.LIB- 多线程的;ESTGDCUL.LIB- 多线程的 DLL;ESTGDCUD.LIB- 调试多线程的;ESTGDCUDL.LIB- 调试多线程的 DLL。必须将 ESTGDCU.H 包括在源文件中。选择的库依赖于代码产生的选择。

[0115] 【0113】GDCU 系统 10 库程序通常被显示在以下的表中:

[0116]

程序	功能
GdcuSetPortDirection	设定 4 个 8 位端口之一的方向
GdcuSetPortData	设定数字 I/O 端口之一上的输出数据
GdcuSetAllPortsData	设定在单一调用中的全部 4 个数据端口

GdcuGetAllPortsData	从数字 I/O 端口获得数据
GdcuSelectRS232	将串行 I/O 设定到 RS-232, 并确定波特率
GdcuSelectRS422	将串行 I/O 设定到 RS-422/RS-485, 并确定波特率
GdcuSendSerialData	将数据块放入串行输出缓冲器
GdcuReceiveSerialData	返回任何接收到的串行数据
GdcuNvmRead	从非易失性串行 EEPROM 读取数据
GdcuNvmWrite	向非易失性串行 EEPROM 写入数据
GdcuGetFirmwareVersion	返回 GDCU 板的固件版本
CountOurUsbDevices	返回 GDCU 板的计数, 并列举其符号的名称 (低级程序)
GetGdcuSerialNumbers	返回所有 GDCU 板的序列号和状态 (低级程序)
GdcuWrite	将数据从主机传递到设备 (低级主机到设备数据传递)
GdcuRead	将数据从设备传递到主机 (低级设备到主机数据传递)

[0117] 【0114】以下部分概述了用于 GDCU 系统 10 库程序的使用信息。在本发明的一个优选实施例中, GDCU 系统 10 程序包括以下: CountOurUsbDevices、GdcuGetAllPortsData、GdcuGetFirmwareVersion、GdcuNvmRead、GdcuNvmWrite、GdcuRead、GdcuReceiveSerialData、GdcuSelectRS232、GdcuSelectRS422、GdcuSendSerialData、GdcuSetAllPortsData、GdcuSetPortData、GdcuSetPortDirection、GdcuWrite、及 GetGdcuSerialNumbers。

[0118] 【0115】GDCU 系统 10 CountOurUsbDevices 程序将当前所连接的 GDCU 板的数量返回到系统的 USB 总线。这些设备的每一个都具有复杂的设备名, 其中该设备名由系统来分配。这些名称被填入到 ppDeviceNames 阵列。该阵列在第一次调用 CountOurUsbDevices 程序之前应进行清除。如果任一 ppDeviceNames 指针不是 NULL, 该程序就尝试用 C++ 删除操作符来释放它们。随后的对 CountOurUsbDevices 的调用导致再次执行该列举, 这样将来自任何以前调用的结果进行释放。在对 CountOurUsbDevices 的最后调用之后, 由用户负责释放由那些字符串所代表的存储器。

[0119] 【0116】CountOurUsbDevices 程序在内部由其它库程序进行使用, 用于保持对连接

到系统的 GDCU 板的跟踪。然而,它并不要求正常使用。该程序与 GetGdcuSerialNumbers 程序一起是为了便于列举所有连接到系统的板卡而提供的。

[0120] **[0117]** 在本发明的优选实施例中, GDCU 系统 10GdcuGetAllPortsData 程序从数字 I/O 端口取回数据。在指定了目标 GDCU 板的设备 ID(BDC 值从 0000 到 9999) 之后,对 pbyData 阵列的大小进行初始化(其可以是 1 到 5 的任一值)。pbyData 阵列是要由程序进行填充的 BYTES 的阵列。

[0121] **[0118]** GdcuGetFirmwareVersion 程序取得 GDCU 固件的版本级别。GdcuNvmRead 程序以 16 字节的块读取非易失性串行 EEPROM 存储器。该程序包含指针和字节阵列可利用的大小,其中指针指向要被填充的字节的阵列。

[0122] **[0119]** 此外, GdcuRead 程序从设备向主机传递数据。该程序还包括指针,以及用于缓冲器可利用容量的变量和接收到的字节数,其中指针指向要从 GDCU 系统 10 填充的缓冲器。只有当为 GDCU 固件产生定制代码时才使用 GdcuRead 程序。除非在 GDCU 系统 10 中有等待被传递的信息,否则不调用 GdcuRead 程序。如果在 GDCU 系统 10 没有数据要输出时, GDCU 系统 10 从 USB 主机接收读取请求, GDCU 系统 10 就借助于发送回单一 ASCII 问号字符来做出响应。

[0123] **[0120]** GDCU 系统 10 库包含 GdcuReceiveSerialData 程序,该程序返回任何所接收到的串行数据。该程序还包括指针,以及针对阵列的可利用大小和阵列中接收的字节数的变量,其中指针指向要被填充的字节的阵列。

[0124] **[0121]** GdcuSelectRS232 程序将串行 I/O 设定到 RS-232,并包括一个变量,该变量将波特率确定为 300、600、1200、2400、4800、9600、19200、或 38400 中的一个。任何其它的值都使得电路默认为 2400。尽管 GDCU 系统 10 包含用于 RS-232 和 RS-422/RS-485 通信的电路,但一次只有其中之一能被允许操作。调用该程序指定了随后的 RS-232 通信。

[0125] **[0122]** 在本发明的优选实施例中, GDCU 系统 10 库还包含 GdcuSelectRS422 程序。该程序将串行 I/O 设定到 RS-422/RS-485,并包含一个变量,该变量是针对将波特率确定为 300、600、1200、2400、4800、9600、19200、或 38400 中的一个。再一次,任何其它的值都使得电路默认为 2400。该程序还包含 OutputOn 变量,该变量用于在 TRUE(真)RS-422 模式(默认的)与 FALSE(假)RS-485 模式之间进行指定。如上所述,尽管板卡包含用于 RS-232 和 RS-422/RS-485 通信的电路,但一次只有其中之一能被允许操作。调用该程序指定了随后的 RS-422/RS-485 通信。在 RS-422 与 RS-485 通信之间的区别在于 RS-422 是连续运行的,而 RS-485 输出驱动器只在设备传输时才启动。本发明的一个优选实施例还考虑了该程序包含变量,以支持在传输时驱动器自动转换到 ON(开启)状态。

[0126] **[0123]** GDCU 系统 10 库还包括 GdcuSendSerialData 程序,该程序将数据块放入串行输出缓冲器。该程序包含一个指针,以及针对要被传输的字节数的变量,其中指针指向要被传输的字节的阵列。直到将缓冲器中的所有字节都已经传输到 GDCU 系统 10,否则不返回该程序。

[0127] **[0124]** 另外, GDCU 系统 10 库还包括 GdcuSetAllPortsData 程序,该程序设定在单一单元中的全部 4 个数据端口。该程序包含一个指针,指向要被锁存到 4 个输出端口中的 4 字节的数据。pbyData 变量必须指向至少 4 字节的有效阵列,以避免可能的存储器异常错误。

[0128] **【0125】**接着, GDCU 系统 10 库包括 GdcuSetPortData 程序。该程序包含设定了以下值的变量:GDCU_PORT_0:在连接器 J8 上的端口;GDCU_PORT_1:在连接器 J9 上的端口;GDCU_PORT_2:在连接器 J10 上的端口;及 GDCU_PORT_3:在连接器 J11 上的端口。该程序还包含一个变量,该变量指定了要被锁存到端口中的 8 位数据。应当指出,即使当数据被设定为 GDCU_PORT_INWARD,该数据也能够被锁存到一个端口中。当端口方向随后被转换到 GDCU_PORT_OUTWARD 时,在那时,以前锁存的数据出现在该端口上。

[0129] **【0126】**GDCU 系统 10 库还包含 GdcuSetPortDirection 程序,该程序设定 4 个 8 位端口之一的方向。该程序包含一些与在 GdcuSetPortData 程序中的相同的变量,GdcuSetPortData 程序涉及将 GDCU 端口 0-3 的值分别设定到连接器 J8-J11 上的端口。GdcuSetPortDirection 程序还包含针对以下值的变量。GDCU_PORT_INWARD:读取端口;及 GDCU_PORT_OUTWARD:驱动端口。

[0130] **【0127】**此外,GDCU 系统 10 库还包含 GdcuWrite 程序,所述程序将数据从主机传输到设备。该程序包含一个指针,指向要被发送到 GDCU 的缓冲器;变量,与要被发送到缓冲器的字节数有关(缓冲器容量);及最终发送的字节数(所传输的字节)。只有当为 GDCU 固件产生定制代码时,才使用 GdcuWrite 程序。

[0131] **【0128】**最后, GDCU 系统 10 库还包括 GetGdcuSerialNumbers 程序。该程序包含几个指针,其中的第一个指针是指向 127 个字符指针阵列的指针,该 127 个字符指针阵列包含为总线上的 GDCU 板系统定义的名称。该阵列用 CountOurUsbDevices 程序来填充。GetGdcuSerialNumbers 程序还包含指向 127 个 BOOL 变量的阵列的指针。在返回时,该阵列对于每个有效 DeviceName(设备名)都包含 TRUE(真)(FALSE(假)意味着板卡上有故障。此时某个其它程序具有对该板卡的打开处理(handle),或者在最后几秒中已经存在意外断开,因此系统仍没有确定该板卡不再存在)。该程序还包含指向 127 个 WORD(字)变量的阵列的指针。每个 WORD 变量用当前连接到 USB 总线的每一个有效 GDCU 板的 Device ID(设备 ID)来得到填充。最后,GetGdcuSerialNumbers 程序还包含指向 127 个 DWORD 变量的阵列的指针。每个 DWORD 变量用当前连接到 USB 总线的每一个有效 GDCU 板卡的二进制序列号来得到填充。GetGdcuSerialNumbers 程序被其它库程序在内部使用,以保持对连接到系统的 GDCU 板卡的跟踪。在正常情况下是不需要这些的。该程序与 CountOurUsbDevices 程序一起是为了便于列举所有连接到系统的板卡而提供的。

[0132] **【0129】**总之,通用设备控制器单元系统的优选实施例包括通用“真正实时”外围设备控制器和数据与协议通信接口。该系统是通用的,以便系统能将一个处理器连接到任意数量的各种外围设备,而不是被设计为只将处理器互连到特定外围设备上。该系统在标准非真正实时操作系统与外围设备之间以这样一种方式进行连接,以便使用真正实时外围设备控制,同时允许带宽共享、数据速度差异,并符合各种级别的中断优先权。系统的设备控制器允许标准非真正实时操作系统执行外围设备的真正实时控制。系统在处理器与外围设备之间进行连接,以使得系统的数据和协议通信接口允许处理器使用单一协议及相关数据,以便与使用不同协议及相关数据的外围设备进行通信。

[0133] **【0130】**在本发明的优选实施例中,由于系统的硬件接口允许大量设备以“菊花链”连接到一起,因此设备连接不限于少量的串行通讯端口。本发明消除了依赖于串行通讯端口的需要,串行通讯端口是低速的(通常是 9600 波特),此外它没有解决如本发明的优选实

施例所做的将高速数据（视频）通信与低速数据（鼠标点击）通信相混合的需要。而且，本发明的优选实施例允许使用来自个人计算机、消费电子产品、及工业控制企业的可商业获得的、库存的设备，这增加了产品开发和创新的速度。另外，本发明消除了开发者不得不执行不想进行的 WINDOWS®设备驱动器开发工作的需要。最后，本发明的 GDCU 系统 10 适合于每个特定应用程序的真正实时要求，因此，实际上允许在任意指定应用程序中使用的真正实时的任何定义（例如从 1 毫秒到 1 纳秒）。

[0134] 【0131】尽管相对于游戏系统和游戏装置来说明了本发明的通用设备控制器单元系统，但本领域技术人员可以意识到该通用设备控制器单元系统和方法易于应用到各种其它非游戏技术领域。这些其它非游戏技术领域包括，仅作为实例而不是限制：制造、游乐园、控制系统、安全系统和机械部件生产线。

[0135] 【0132】根据本发明构建的嵌入式附加用户接口的优选实施例是针对在游戏机中集成嵌入式附加用户接口，以通过提供更丰富的游戏体验来增加用户的兴奋性。嵌入式附加用户接口提供增强的玩家满足感和兴奋性，并改善游戏设备的可靠性、交互性、灵活性、安全性、和可计量性。在此用户接口有时被称为“附加的”，这是因为用户接口与游戏屏幕（或其它游戏显示）分离。此外，在此用户接口有时被称为“嵌入式”，这是因为在本发明的一些优选实施例中用户接口包括其自己的处理器。另外，显示屏在此通常被称为 web 内容显示屏幕，也可以（或可选的）是动画显示屏幕、web 页显示屏幕、或多媒体显示屏幕。

[0136] 【0133】现在参考附图，其中在全部附图中，相同的参考数字表示相同或相应的部分，更具体的，对于图 6-10 示出了嵌入式附加用户接口 310 的一个实施例。具体的，图 6 示出了嵌入式附加用户接口 310，包括 web 页显示屏幕 320 和嵌入式处理器 330。将用户接口 310 包含在游戏机 340 中，反过来游戏机 340 又包括游戏屏幕 350（和 / 或非屏幕游戏区域 350，例如旋转卷轴或其它游戏显示），游戏处理器 360 及游戏监控单元 365。嵌入式处理器 330 使用内部操作系统，并优选地通过游戏监控单元 365 与游戏处理器 360 通信。嵌入式处理器 330 读取输入的数据，将数据转换为 web 编辑语言，并将数据映射到 web 页显示屏幕 320。显示屏幕 320 通过显示屏向用户显示 web 页信息，从而通过提供更丰富的游戏体验来增加用户兴奋性。游戏监控单元 365 监控通过用户接口 310 输入的信息。这对于过去用以提供用户信息的传统系统部件 370 提供了显著的改善。和以前的系统部件 370 与游戏监控单元通信相同的方式，用户接口 310 与游戏监控单元 365 进行通信。

[0137] 【0134】如图 7 所示，现有技术游戏设备通常使用单一视频显示屏幕作为游戏机 340 的游戏屏幕 350，同时另外的系统部件 370 连接到游戏机或与其并列放置。显示器可以包括例如 2 行 20 字符 VF（真空荧光）显示器 320。输入设备可以包括 12 数字辅助键盘 371。

[0138] 【0135】然而，再次参考图 6，在本发明的优选实施例中，在现有技术系统中使用的系统部件 370 被嵌入式附加用户接口 310 所代替，以提供 web 页显示屏幕 320 的先进功能。这种功能包括，仅作为实例而不是限制：显示动画、多媒体和其它 web 型内容的能力。嵌入式附加用户接口 310 允许实现通过 web 页显示屏幕 320，以令人兴奋、引人注目的形式，向玩家（或可能的玩家）显示额外信息（例如增强的玩家信息），而不干扰正在游戏屏幕 350 上显示的正常游戏过程。此外，嵌入式附加用户接口 310 不干扰在游戏机 340 中的普通游戏硬件，而是易于集成到游戏机 340 中。

[0139] 【0136】在涉及多个游戏机（或游戏部件）制造商的情况下，可将嵌入式附加用户接口 310 包含在游戏机中（或者是原来的，或者是改装的），而无需要求访问游戏逻辑电路或可能是专有的并且对于来自另一个游戏制造商的游戏机是不可访问的其它游戏系统。这样，在本发明的优选实施例中，除了通常在游戏机中遇到的标准游戏屏幕 350 之外，游戏机 340 还包括嵌入式附加用户接口 310，嵌入式附加用户接口 310 包括 web 页显示屏幕 320，以向玩家显示补充信息。还可以将嵌入式附加用户接口 310 包含在游戏机 340 中，游戏机 340 使用游戏区域（例如旋转卷轴）以代替标准游戏屏幕 350。该补充信息可以包括通常的游戏信息，玩家特定信息，引起玩家兴奋性和兴趣的内容，广告内容（有目标的或无目标的）等。此外，在优选实施例中，嵌入式附加用户接口 310 可以具有与游戏处理器 360 的游戏逻辑相互作用的能力，优选地通过游戏监控单元 365，这样提供进一步的功能，例如有奖游戏、系统游戏、和 / 或将奖品、奖励提供、或礼品从 web 页显示屏幕 320 并入到游戏屏幕 350 中的能力。而且，当没有游戏进行时，web 页显示屏幕 320 可以在“吸引模式”下显示补充信息。游戏处理器 360 也可以使用 web 页显示屏幕 320 来向游乐场雇员显示基于 web 的对话，以利于游戏机的配置和事件调查活动，而无需妨碍游戏屏幕 / 区域 350。

[0140] 【0137】在本发明的优选实施例中，嵌入式附加用户接口 310 被用于使得游乐场服务对于游乐场顾客更便于访问及友好的。在一个优选实施例中，嵌入式附加用户接口 310 被设计为在现有游戏通信系统网络中，与当前使用的游戏平台的硬件结构相连接，这样以减小游乐场的执行成本。到系统网络的标准游戏网络接口，例如 Mastercom 系统，包括传输到辅助键盘和显示器的多点总线方法。Mastercom 系统可由 Bally Manufacturing 获得，并在 Raven 等人的美国专利 No. 5, 429, 361 中加以说明，该专利内容合并于此作为参考。一个这种当前使用总线是 EPI（增强玩家接口），该 EPI 使用工业标准 I²C 总线和信号传输。

[0141] 【0138】在一个优选实施例中，嵌入式附加用户接口 310 被用于替换 / 升级 EPI。优选地，嵌入式附加用户接口 310 以“即插即用”方式替换游戏机的 EP 到。换句话说，旧的 EPI 可以是非插入的，新的嵌入式附加用户接口 310 能够简单插入到游戏机 340 中的游戏监控单元 365 的 I²C 总线中。用户接口 310 使用当前使用的工业标准 I²C 总线，并进行信号传输，而无需任何进一步的修改。嵌入式附加用户接口 310 的嵌入式处理器 330 读取输入的 I²C 数据（内容），将该数据转换为 web 编辑语言（例如，HTML，DHTML，MAXROMEROMEDIA FLASH），并将数据映射到 web 页显示屏幕 320。以这个方式，以前的 I²C 数据消息，该 I²C 数据消息通常被显示在 2 行 20 字符 VF 显示器上，被嵌入式附加用户接口 310 自动变换为引人注意的、动画的（多媒体）web 页型形式。这以极小的改装要求引起了玩家满意度和兴奋性的提高。

[0142] 【0139】在一个优选实施例中，由于嵌入式附加用户接口 310 使用 I²C 硬件和信号传输，这允许用户接口 310 表达并理解 I²C 协议消息集，因此以与游戏机以前与 EPI 通信相同的方式来直接与游戏机 340 的游戏处理器 360（或其它类似的联网设备）通信。因此，在本发明的优选实施例中，能够对以前使用的硬件的功能（例如 EPI）进行替换或强化，从而以嵌入式附加用户接口 310 集成到游戏机 340 中而在实质上得到升级。如此，由这种系统部件 370 的低功能外部硬件（例如，辅助键盘和 2 行 20 字符 VF 显示器）置于游戏处理器 350 上的限制可以被消除。

[0143] 【0140】如上所述，在一个优选实施例中，由嵌入式附加用户接口 310 接收的输入

数据符合 I²C 信号传输协议；然而，在其它优选实施例中，可以使用其它串行通信协议（或电子通信形式）。优选地，嵌入式处理器 330 经游戏监控单元 365 和 / 或其它相连的设备，通过 I²C 总线（或在使用另一个协议的实施例中通过另一个串行通信总线）与游戏处理器 360 通信。嵌入式附加用户接口 310 的 web 页显示屏幕 320 优选是彩色图形触摸屏显示器。优选地，嵌入式处理器 330 至少是 32 位处理器。一个优选地实施例使用 32 位处理器，这是因为加密技术例如 SHA-1（或更好的）和 DSA 算法是写在 32 位系统上，并固有的运行在 32 位系统上。另外，Microsoft®WINDOWS®环境也是 32 位的，在本发明的一些优选实施例中使用该 Microsoft®WINDOWS®环境。此外，可对嵌入式附加用户接口 310 的内部操作系统进行修改或定制，以匹配由游戏机 340 中的设备使用的特定通信总线硬件，内部操作系统与游戏机 340 通信。

[0144] **[0141]** 优选地，嵌入式附加用户接口 310 是除了嵌入式处理器 330 和 web 页显示屏幕 320 之外的嵌入式计算机板，如图 6 所示，嵌入式附加用户接口 310 还包括可移动 COMPACT FLASH®卡 375（或其它存储器存储设备）及网络适配器端口。对嵌入式附加用户接口 310 的内容和特点的更新借助于在物理上换出 COMPACT FLASH®卡 375（或其他存储器存储设备）来完成。这样，为了从嵌入式附加用户接口 310 取回数据，借助于在物理上移除并读取 COMPACT FLASH®卡 375 来访问数据。在其他实施例中，如下所述，借助于通过网络直接或对等（peer-to-peer）下载来提供更新。

[0145] **[0142]** 在一个优选实施例中，被嵌入式附加用户接口 310 的嵌入式处理器 330 所使用的内部操作系统是 4.2 版本的 WINDOWS®CE（或更高的）。优选地，嵌入式附加用户接口 310 构建在由 Kontron Corporation 开发的基于 PXA255 的板卡上。另外，在嵌入式附加用户接口 310 的一个优选实施例中，用于 web 页显示屏幕 320 的浏览器控制是 MICROSOFT®INTERNET EXPLORER®6.0（或更高的），其是标准安装在 WINDOWS®CE 4.2 上的，WINDOWS®CE 4.2 是嵌入式处理器 330 的优选内部操作系统。

[0146] **[0143]** 嵌入式附加用户接口 310 的优选实施例还提供一个机构，用于将系统信息输入到游戏机 340，并从游戏机 340 取回系统信息。如上所述，嵌入式附加用户接口 310 优选地使用工业标准 I²C 硬件和信号传输。I²C 协议具有多个主要功能，即能够参与作为从设备和主设备。嵌入式附加用户接口 310 允许将系统信息（例如被玩家输入到 web 页显示屏幕 300 的信息）从游戏机 340 发送到投币系统网络（slot system network）（或到另一个目的地）。同样的，嵌入式附加用户接口 310 还允许系统信息（例如显示消息）从系统网络（或从另一个源位置）发送到游戏机 340，用于使玩家通过 web 页显示屏幕 320 进行观看。

[0147] **[0144]** 在一个优选实施例中，也可以由用户将信息输入到用户接口 310 的 web 页显示屏幕 320。用户接口 310 的 web 页显示屏幕 320 使用虚拟辅助键盘。此外，用户接口 310 使用辅助键盘字典（keypaddictionary），这样允许用户能够输入比以前使用 12 数字 VF 辅助键盘所可能输入的大得多的信息量。例如，由浏览器显示的触摸屏上的虚拟按键由用户进行按压。这通过用字符串调用其分派（Dispatch）接口来调用辅助键盘（Keypad）对象，该字符串定义了哪一个虚拟键被按压。辅助键盘对象在字典（Dictionary）对象中查找该字符串，该字符串在初始化时用一组按键载入，以在该字符串传递给辅助键盘对象时返回。当辅助键盘对象取回这组 0 或更多按键字符时，辅助键盘对象通过调用被该对象揭示的接口将这组 0 或更多按键字符传递到 GMU。

[0148] 【0145】通常,将网络接口(或等效的系统)用于控制在特定游乐场中的游戏机 340 所用的资金的流动。借助于使用本发明嵌入式附加用户接口 310,仅通过触摸 web 页显示屏幕 320,游戏网络接口能够被指示在玩家账户与游戏设备之间转移资金。另外,可以提供许多其它更复杂的命令和指令。因此,嵌入式附加用户接口 310 直接在游戏设备自身上改善了玩家与游乐场雇员对游戏机 340 的接口。

[0149] 【0146】在本发明的一个优选实施例中,嵌入式附加用户接口 310 的 web 页显示屏幕 320 使得玩家在动画的、多媒体、web 内容型式环境中显示玩家消息。这些消息以前以普通得多的形式被显示在单独的显示设备(例如 2 行 VF 显示设备)上。在一些优选实施例中,web 页显示屏幕 320 的触摸屏按钮被玩家用来在 web 页显示屏幕 320 的窗口之间导航,并允许访问系统功能,例如无现金的撤消、收支平衡请求、系统请求、点数补偿(point redemption)等。在本发明的其他优选实施例中,web 页显示屏幕 320 使用本领域公知的各种其它数据输入技术代替触摸屏数据输入。因此,嵌入式附加用户接口 310 的实现是对游戏机 340 有效的、非常有益的且实质上的升级,这极大增加了通过以前使用 EPI 所可能实现的功能。

[0150] 【0147】在一个优选实施例中,嵌入式处理器 330 将文本数据消息转变为 web 页导航请求,并如下相对于图 11A 和 11B 所示及讨论的,该 web 页导航请求随后显示在 web 页显示屏幕 320 上。脚本语言(scriptlanguage),例如 JAVA SCRIPT 和 VB SCRIPT,也用于一些 web 页。优选地,嵌入式附加用户接口 310 在 web 页显示屏幕 320 上仿真 12 数字辅助键盘和 2×20VF 显示器,所述 web 页显示屏幕 320 具有触摸屏功能。在该实施例中,以前显示在 2×20VF 显示器上的命令与相应的 URL 相匹配,浏览器被用于在 web 页显示屏幕 320 上呈现页面。显示的 web 页包含触摸屏按键,该触摸屏按键有效地对硬件按键进行仿真。

[0151] 【0148】参考图 11A 和 11B,在本发明的一个优选实施例中,字典 URL 方法用于将数据消息转变为 web 页信息。以该方式,在字典数据文件中“查找”数据消息,在此将它们重定向到有吸引力的 URL。嵌入式处理器 330 响应在 I²C 总线上的请求,I²C 总线是计划供现有技术增强玩家接口(EPI)VF 显示使用的。web 页显示屏幕 320 不是象传统 PC 监视器一样的被动显示设备,而是显示屏幕 320 必须以文本型响应来对命令做出响应。这些请求包括初始化请求、状态请求和显示请求。参考图 12,随着每一条要被显示的文本数据消息被传递到嵌入式处理器 330 中,处理器 330 调用 URL 字典来查找一个 URL,以代替文本数据消息。一旦完成该替换,嵌入式处理器 330 就指示 web 页显示屏幕 320 显示(或导航到)适当的 web 页面。

[0152] 【0149】因此,参考图 13,URL 字典部件被用于将一个文本串,映射到一个 URL,其中该文本串从嵌入式处理器 330 发送,并计划供在 2×20VF 显示器上的显示使用,其中 URL 能被用于显示相同消息的在视觉上增强了很多的图形显示。这样,URL 字典部件包含可能被支持的文本消息列表,该文本消息列表能够从嵌入式处理器 330 发送,并包含到一组预期的引人注目的 web 内容的映射,该 web 内容要被显示在 web 页显示屏幕 320 上。在一条消息不在 URL 字典中的情况下,这样的消息映射到替换 2 行模式的页面。

[0153] 【0150】在上述优选实施例中,嵌入式附加用户接口 310 的嵌入式处理器 330 读取输入的 I²C 数据消息,将 I²C 数据消息转变为 web 编辑语言(例如 HTML、DHTML、XML、MACROMEDIA FLASH),并将新转变的 web 页数据消息映射到 web 页显示屏幕 320。另外,嵌入

式附加用户接口 310 还能够读取已经是 web 编辑语言（例如 HTML、DHTML、XML、MACROMEDIA FLASH）的输入数据消息，并将该 web 页数据消息映射到 web 页显示屏幕 320。此外，并且极其有利的是，本发明的一个优选实施例还允许使用该嵌入式附加用户接口 310 的游乐场来设计并使用其自己的内容，从而使得游乐场具有这样的能力，即决定显示在用户接口 310 的 web 页显示屏幕 320 上的 web 页看起来象什么。

[0154] **【0151】**现在参考图 8，在该优选实施例中，可在本地下载内容。具体的，在一个优选实施例中，通过用于下载新内容的物理 USB（或其它连接）来更新内容。在一个优选实施例中，能够通过将独立计算机 378 连接到嵌入式附加用户接口 310 的网络适配器端口来访问在 COMPACTFLASH®卡 375 上的数据。该实施例允许更新操作系统的内容，改变操作系统自身，并从 COMPACT FLASH®卡 375 接收数据。COMPACTFLASH®卡 375 的物理移除也仍是用于在嵌入式附加用户接口 310 上的文件更新和检查的一个选项。

[0155] **【0152】**在一个优选实施例中，便携式计算机用于将数据内容存储并公布到嵌入式附加用户接口 310 上的 COMPACT FLASH®卡 375，以及从嵌入式附加用户接口 310 上的 COMPACT FLASH®卡 375 接收数据。在该实施例中，在嵌入式附加用户接口 310 上的所有内容都如同该所有内容是游戏机一样被认证。

[0156] **【0153】**在另一个优选实施例中，网络适配器端口运行在用户接口 310 的嵌入式计算机板上。该实施例还包括引导装入程序。此外，在该实施例中，便携式计算机 378（上述的）包括，用于将数据上传到嵌入式附加用户接口 310 上的 COMPACT FLASH®卡 375，并从 COMPACTFLASH®卡 375 下载数据的部件。具体的，运行在便携式计算机 378 上的部件是用于将新数据内容移动到嵌入式附加用户接口 310，并用于在嵌入式附加用户接口上的数据内容的确认和验证。优选地，用于更新 COMPACT FLASH®卡 375 的所有数据都通过板卡上的单一内置网络适配器端口移动到嵌入式附加用户接口 310，或从嵌入式附加用户接口 310 移出。

[0157] **【0154】**在本发明的嵌入式附加用户接口 310 出现之前，游戏管理者不愿允许游乐场经营者设计其自己的内容。然而，由于在嵌入式附加用户接口 310 中的嵌入式处理器 330 所用的加密技术，本发明以充分的安全性来为游戏管理者提供认证过程，以允许游乐场经营者设计其自己的内容。具体地，在一个优选实施例中，提供的认证过程确保游乐场经营者设计的 web 内容的授权和不可否认性。优选地，在本发明中，提供的认证过程还确保可审计性和可跟踪性。在本发明的优选实施例中使用了各种加密技术，例如认证和不可否认性（在此以下所述的），来为游戏管理者提供充分的安全性，以允许游乐场经营者设计其自己的内容。

[0158] **【0155】**在一个优选实施例中，该认证过程用于以与认证“签名的程序”相同的方式来认证“签名的内容”（由游乐场所有者产生）。优选地，在该认证过程中使用了 PKI（公共密钥基础结构 public key infrastructure）。PKI 是数字证书、证书管理机构、及验证真实性和有效性的其它注册管理机构的系统。在一个优选实施例中，产生一个“新层（new tier）”或来源于基本 PKI 的第二 PKI，并且利用允许进行有限访问的证书（例如 x509 证书）的能力。因此，该优选实施例允许证书中的属性被用于提供游戏产业中的代码访问和接受的“级别”。

[0159] **【0156】**在一个实施例中，内容受到使用 DSA（数字签名算法）或

RSA (Rivest-Shamir-Adleman) 技术的数字签名验证的保护。在这点上,优选地用数字签名验证来保护内容,以使得任何未授权的改变都是易于识别的。数字签名是手写签名的数字等效物,这是因为数字签名将个人的身份绑定到一条信息上。数字签名方案通常包括签名产生算法和相关的验证算法。数字签名产生算法被用于产生数字签名。数字签名验证算法用于验证数字签名是真实可信的(即它确实是由特定实体产生的)。在另一个实施例中,用其它适合的技术来保护内容。

[0160] **[0157]** 在一个优选实施例中,安全散列函数-1 (Secure HashFunction-1) (SHA-1) 用于从数据内容或固件内容计算 160 位散列值。该 160 位散列值也被称为短缩位串 (abbreviated bit string) 随后被处理,使用被称为数字签名算法 (DSA) 的单向私有签名密钥技术来产生游戏数据的签名。DSA 使用私有密钥 / 公开密钥对的私有密钥,并随机或伪随机地生成整数,以产生数据内容或固件内容的 160 位散列值的 320 位签名。除标识号之外,也将该签名存储在数据库中。在其它优选实施例中,使用了更高级别的安全散列函数,例如 SHA-256 或 SHA-512。

[0161] **[0158]** 在另一个优选实施例中,本发明使用了消息认证码 (MAC)。消息认证码是特定类型的消息摘要,在消息摘要中包括密钥以作为指纹的一部分。然而,正常的摘要包括一个散列(数据),MAC 包括一个散列(密钥+数据)。因此,MAC 是一个数据(或者是明码文本,或者是加密文本)和密钥的函数的位串。将消息认证码附加到数据上,以允许数据认证。此外,MAC 可以用于同时验证数据完整性和消息的真实性。通常,消息认证码 (MAC) 是单向散列函数,该单向散列函数将对称密钥 (symmetric key) 和一些数据都作为输入。对称密钥算法是用于加密的算法,使用了相同的加密密钥来加密及解密消息。

[0162] **[0159]** 能够比使用数字签名验证技术更快的速度生成消息认证码;然而,消息认证码不如数字签名验证技术稳健。因此,当处理速度是至关重要的时候,消息认证码的使用提供了优势,这是因为能够比数字签名验证技术更快地产生并存储消息认证码。

[0163] **[0160]** 在一个优选实施例中,认证技术使用了 BKEY (电子密钥) 设备。BKEY 是附属于特定个体的电子标识符。以这种方式,使用用于认证的 BKEY 做出的内容的任何增加、访问或修改都被链接到该 BKEY 相关联的特定个体。因此,由此为管理者和 / 或要求此类数据或系统认证的其它实体建立了审计追踪。

[0164] **[0161]** 本发明的另一个优选实施例为使用加密安全性的验证使用了“部件绑定”。在部件绑定中,出现配备了不可改变的序列号的一些部件。另外,例如 web 内容或游戏机柜的部件也可以由所有者给与另一个随机标识号。系统中的其它部件,例如主板中的 CMOS 存储器、硬盘驱动器和非易失性 RAM 也被配给了随机标识号。当对这些随机标识号的全部或一些共同在一个分组中一起进行保护时,将该保护的分组称为“绑定”。机器的每个部件都包含其绑定的部分。

[0165] **[0162]** 在一个这种优选实施例中,对内容所做的每个关键日志记录都被以散列消息授权码 (Hashed Message Authorization Code) (HMAC) 来签署,其中散列消息授权码基于记录自身和单独的绑定码。以该方式,由绑定产生的安全性确保了所做的日志记录不能被伪造或否认。

[0166] **[0163]** 在选择了关键游戏和 / 或系统部件,指定单独的标识符,并合并成使用部件“绑定”来保护的受保护分组之后,对这些部件的任何改变随后都会被检测到,授权并载

入日志。例如,在绑定中的内容是用来源于绑定的密钥而数字签名的(SHA-1或更好的)。无论何时对绑定中的一个部件做出记录时,都对该签名进行验证。如果签名是错误的,则记录该安全性的违反和违反者,但通常该记录不被禁止。在其它实施例中,也可以禁止该记录。因此,部件绑定产生了对绑定内任何部件做出改变的个体的加密的审计追踪。

[0167] **[0164]**而且,绑定确保了游戏机系统的关键部件或在此使用的内容,该关键部件被选择为绑定内的部件,不会被以未经授权的方式进行交换或改变。优选地,绑定使用唯一的标识号,该唯一的标识号被指定到游戏平台的极重要的部分,包括,仅作为实例而不是限制:机柜、主板、专用软件、非易失性RAM卡、内容(数据)、和硬盘驱动器。这些标识号以一种加密方式进行合并,以构成保护并实质上封闭了所包括部件的“绑定”,以使得绑定内的部件在不产生审计追踪和要求认证的情况下,不能被修改、移除或替换。因此,对于绑定内的这些部件中要被改变的一个部件,要求适当的认证且产生日志文件记录,文件编制做出改变的个体的行为和身份。在一个优选实施例中,需要BKEY清除或分类的特定级别,以做出特定的改变。

[0168] **[0165]**现在参考图9,在一个优选实施例中,嵌入式附加用户接口310连接到以太网架构380,而不是本地系统网络。当前,游乐场网络不是以太网,而是更小、更简单化的本地系统网。因此,在该以太网架构380实施例中,当前系统网络被工业标准以太网架构代替,例如运行在Cat 3、4、5、6或更高上的10/100 base T(10/100基本T)以太网。因此,将标准10/100 base T以太网卡增加到该实施例中的处理器。优选地,网络使用TCP/IP、HTTP、和XML消息收发或XML变量。然而,任何适合的协议都可以使用。

[0169] **[0166]**此外,在另一个优选实施例中,嵌入式附加用户接口310通过如图9所示的上述以太网架构380连接到全特征的(full featured)、后端、下载配置服务器390。在这种实施例中,全特征的服务器390能够安排内容(游戏或其它的)下载以及从游戏机340上载信息,例如游戏机340当前拥有的何种选项。因此,在优选实施例中,服务器390的基本用途是作为数据下载和数据检索服务器。虽然该服务器390执行上载和下载web内容型信息,但是它通常不连接到万维网。

[0170] **[0167]**必须对该服务器390进行认证(正如游戏机一样),以使得供给嵌入式附加用户接口310的内容对于游戏管理者是可接受的。优选地,以太网架构380和服务器390的使用提供了许多系统益处,包括但不限于可靠性、可维护性、安全性、内容分级、内容测试、部署过程和事件恢复。在一个实施例中,可交付使用的还优选地包括给游乐场所有者和经营者的内容模板和指导方针,以便产生其自己的web内容来部署到web服务器。在一个实施例中,web服务器390认证其内容的方式与嵌入式附加用户接口310允许将内容下载到web页显示屏幕320的方式相同。

[0171] **[0168]**现在参考图10,在本发明的另一个优选实施例中,如图6-9所示的游戏机340的游戏监控单元365以前所执行的功能由嵌入式附加用户接口310的嵌入式处理器330来支持。除非另有说明,将GMU代码从游戏监控单元365转移到嵌入式附加用户接口310的嵌入式处理器330中。因此,这种结构消除了在游戏机340中对游戏监控单元365的需要。这致使在硬件数量和复杂性上相当大地减小,且完成了更加传统型游戏机到更加现代化升级的游戏机的阶段性转变。

[0172] **[0169]**因此,在这种优选实施例中,本发明将注意力集中在包含在游戏机340中

的嵌入式附加用户接口 310 上,游戏机又包括游戏屏幕 350 或其它适当的游戏区(例如旋转卷轴),但不包括游戏监控单元 365。这种嵌入式附加用户接口 310 还包括 web 内容显示屏幕 320 和嵌入式处理器 330。再一次,web 内容显示屏幕 320 通过显示屏向用户显示 web 信息。嵌入式处理器 330 优选地使用内部操作系统。而且,在该实施例中,由于嵌入式处理器 330 代替了游戏机 340 中的游戏监控单元 365,因此嵌入式处理器 330 另外还包括标准游戏监控单元功能(GMU 代码)。如同以前的一样,嵌入式处理器 330 读取输入的数据,如果有必要则将数据转换为 web 协议(web 编辑语言),并将数据映射到 web 内容显示屏幕 320。

[0173] **[0170]** 在一个实施例中,嵌入式附加用户接口 310,在游戏屏幕 350 用于玩游戏时,将消息在 web 页显示屏幕 320 中快速显现(flash)(例如动画、多媒体等)给玩家。可以将这些 web 页型的消息设定为任何实际预期的长度、格式或型式。例如,消息可以显示为“欢迎来到 Harrah's Las Vegas! 你有 1200 个奖金点数。你想要预定宾馆或定餐吗?”重要的是,尽管以前使用的 EPI 只能够在四分之一英寸(0.25")高的单色文本中滚动该消息,与此相反,web 页显示屏幕 20(320)会在 6 英寸(6.0")乘 3 英寸(3.0")的彩色图形显示器上以鲜红色、白色、黑色和绿色的动画形式“快速显现”该消息。另外,在一些实施例中,将玩家身份识别卡插入读卡器和 / 或选择玩家服务按钮会激活额外的玩家服务功能。

[0174] **[0171]** 在使用读卡器(或其它识别技术,例如玩家 ID 码)来识别特定玩家的嵌入式附加用户接口 310 的一个示范性实施例中,web 页显示屏幕 320 向玩家显示引人注目的 web 页型消息,例如,显示“欢迎,史密斯先生!”,以响应对史密斯先生的身份确认。优选地,web 页显示屏幕 320 还具有触摸屏功能,仅作为实例但不是限制,触摸屏功能包括“饮料”、“兑换”、“服务”、“交易”和“返回游戏”、在一个实施例中,当选择每个触摸屏图标按钮时都会在 web 页显示屏幕 320 中为玩家开启一个新的全屏显示。

[0175] **[0172]** 例如,在一个实施例中,当选择了“交易”触摸屏图标按钮时,启动一个新的屏幕,该新的屏幕包括 web 页型消息,“史密斯先生,帐户余额:奖金点数 = 1200,玩家资金 = \$150,可资利用的信用 = \$850,可资利用的游乐场匹配资金 = \$25,”以及“返回游戏”图标按钮。作为另一实例,当在另一个实施例中玩家选择“无现金取款”按钮时,启动一个新的屏幕,其包括触摸屏辅助键盘并快速显现问题,“你需要多少?”以及“输入”、“清除”和“后退”按钮。优选地,该界面还包括“信息”按钮,当选择该“信息”按钮时,在 web 页显示屏幕 320 中启动一个新的屏幕,用于提供对经常提问的问题的答案及其它有用的信息。而且,web 页显示屏幕 320 优选地还包括“历史”按钮,当选择该“历史”按钮时,在 web 页显示屏幕 320 中启动一个新的屏幕,用于提供所有交易和在游戏机 340 上执行的其它活动的历史记录。

[0176] **[0173]** 根据另一个优选实施例,本发明将注意力集中在一种方法上,所述方法通过经由包含在游戏机中的嵌入式附加用户接口提供更丰富的游戏体验,来增加与游戏机有关的用户兴奋性。该方法优选地包括:通过嵌入式附加用户接口 310 中的串行通信总线(例如 I²C)来接收包含增强玩家信息的串行数据消息(例如 I²C 数据消息);将数据消息转换(用嵌入式处理器 330)为 web 编辑语言;以及将数据消息映射到 web 页显示屏幕 320,其中显示屏幕通过显示屏向用户显示 web 页信息。

[0177] **[0174]** 使用本发明的嵌入式附加用户接口 310 可能的优势很多。仅作为实例而不是作为限制,这些可能的优势包括:提供动画的和 / 或多媒体 web 型内容;提供更大且在美

学观点上更吸引人的字体和图标；向玩家提供特定服务，（例如多种语言，帮助残疾人）；促进 web 页显示屏幕 320 的交互式使用；提供为玩家和游乐场雇员定制 web 页显示屏幕 320 的“外观和感觉”的能力；增加玩家兴奋性和参与；及从 EPI 或其它类似的非 web 页型部件的简化可替换性和 / 或可升级性。

[0178] **[0175]**通常，玩家跟踪系统长时间来受到小显示器及相当普通的声音性能的限制。然而，希望将广泛种类输出（及可能的输入）设备并入玩家跟踪系统。另外，基于晋级系统的游戏相对来说较新，且迄今为止还仅限于在相当小的屏幕上的视频显示。产生能够使玩家跟踪系统和 / 或晋级系统游戏使用更大的游戏显示或其它外围设备 440 的设备和 / 或系统是有利的。

[0179] **[0176]**在这点上，对于本发明的另一方面，图 14 示出了通用游戏监控单元 410（该通用游戏监控单元 410 同时包括 GMU 365 和 iView310 的功能，且构建在 GDCU 10 结构之上）的优选实施例，该通用游戏监控单元 410 包含显示屏 420 和 UGMU 处理器 430（例如扩展显示设备控制器），该 UGMU 处理器 430 与游戏机 340 中的游戏处理器 360、一个或多个外围显示设备 440 以及一个或多个后端系统 450 通信。在一些优选实施例中，显示屏 420 是能够向玩家显示并从玩家接收信息的交互式触摸屏。通用游戏监控单元 410 (UGMU) 的一个优选实施例使得基于系统的 iView 310 游戏不限于在 iView 本身内玩游戏，而是允许使用 iView 部件之外的外围设备 440 和系统 450。在一些优选实施例中，没有将显示屏 420 包含在通用游戏监控单元 410 中，而是包含了连接到通用游戏监控单元并与其相关联的外围显示设备 440 中的另一个。

[0180] **[0177]**在一些优选实施例中，通用游戏监控单元 410 使用编程和操作系统，所述操作系统允许使 UGMU 扩充超出传统 GMU 365 的功能，以包括系统游戏特征，仅作为实例而不是作为限制，所述特征包括：(1) 驱动图形显示器（例如视频屏幕），用于向游乐场顾客显示游戏；(2) 通过例如 USB 的接口，驱动机械卷轴 (mechanical reel)（或其它机械游戏显示部件）；或 (3) 驱动其它游戏外围设备 440（例如硬币接收器，钞票接收器，退币器，打印机等）。

[0181] **[0178]**在这点上，通用游戏监控单元 410 能够用于实现几种不同的功能，仅作为实例而不是作为限制地包括：(1) 外部系统游戏控制器（即，控制外部（相对于 UGMU/iView）“付费玩”系统游戏）的输出，(2) 外部系统“扩展的主要付费指示器”控制器（即，控制外部“付费玩”系统奖金的显示），(3) 外部“晋级”系统奖金显示控制器（即，控制外部“晋级”系统奖金的显示），及 (4) 与游戏无关的系统信息显示控制器（即，控制外部与游戏无关的信息的显示）。

[0182] **[0179]**在本发明的一个优选实施例中，通用游戏监控单元 410 包括提供在传统小图形显示屏 420 上的引人注意的彩色动画和玩家跟踪系统。除了产生更引人注目的显示之外，通用游戏监控单元 410 具有需要另外的硬件支持这些高级 (premium) 玩家跟踪系统的使用的潜在销售优势。而且，能够将许多的支付指示器和外围设备 440（例如轮盘、卷轴、灯、按钮、读卡器等）连接到游戏机 340 处的通用游戏监控单元 410 的 UGMU 处理器 430（扩展的显示设备控制器），以用于 UGMU 410 的玩家跟踪系统的显示。另外，如上所述，在一些优选实施例中，显示屏 420 实际上是外围设备 440，该外围设备 440 连接到通用游戏监控单元 410 并与之相关联，而不是包含在 UGMU 自身内。

[0183] **【0180】**因此,能够直接或间接驱动的(使用例如卷轴控制单元(RCU)的外围控制器)任何显示设备或其它外围设备 440 都能够由通用游戏监控单元 410 进行控制。具体地,仅作为实例而不是限制,在 UGMU(并包括 iView)设备 410 之外且可由通用游戏监控单元进行控制的这些外围设备 440 包括,卷轴、轮盘、轻轮盘(light wheel)、灯、计量器、滑动指示器、旋转指针、声音设备等(即,能够由基于玩家跟踪系统的游戏计算机直接控制的或能够连接到此的控制器间接控制的任何设备)。在这点上,外围设备 440 可以是用于显示基于主要或次要系统的游戏输出的任何设备,例如图 15 所示的扩展的主要付费指示器。

[0184] **【0181】**再次参考图 15,一个可由通用游戏监控单元 410 的 UGMU 控制器 430 控制的外围设备 440 是蒙特卡罗(Monte Carlo)型轮盘(或其它类似的轮盘显示)。在这方面,作为非限制性实例,蒙特卡罗型轮盘控制器用 USB、串口、或其它适当的接口连接到通用游戏监控单元 410 的 UGMU 控制器 430。使用该结构,基于系统的蒙特卡罗卷轴旋转游戏能够显示在 UGMU 显示屏 420 上(UGMU 显示屏 420 现在是 iView 屏幕),从而使轮盘设备根据需要旋转,以增强对玩家的吸引力。通过以这种方式使用通用游戏监控单元 410,任何流行的游戏(例如 BallyMonte Carlo 游戏)现在都能够在游乐场中的任何游戏机 340 上获得,而不用管基本游戏和基本游戏的制造商。这显著地增加了在游乐场场地中可获得的游戏主题的多样性和快速扩大,并且打破了竞争游戏制造商所制造的障碍。

[0185] **【0182】**如上所述,例如蒙特卡罗型轮盘(或其它类似的轮盘显示)的外围设备 440 能够被用作(1)基于系统的游戏输出的单独显示、(2)传统“奖金”设备(例如蒙特卡罗)或(3)基于系统的游戏的扩展的主要付费指示器。在这点上,图 16 示出了可以如何把例如图 15 中所示外围设备 440 实现为扩展的主要付费指示器。更具体地,图 16 是示出了基于系统的游戏的游戏过程的逻辑流程图,其中基于系统的游戏使用了由通用游戏监控单元 410 控制的扩展付费显示。

[0186] **【0183】**再次参考图 14,在通用游戏监控单元 410 的另一个优选实施例中,除了用于基于系统的游戏显示之外,设备也能够用于其它显示。除非另有说明,通用游戏监控单元 410 能被用于任何种类的系统信息显示(例如,通过将指针移动到正餐、演出或房间来奖励神秘的奖金)。因此,奖金不必严格的是游戏的一部分,而仅仅是基于系统。

[0187] **【0184】**而且,在优选实施例的另一方面,当游乐场需要一些信息比传统小显示屏获得的效果具有更引人注目的效果时,通用游戏监控单元 410 就显示来自 UGMU 计算机的不同的“消息”(例如消息不限于基于系统的游戏输出)。在一个特定、非限制性实例中,“黄金”玩家参与到游戏中会点亮可由投币主机(slot host)观察到的蜡烛显示。实质上,通用游戏监控单元 410 被变换为浏览器的扩展。在另一个实例中,赢得特定数量的盘点数的玩家会开启灯、响铃或接受某种奖励。

[0188] **【0185】**在一个优选实施例中,通用游戏监控单元 410 能够由与游戏基线板合并的基于 PC 的处理引擎来构建。因此,与来自 PC 工业的产品性能和/或技术进步相关的任何优势都能够由通用游戏监控单元 410 来利用。另外,由于通用游戏监控单元 410 的性能,将游戏基线板连接到标准 PC 处理引擎是有利的(但不是必需的)。

[0189] **【0186】**在一些实施例中,以最小的成本和功能来设计游戏基线板,随后在稍后的日期进行升级以提供更高的特性和益处。优选地,可以定制游戏基线板,以允许与旧的系统和游戏兼容,同时还提供了升级到例如以太网的更新网络标准的硬件接口。

[0190] **【0187】**在一个优选实施例中,通用游戏监控单元 410 使用可从包括 Kontron 和 Axiomtek 的多个制造商获得的 ETX 模块形状因子单板计算机来构造。该模块包括与标准物理尺寸和连接器管脚引出线相组合的 PC 的核心处理功能。因此,设备获得了采购方面的灵活性。ETX 模块还具有非常小的物理覆盖区 (footprints), 这为诸如通用游戏监控单元 410 之类需要装配至由多个制造商生产的投币游戏机的设备提供了优势。因而,小物理尺寸是有利的。而且,低功耗和只需要单一电源电压也是有利的。

[0191] **【0188】**另外, ETX 模块可兼容很宽范围内的操作系统: 专有操作系统、LINUX[®]、WINDOWS[®] CE 和 WINDOWS[®] XP, 所述操作系统在编程和使用中提供了更大的灵活性。接着, 由于 ETX 模块共用公共的覆盖区, 并具有相似的能力运行要求, 极大地简化了从较低成本 ETX 模块到一个具有更高处理能力的 ETX 模块的升级任务。在基本单元起初已经销售到市场中之后, 当增加例如系统游戏的要求更大处理能力的新特性时, 这种升级能是必须的。

[0192] **【0189】**在一个优选实施例中, 能够使用适当的基线板设计来设计通用游戏监控单元 410, 以适合于到用户接口部件的遗留的连接。这种部件可以包括辅助键盘、读卡器和两行显示器。还可以配置通用游戏监控单元 410 以通过 LVDS 有线驱动电路连接到 VGA、XGA 或更好的图形显示器以及触摸屏。以这种方式, 通用游戏监控单元 410 允许用户接口部件距 UGMU 处理设备 420 远程定位。

[0193] **【0190】**在各种可选的优选实施例中, 可以以多种方式来安置通用游戏监控单元 410, 仅作为实例而不是限制地包括: (1) 作为单独的设备; (2) 作为视频、网络和游戏处理设备 (从而将 GMU 处理留给次要的设备) 连接到用户接口设备; (3) 作为具有与用户接口部件的远程连接能力的 GMU 和游戏设备 (即, 到触摸屏和辅助键盘的长线缆)。

[0194] **【0191】**现在参考图 17-21, 优选实施例的另一个方面将注意力集中在一种系统和方法, 用于跟踪顾客的游戏进行, 以及在本身不支持玩家跟踪功能的游戏设备的网络上提供其它营销和游戏功能。这种通用游戏监控单元 410 的实施例允许实现上述功能, 而无需在现有游戏设备认可的软件中进行改变。这对于此类网络的经营者具有重要价值, 并且通过使用系统到系统接口来实现的且与使用传统技术所可能的实现相比, 该系统到系统接口使得能够更加快速和经济地实现这种玩家跟踪系统。

[0195] **【0192】**在这点上, 图 17 示出了一种在本领域公知的游戏系统, 该游戏系统通过网络设备将游戏设备连接到提供控制和统计功能的后端计算机系统。具体地, 图 18 示出了一种传统游戏系统, 该传统游戏系统包括联网到统计和控制系统服务器的游戏设备, 并且其中游戏系统不包括玩家跟踪系统或游戏监控单元 (GMU)。然而, 当如图 18 所示结构成为标准的一段时间之后, 玩家跟踪系统自此已经变得希望得到和相对普通。因此, 需要能够包含玩家跟踪功能的新系统结构。因此, 图 19 示出了传统游戏系统, 该传统游戏系统包括既联网到统计和控制系统服务器又联网到玩家跟踪系统服务器的游戏设备。接着, 在该结构中, 游戏设备使用游戏监控单元从游戏主处理单元收集统计和其它信息, 并使用游戏监控单元来提供玩家跟踪能力。

[0196] **【0193】**在这点上, 通用游戏监控单元 410 的一个优选实施例提供了一种机构, 用于实现具有一大系列可能的营销和游戏增强特性的玩家跟踪系统。通用游戏监控单元 410 能够在任何类型的环境中实现, 其中系统从游戏设备收集玩游戏的数据, 而不要求对游戏设备自身进行任何的软件修改。现在参考图 20, 示出了新的游戏系统, 该新的游戏系统包括

联网到统计和控制系统服务器以及玩家跟踪系统服务器的游戏设备。在该实施例中，游戏设备包括每一个都单独连接到游戏网络的游戏监控单元和游戏处理器（例如主处理单元（MPU）），所述游戏网络又连接到统计和控制系统服务器以及玩家跟踪系统服务器。在该实施例中，在游戏监控单元与主处理单元之间没有直接连接。这种连接会要求对以上提及的游戏设备修改进行软件修改。

[0197] **【0194】**如图 20 所示，在一个优选实施例中，通用游戏监控单元使用与主处理单元（MPU）分离的处理元件。游戏设备中的外围设备与中央计算机系统通信。在该实施例中，通用游戏监控单元 410 与主处理单元共同驻留在游戏设备中，但与该单元在电气和逻辑上独立。所有必要的交换数据都通过中央计算机进行。

[0198] **【0195】**在传统游乐场系统中，例如图 19 所示的系统，游戏监控单元（例如游戏监控单元（GMU））或其它玩家跟踪设备与中央计算机（例如投币数据系统（Slot Data System），由 Bally Gaming and Systems 生产）和游戏处理器通信。在该传统结构中，游戏处理器驻留在游戏机柜内，中央计算机在远程位置（可能的在同一建筑内，但并不是必需的）。当玩家将玩家跟踪卡插入到游戏设备中时，将信号发送到 GMU。GMU 与游戏处理器通信（用一种通信协议，（例如 Slot Accounting System[®]）），以收集记录了机器的活动程度的“计量器”的状态。计量器信息附加到从卡中读取的识别信息并且将至少包含该数据的信息发送到中央系统，在中央系统中，将所述数据记录在数据库中，用于将来的使用。

[0199] **【0196】**接着，在这种传统结构中，当用户在该游戏机玩游戏时，主处理器更新“计量器”以记录玩的数量、获胜数量、及会影响玩的特性表述的其它特定信息中的变化。另外，当达到阈值或检测到其它营销信息时，处理器可以通知中央计算机。当用户停止玩游戏时，用户通常从机器中取走玩家跟踪卡。这引起了信号被发送到游戏监控单元，游戏监控单元再一次确定游戏设备的更新的计量器读数，并向中央系统发送指示玩家的卡已经被取走的消息。另外，添加更新的计量器信息。在一些实施例中，与玩游戏期间玩家的活动有关的信息也同计量器信息一起发送，或可选的，发送到计量器信息。在这点上，通常将游戏进行的信息记录在系统数据库中，并用于一系列营销功能。

[0200] **【0197】**与此相反，在一个实现通用游戏监控单元 410 的优选实施例（例如图 20）的系统结构中，在通用游戏监控单元 410 与游戏处理器之间没有直接的通信，但仍可获得等效的功能。作为代替，通用游戏监控单元 410 使用有权访问实时游戏进行信息的中央系统。优选地，实时收集该实时游戏进行信息（即，大约每个玩游戏期间进行一次——约 5 秒）。该数据收集速度与“高速”网络（例如，以太网等）的收集速度相一致。现在参考图 20，在优选实施例的另一方面中，使用了实时应用程序接口（API），该实时应用程序接口使得玩家跟踪和营销服务器能够访问游戏进行信息。这能够在特定机器通过查询例如统计和控制系统计算机的中央计算机来执行。

[0201] **【0198】**总的说来，通用游戏监控单元 410 的优选实施例将注意力集中在修改没有玩家跟踪的游戏网络，以包括这些能力而无需单个游戏设备中的软件变化。在一个实施例中，如图 17 所示，游戏设备通过本地通信网互连。在该实施例中，依赖于系统的设计者和安装者，网络能够采取许多方法。所述网络或许是串行网络的几种形式之一，例如集线器布置，或更常用的多点轮询系统。可选地，网络可以是使用互联网协议（IP 网络）的以太网链接。该网络允许以直接方式或者通过使用本地集中设备在多重游戏设备与中央计算机之间

实现双向通信。在一些实施例中,本地集中设备可以采取通用计算机形式或由供应者开发的专有设备,以执行通信功能,并且在一些情况下,还执行辅助功能,例如加密,验票,报告产生及其它操作功能。

[0202] 【0199】顾客使用游戏设备来玩许多不同类型的游戏(例如冒险和/或技巧游戏)。它们的游戏和获胜由游戏设备中的游戏处理器来监控,该信息以及时的方式(例如在10秒内)发送到统计和控制系统。操作周期是当前操作中许多系统所特有的。游戏机可以玩许多类型的游戏,仅作为实例而不是限制地包括:具有机械、机电或视频卷轴的传统游乐场型游戏,poker 游戏,具有奖金模式或奖金设备的视频游戏,基于宾果(bingo)的游戏,中央决策游戏,或各种技巧游戏。奖品可以从小游戏延伸到具有高累积奖金的渐进式游戏,所述累积奖金通过在许多地点玩游戏而积累。

[0203] 【0200】再次参考图 20,在一个优选实施例中,通用游戏监控单元可以安装在游戏机内,或连接到游戏机上,或位于游戏机附近。通用游戏监控单元能够实现在游戏网络内游戏设备处的顾客玩游戏记录。玩游戏的数据可用于营销目的,并还能够提供一系列游戏增强功能,仅作为实例而不是限制地包括:广告,免费赠品,奖金游戏,晋级,以及需要玩家的交互式销售点终端体验的任何其它活动。通用游戏监控单元能够控制一系列外围设备,包括许多类型的玩家识别设备之一。这些玩家识别设备包括读取磁条卡的读卡器、智能卡读卡器、生物计量标识符、射频识别设备(RFID),以及机器可读的并提供识别标记的任何其它设备(该识别标记能够单独的或与其它标记相组合而与玩家相关联)。例如,指纹读出装置可能不具有足够的准确性来唯一一个玩家,但与通过读取卡或人工输入而获得的帐户信息相组合,能够用于确保做出唯一的识别。

[0204] 【0201】现在参考图 21,在一个优选实施例中,当顾客插入玩家跟踪卡或以其他方式向通用游戏监控单元来识别其自身时,玩家跟踪事务开始。通用游戏监控单元向玩家跟踪计算机定制一条消息,所述消息包含自身已经识别的玩家标识。通常,该标识是唯一地识别一个帐户的数字(或字符串)形式。系统证实这是已知的帐户,并访问 API,所述 API 允许该帐户能够与统计和控制计算机通信。统计和控制计算机访问其自身的数据存储器,并取得包含激活的通用游戏监控单元的游戏设备的游戏进行计量器。游戏进行信息随后通过 API 返回到玩家跟踪计算机,玩家跟踪计算机记录了玩家玩游戏期间的初始值。通常,随后将一条消息返回到通用游戏监控单元,其中通用游戏监控单元为玩家提供问候消息,并确认对其游戏进行跟踪。

[0205] 【0202】当玩家在游戏设备上玩游戏时,统计/控制计算机更新其内部的游戏进行信息。当玩家完成其玩游戏的过程时,他指示系统其正在结束其游戏。通常,该操作通过取走其身份标识卡(或等价的标记)来实现,尽管其它方法也是可能的,例如通过按压一个按钮。当该操作发生时,通用游戏监控单元向玩家跟踪计算机发送提供玩家的标识的消息。玩家跟踪计算机再次经 API 访问统计/控制计算机,并提供玩家游戏期间的结束计量器(或其它等价的数据)。玩家跟踪计算机随后在玩家数据记录中记录玩家的全部游戏,所述玩家数据记录可以使玩家有资格拥有营销(或其它)回报。随后游戏系统经营者也可获得该数据,以用于一系列营销计划。

[0206] 【0203】尽管以专用于计算机结构特征、方法动作的语言并借助于计算机可读取介质来说明了本发明,但可以理解在所附权利要求中定义的本发明不必限于所描述的特定结

构、动作或介质。因此,特定结构特征、动作和介质是作为实现本发明的示范性实施例而公开的。

[0207] 【0204】而且,上述各种实施例是仅作为说明而提供的,不应解释为对本发明的限制。本领域技术人员易于认识到对本发明可以做出各种修改和变化,而不必遵循在此所示和所描述的实例实施例和应用,并且这些修改和变化并未脱离在以下的权利要求中进行阐明的本发明的真正精神和范围。

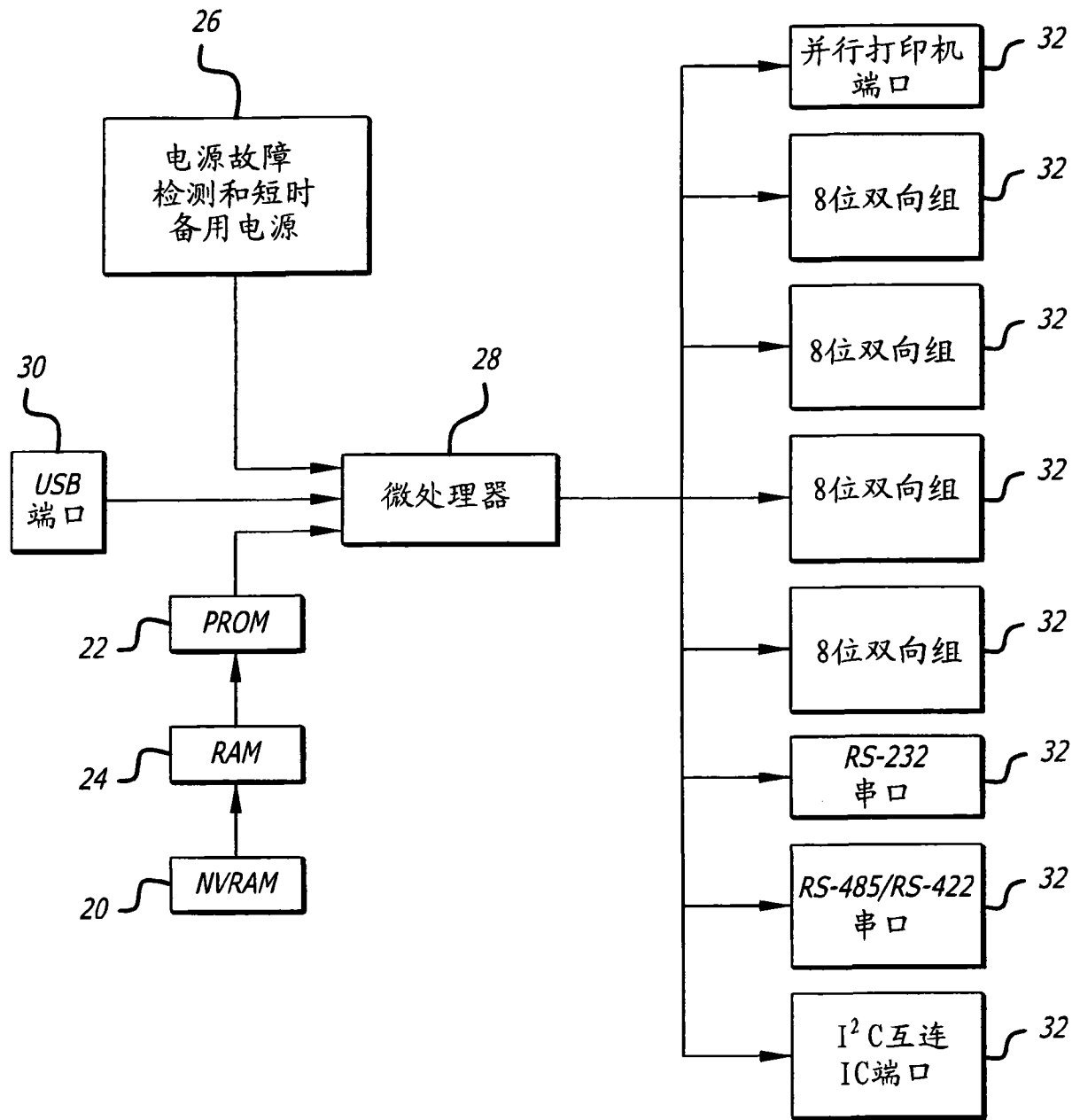


图 1

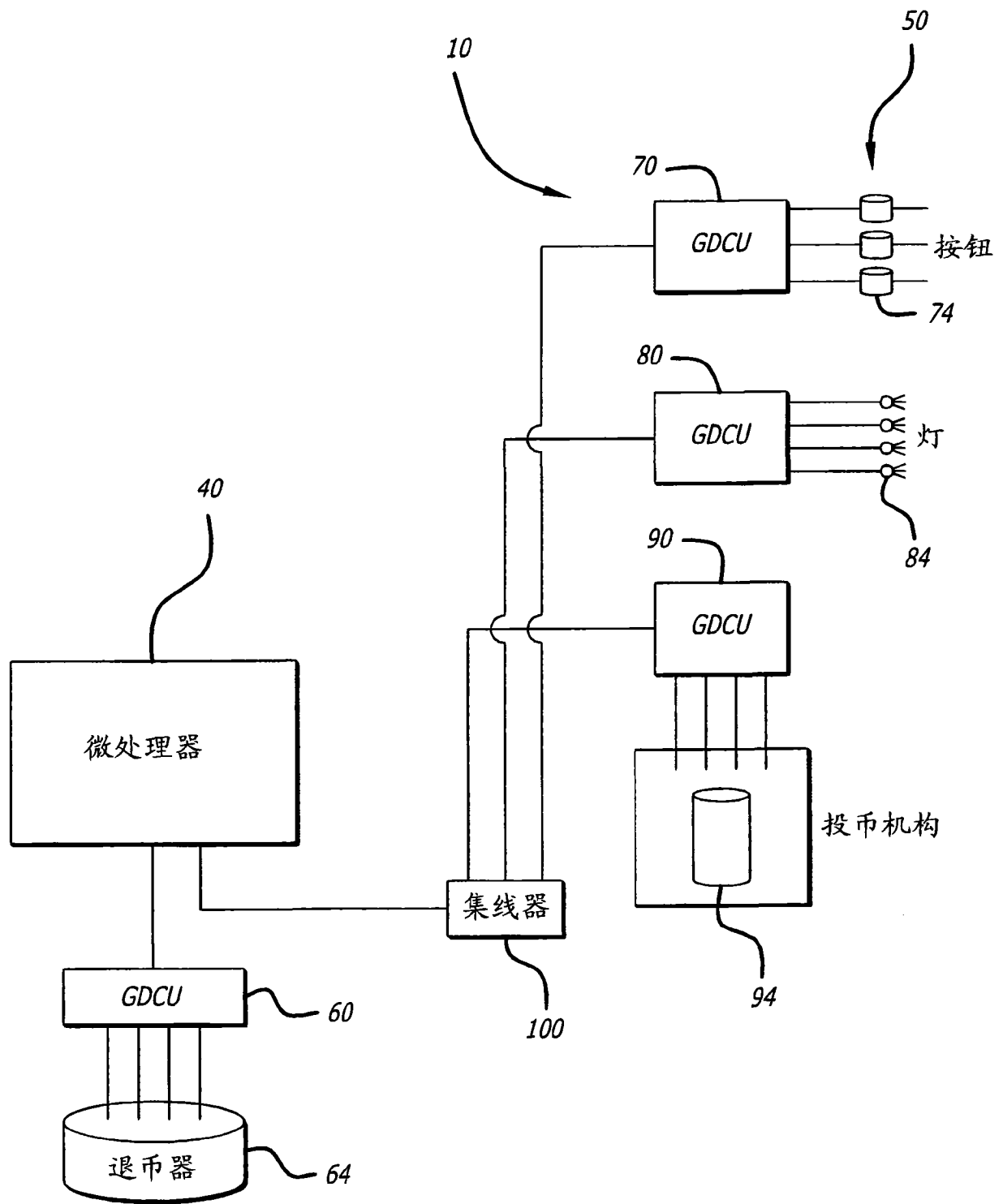


图 2

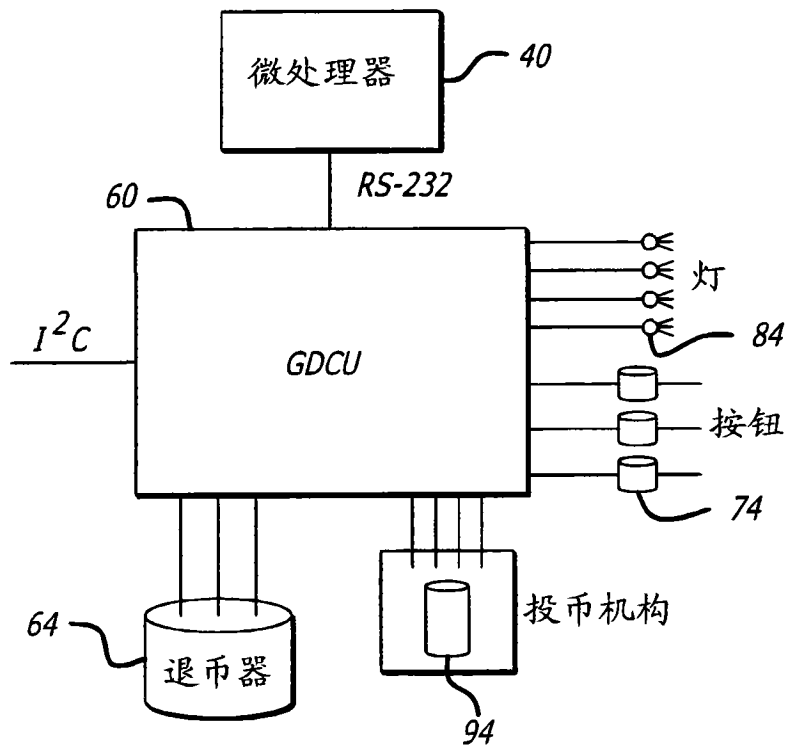


图 3

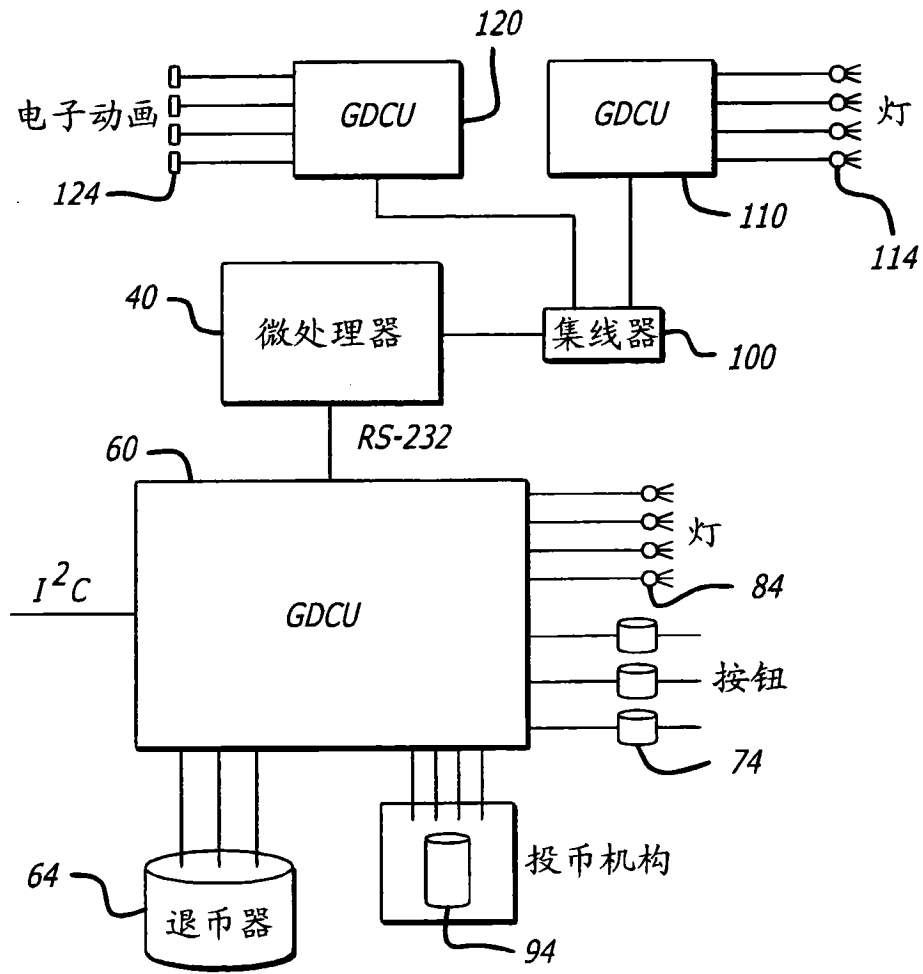


图 4

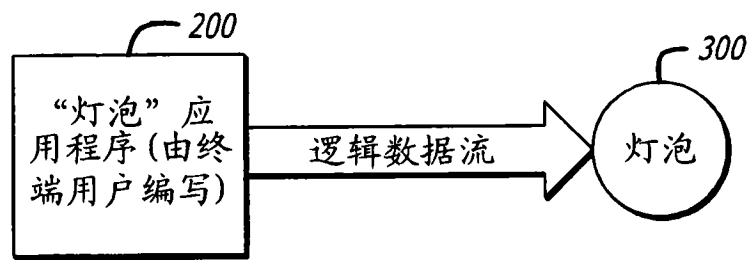


图 5A

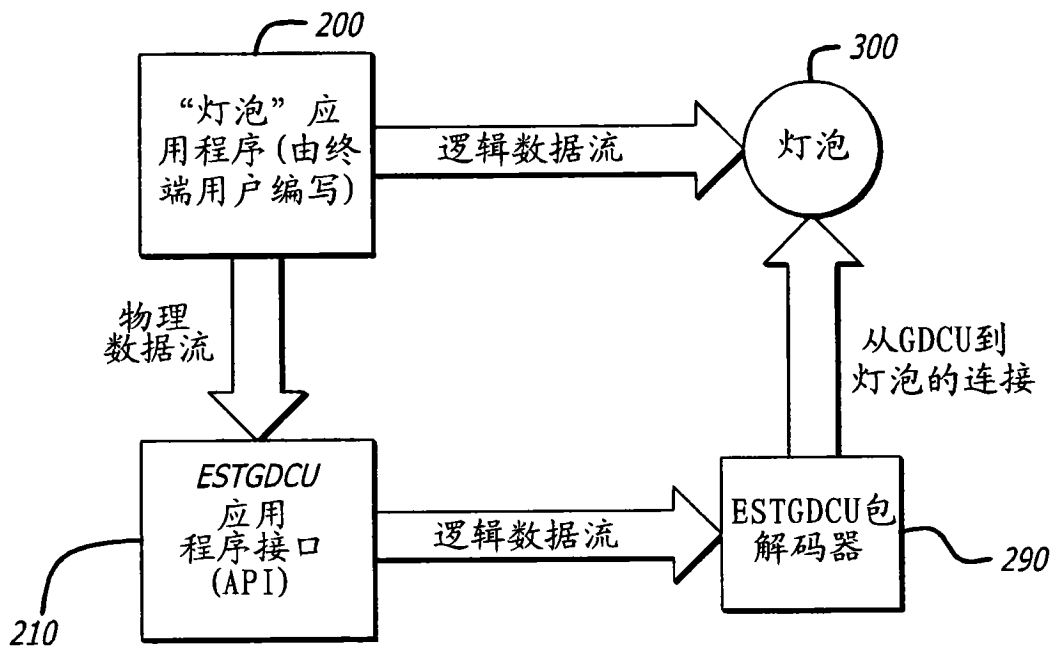


图 5B

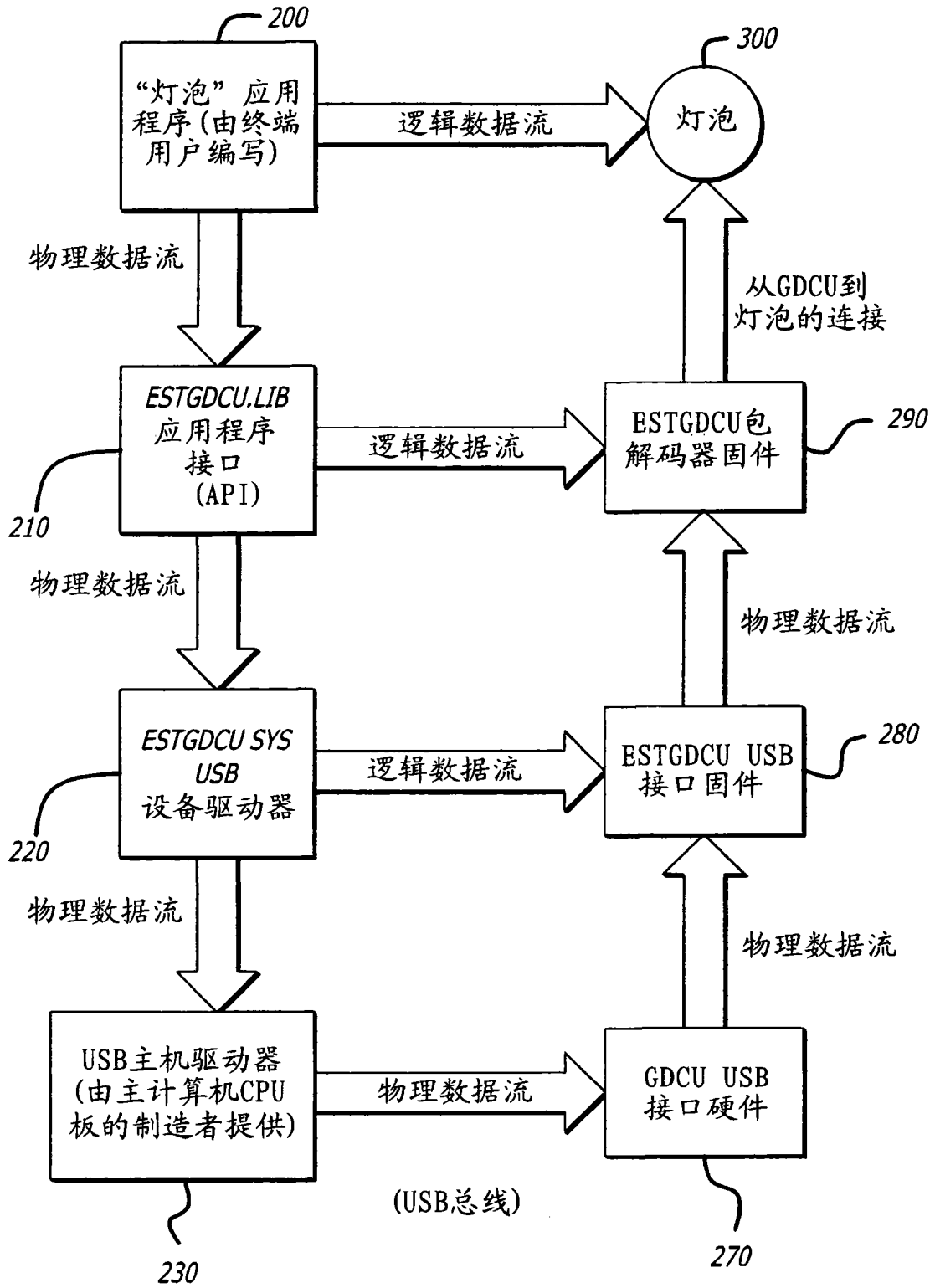


图 5C

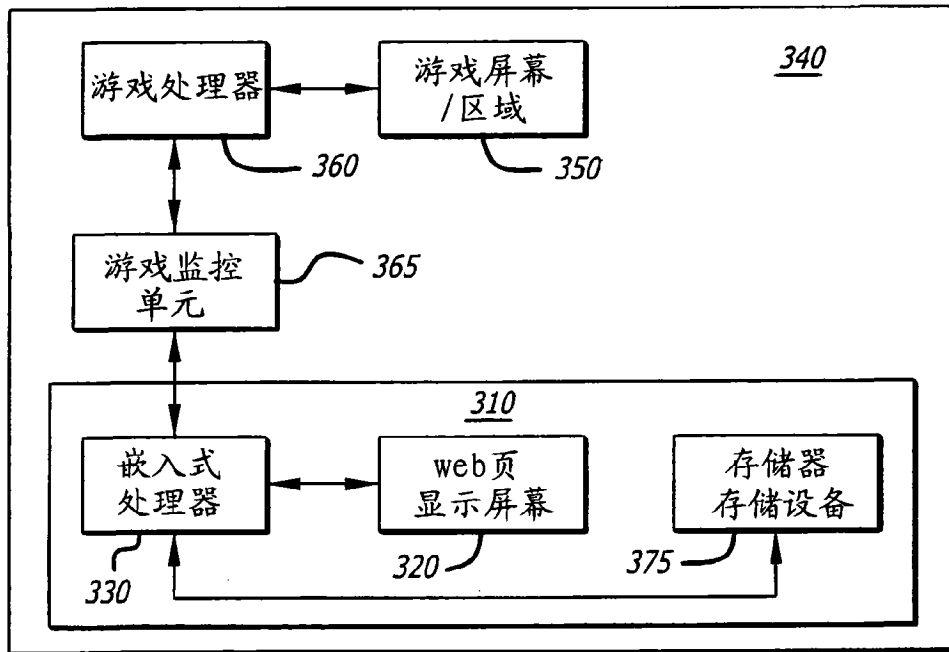


图 6

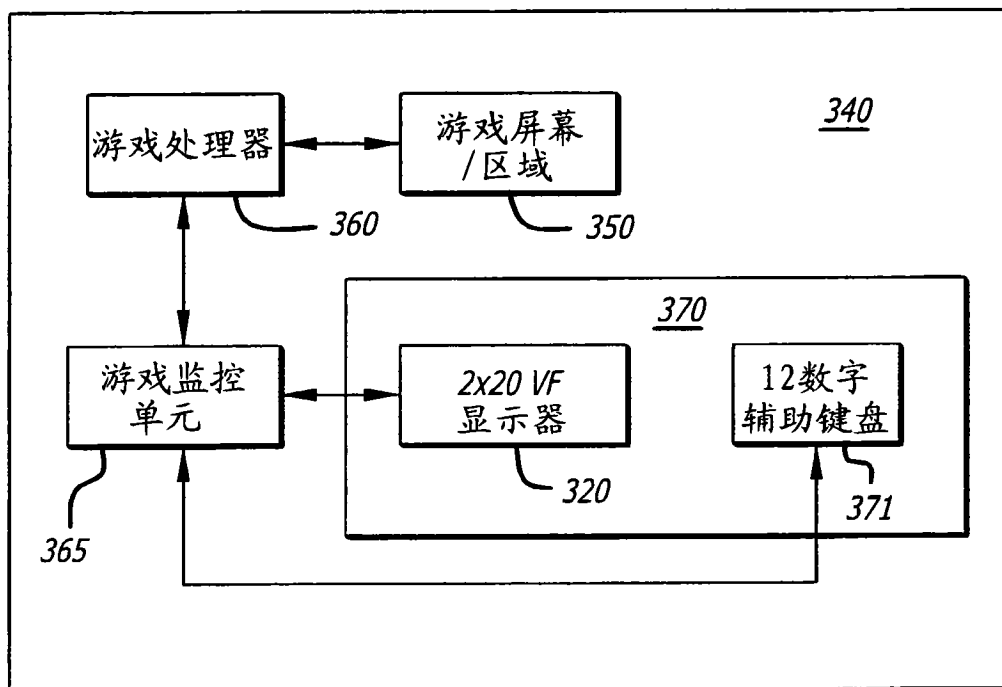


图 7

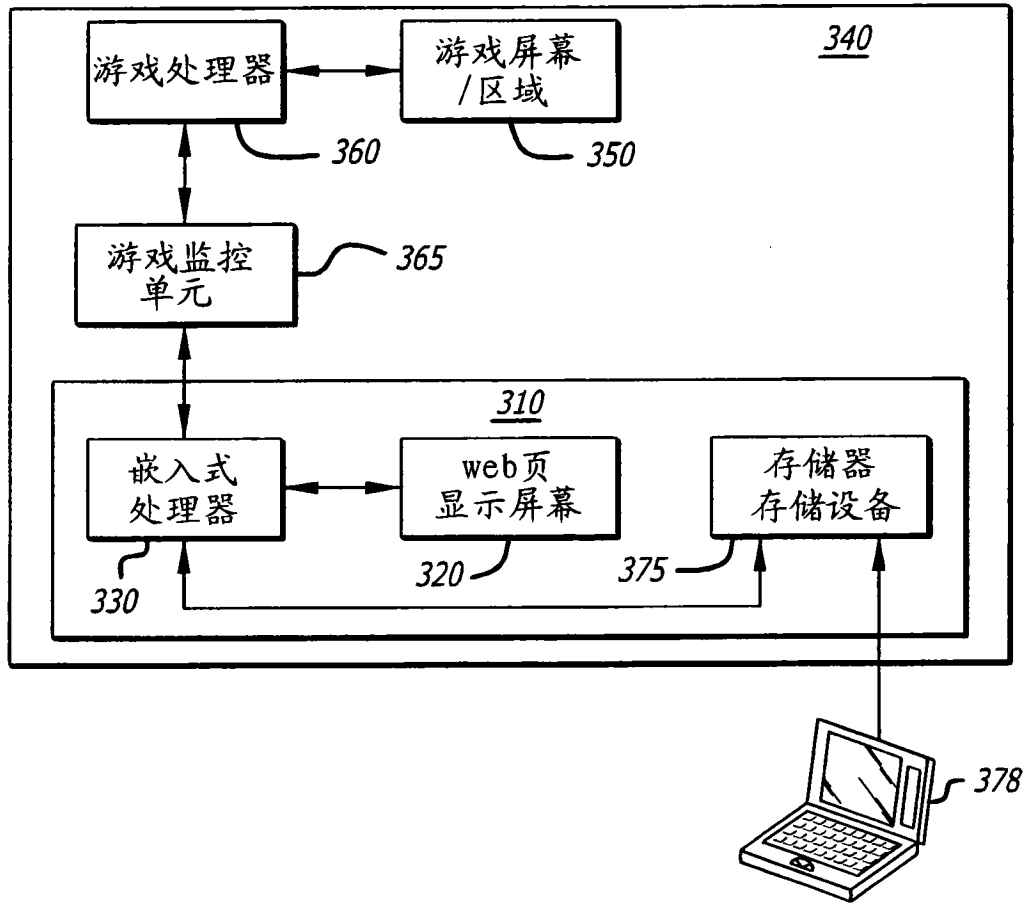


图 8

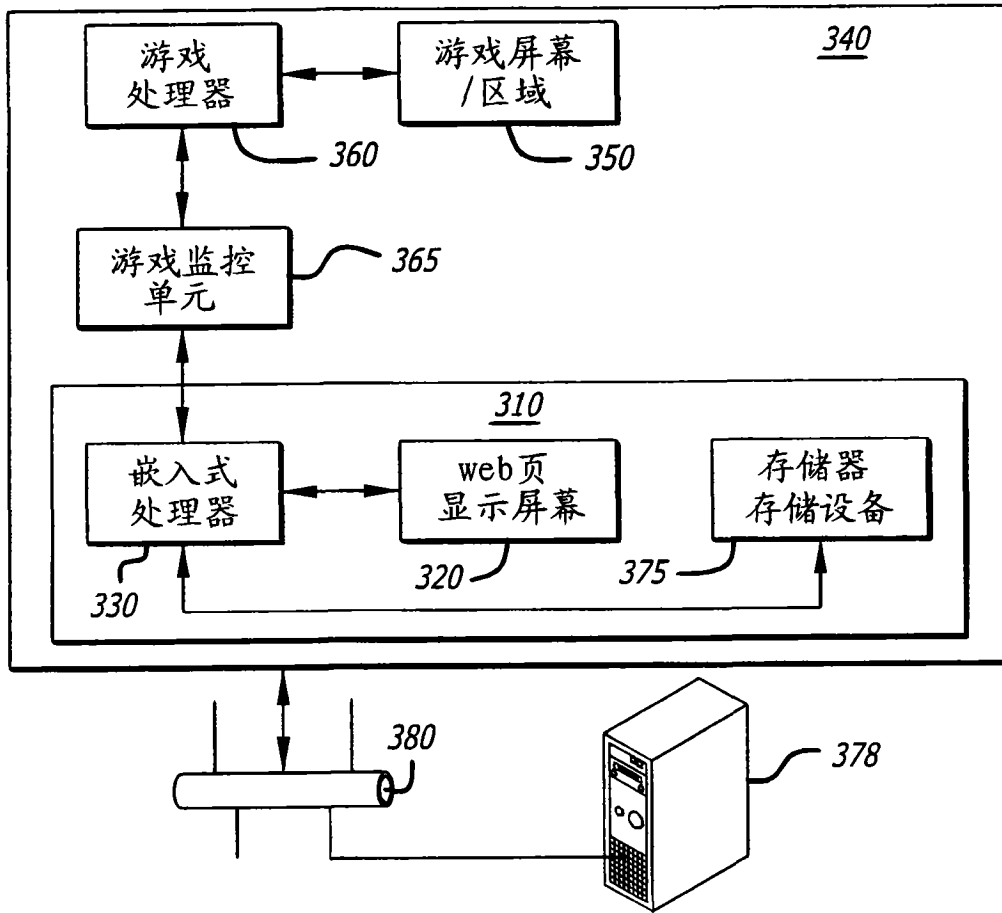


图 9

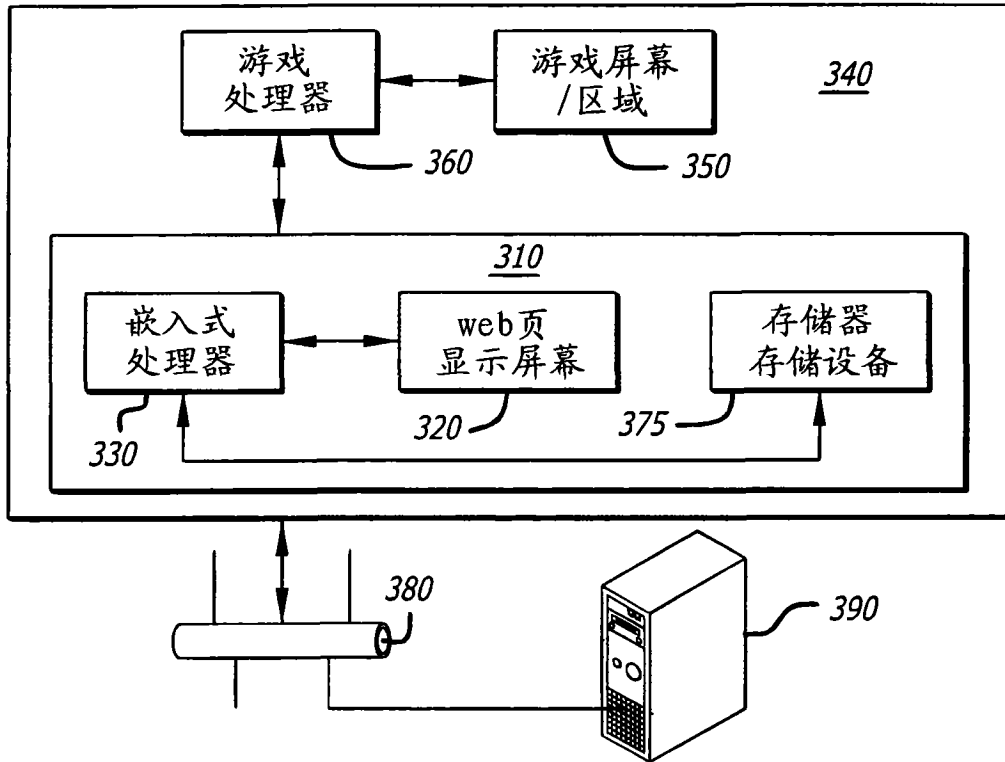


图 10

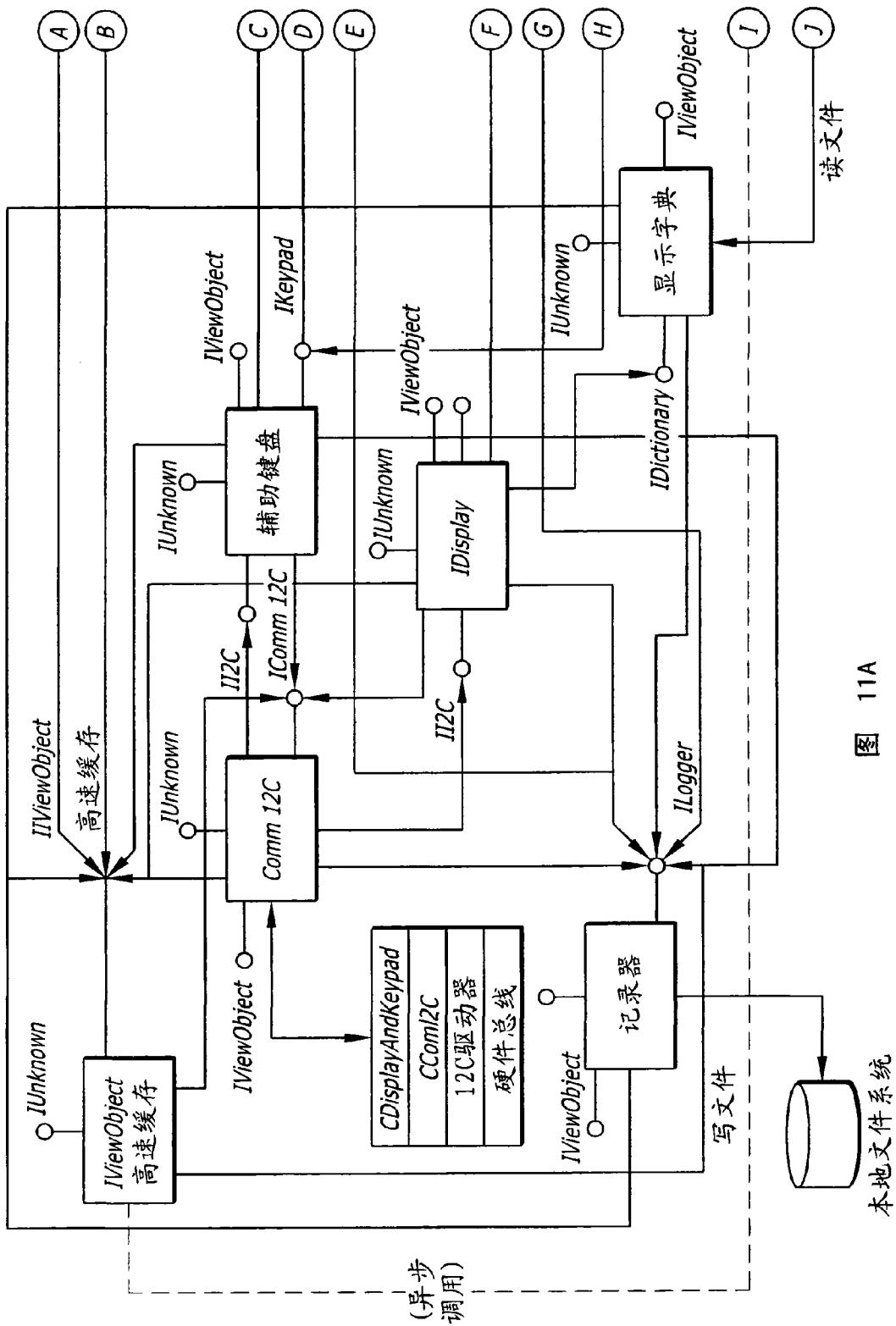


图 11A

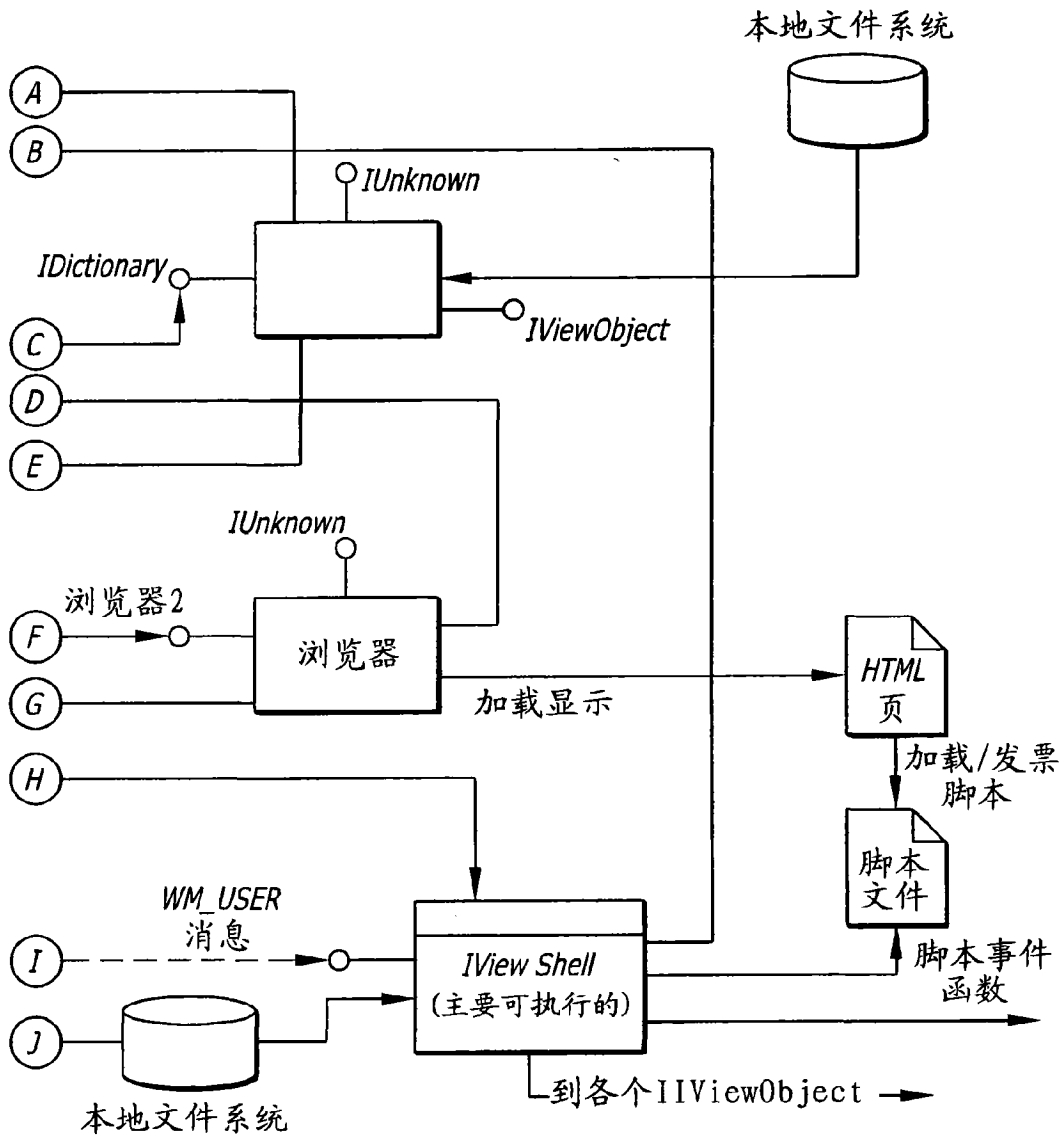


图 11B

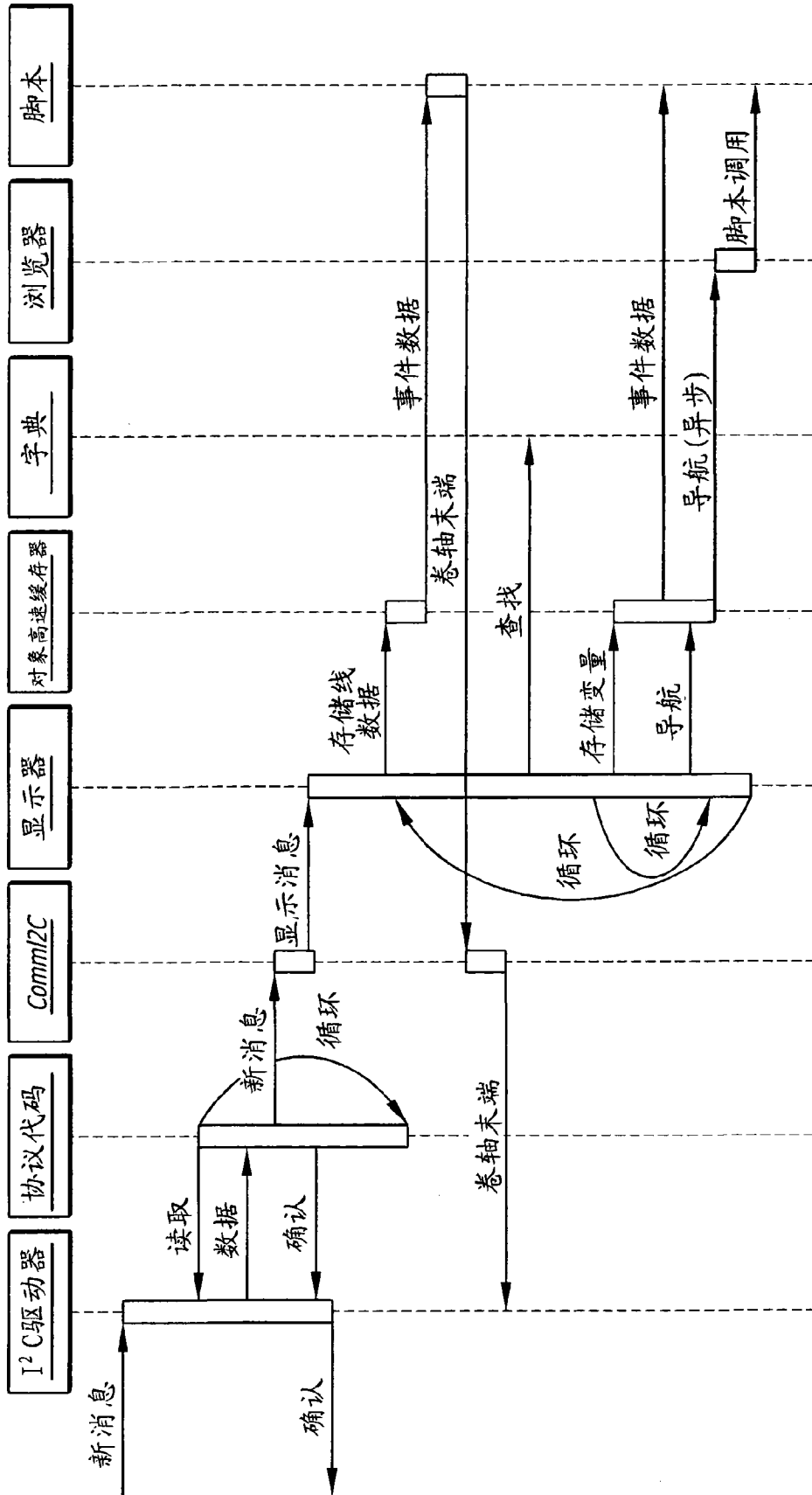


图 12

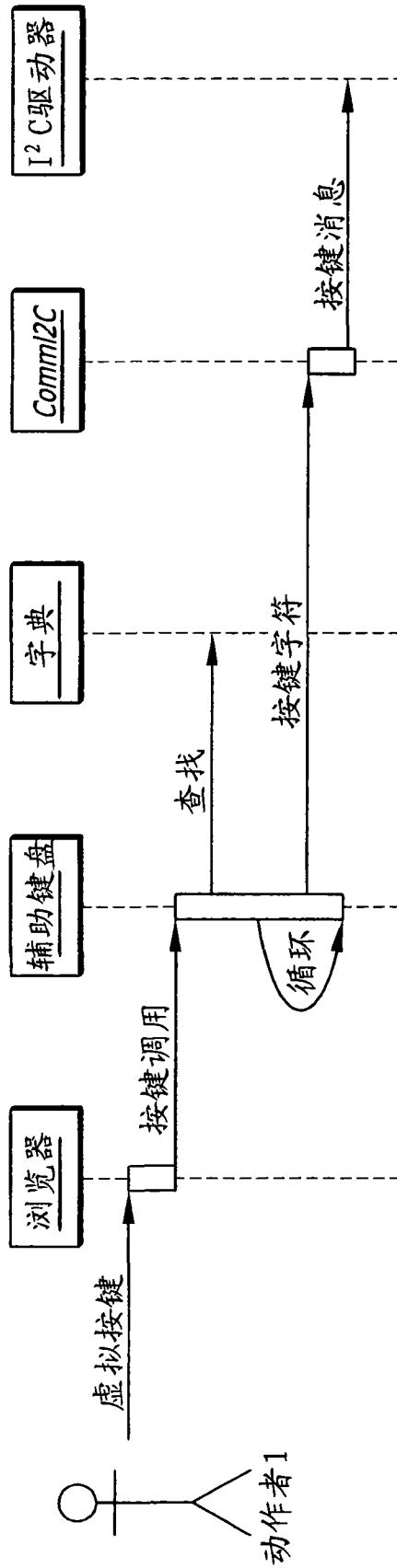


图 13

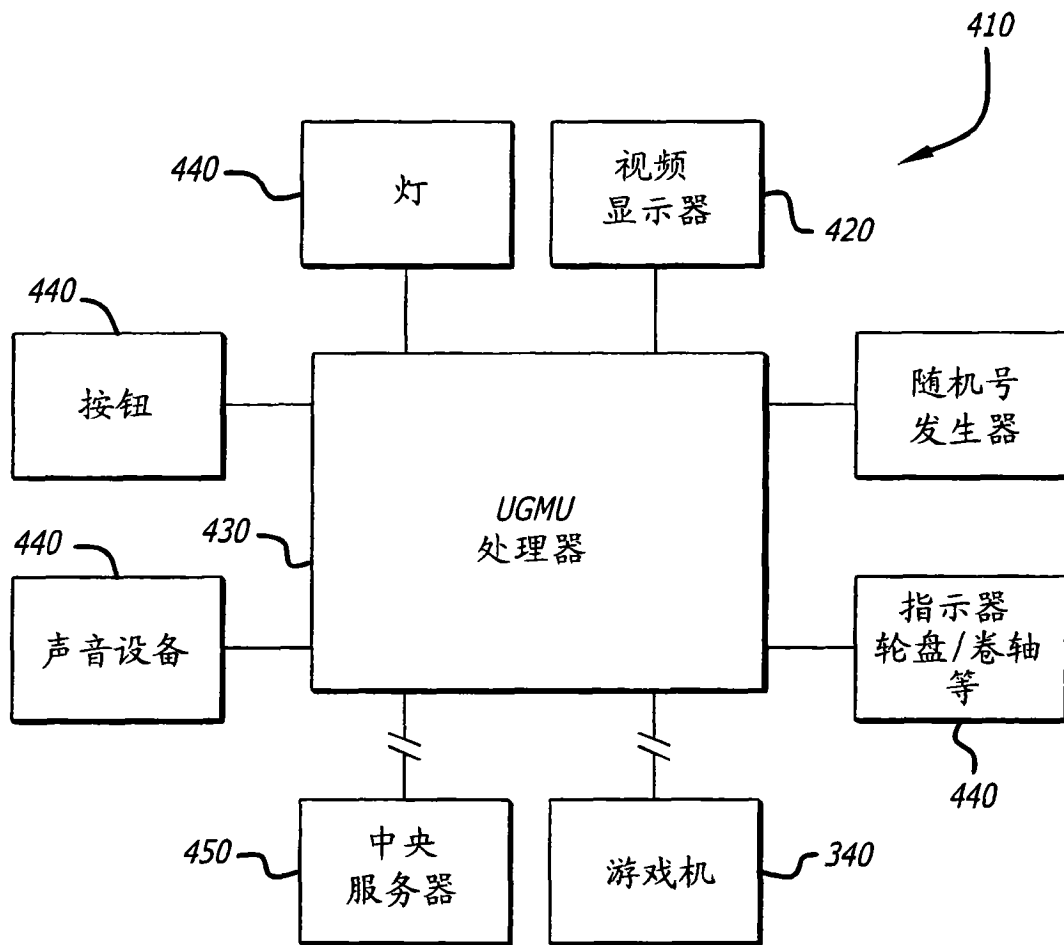


图 14

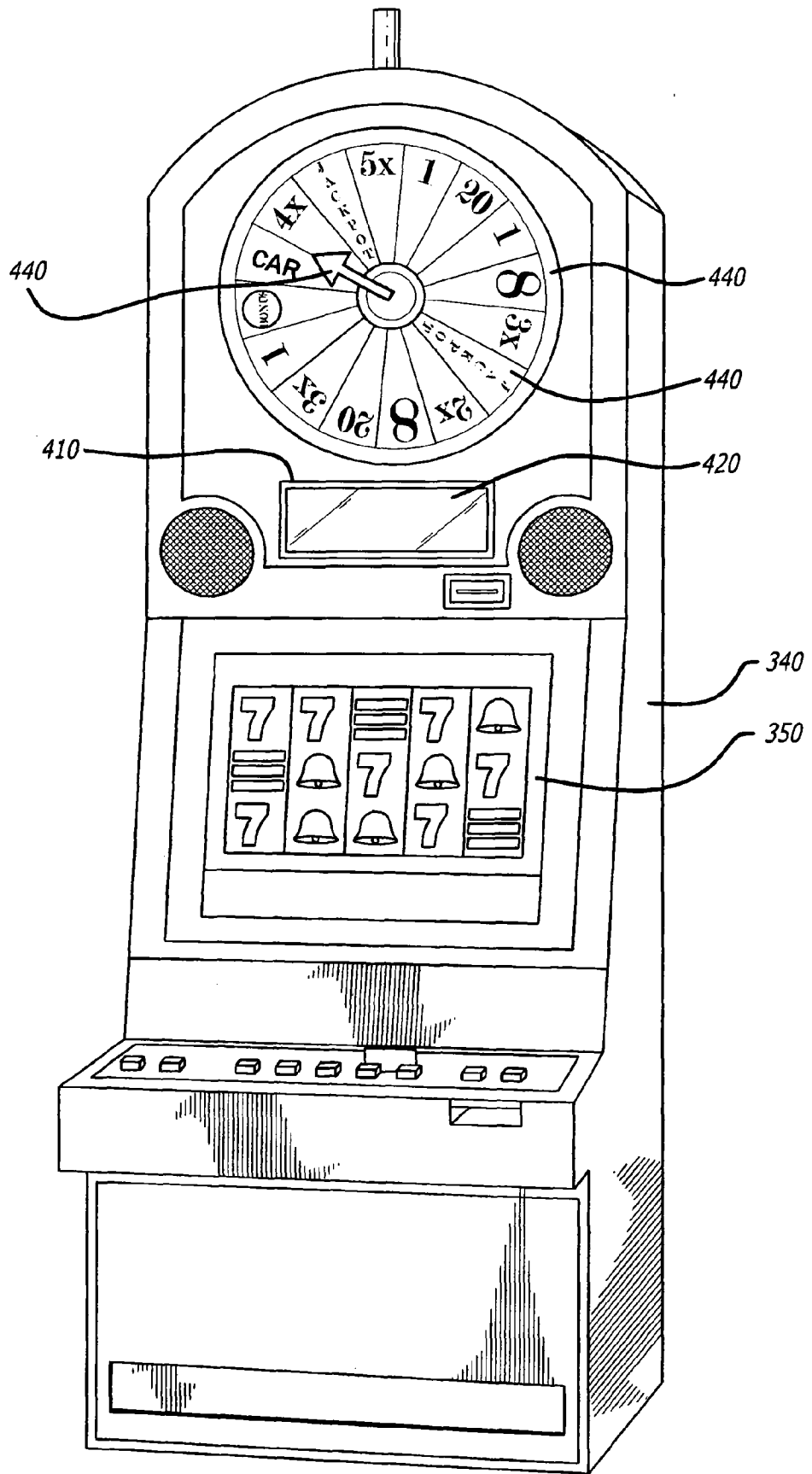


图 15

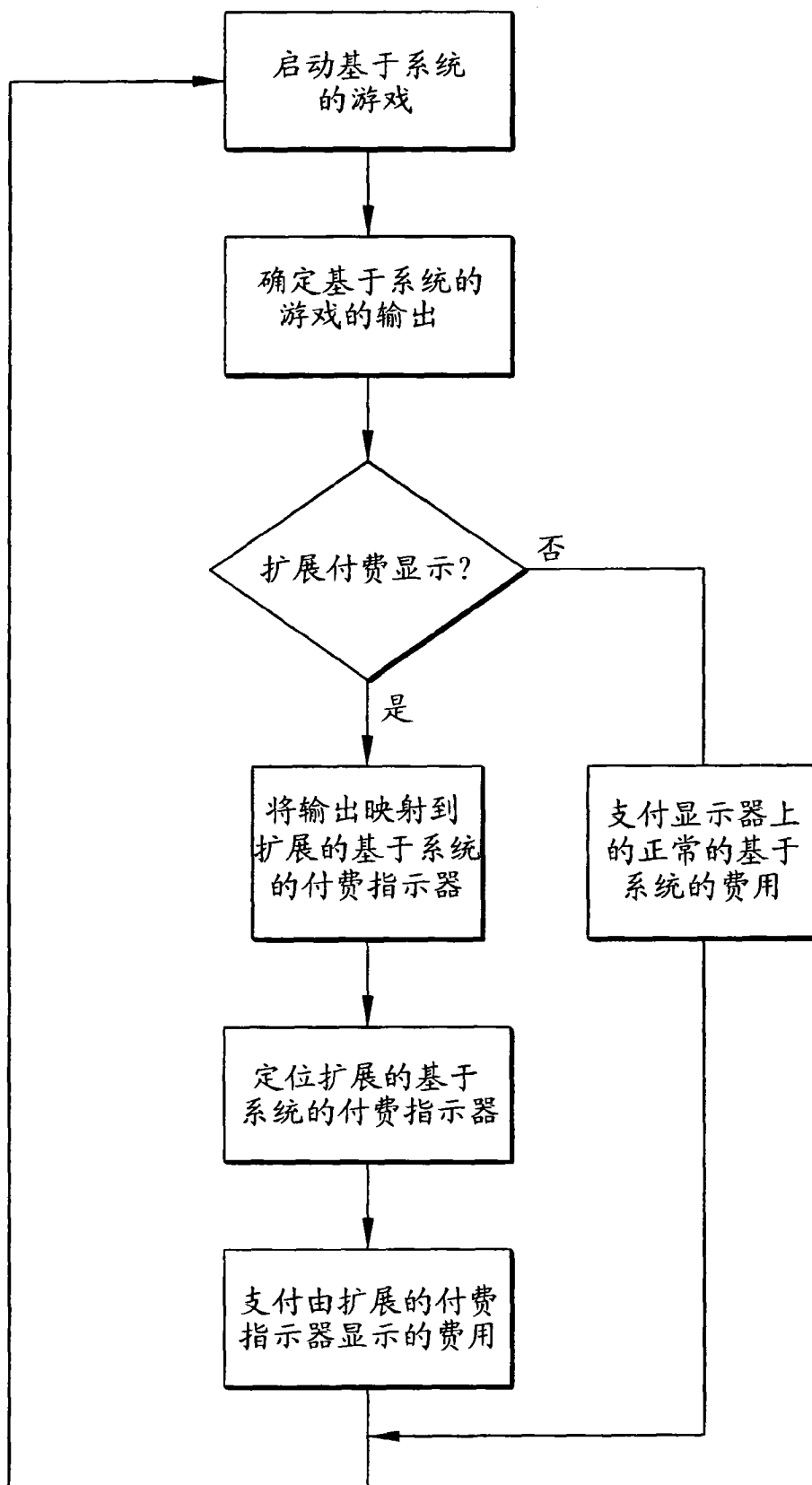


图 16

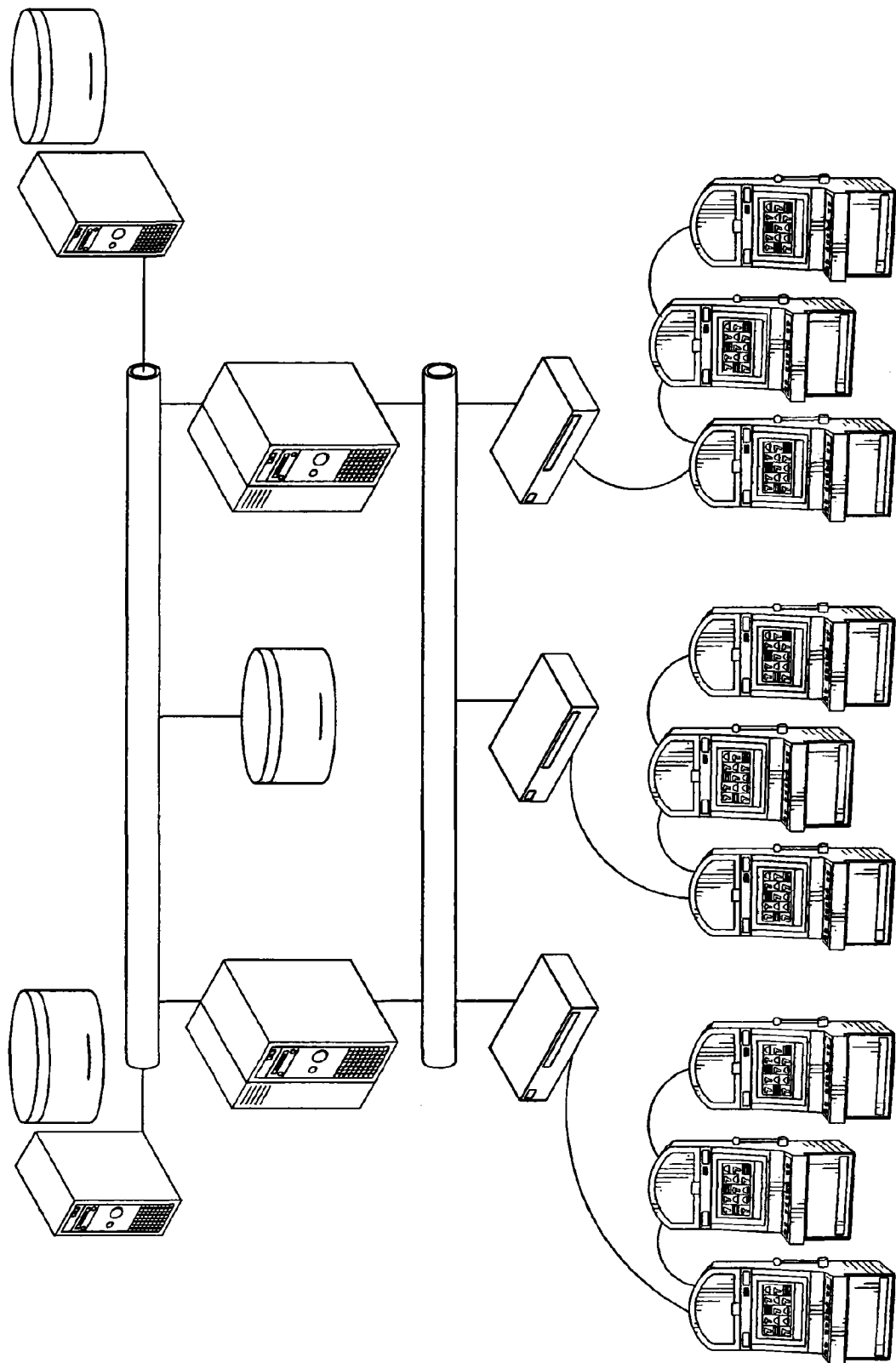


图 17

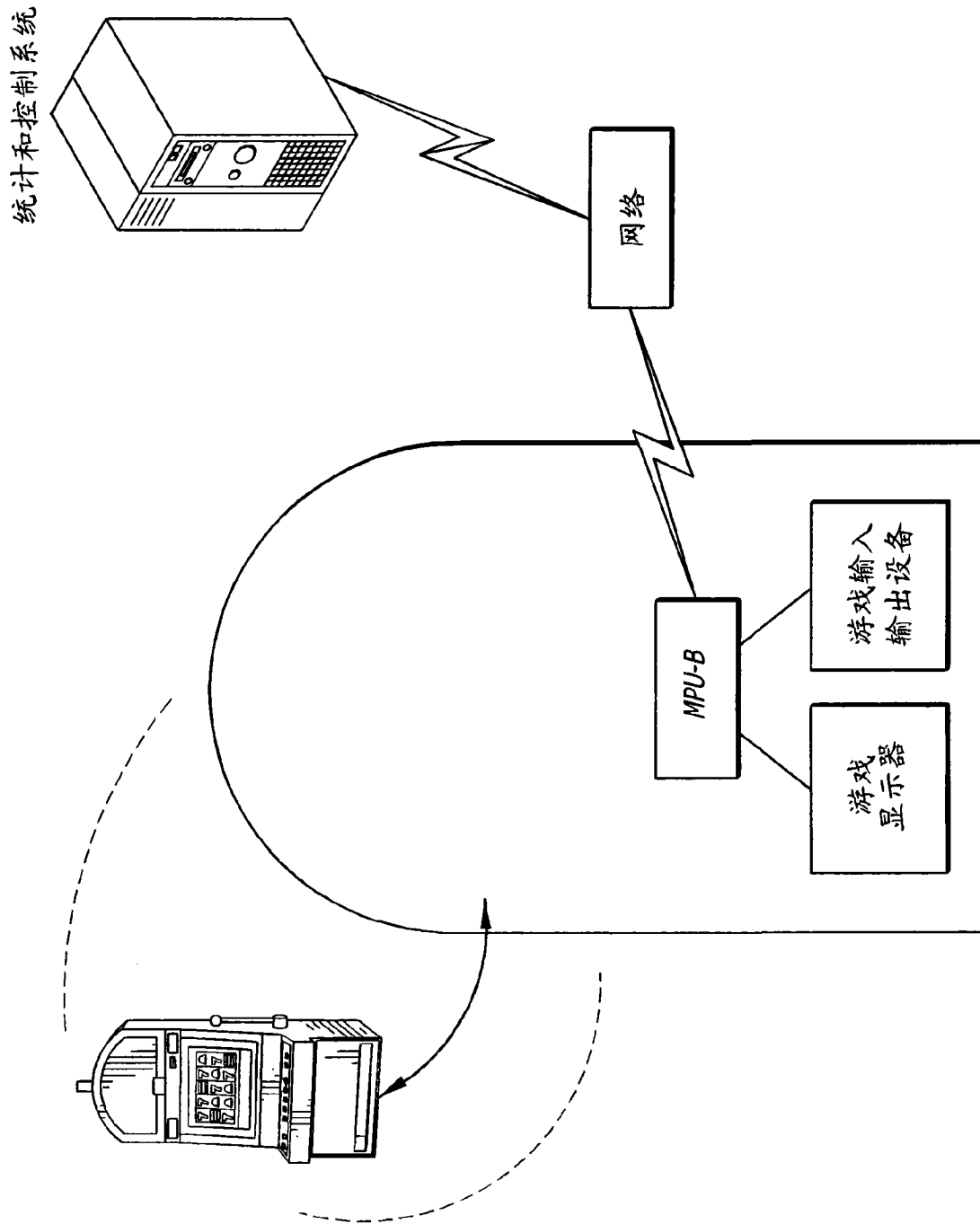


图 18

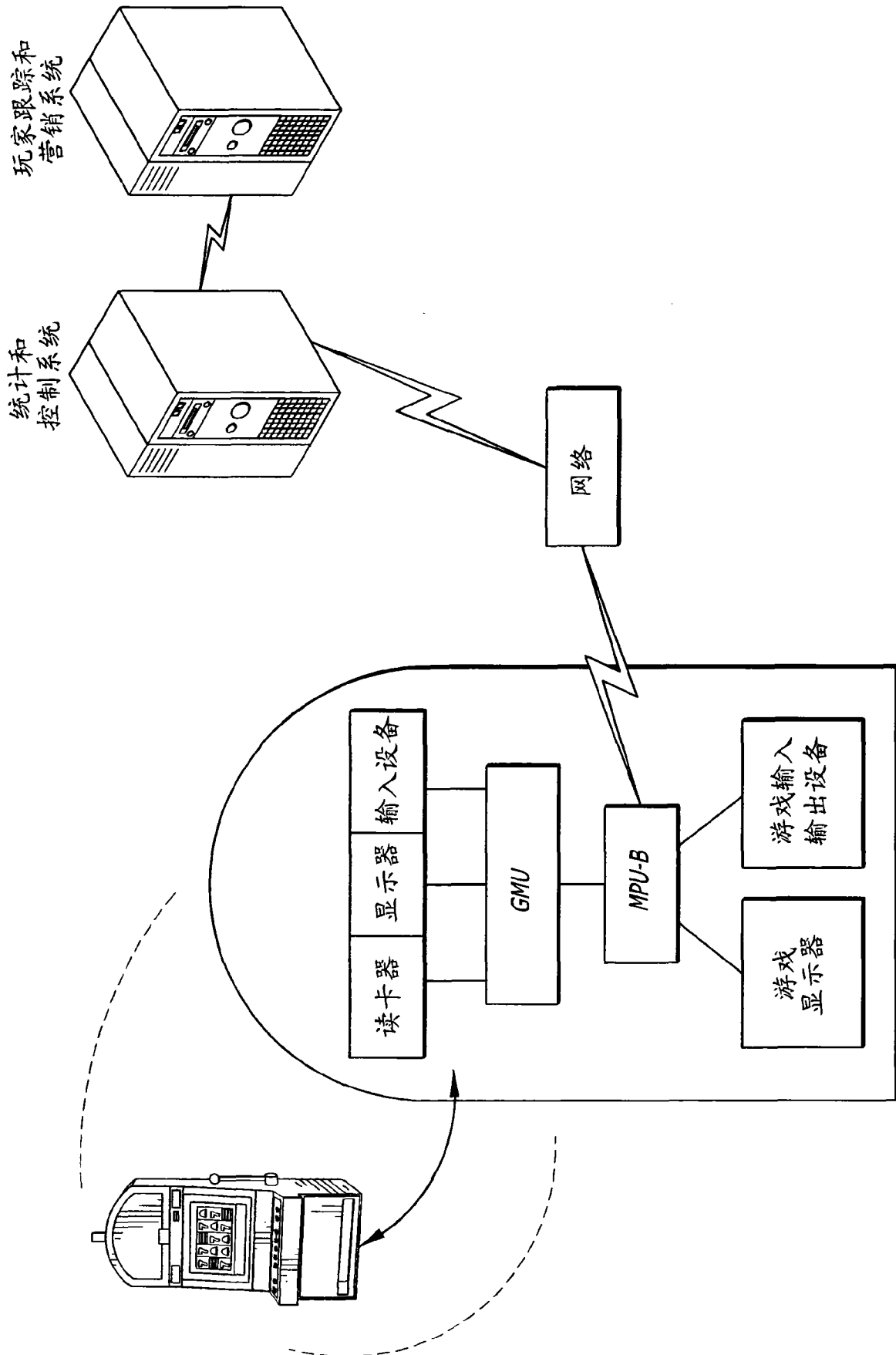


图 19

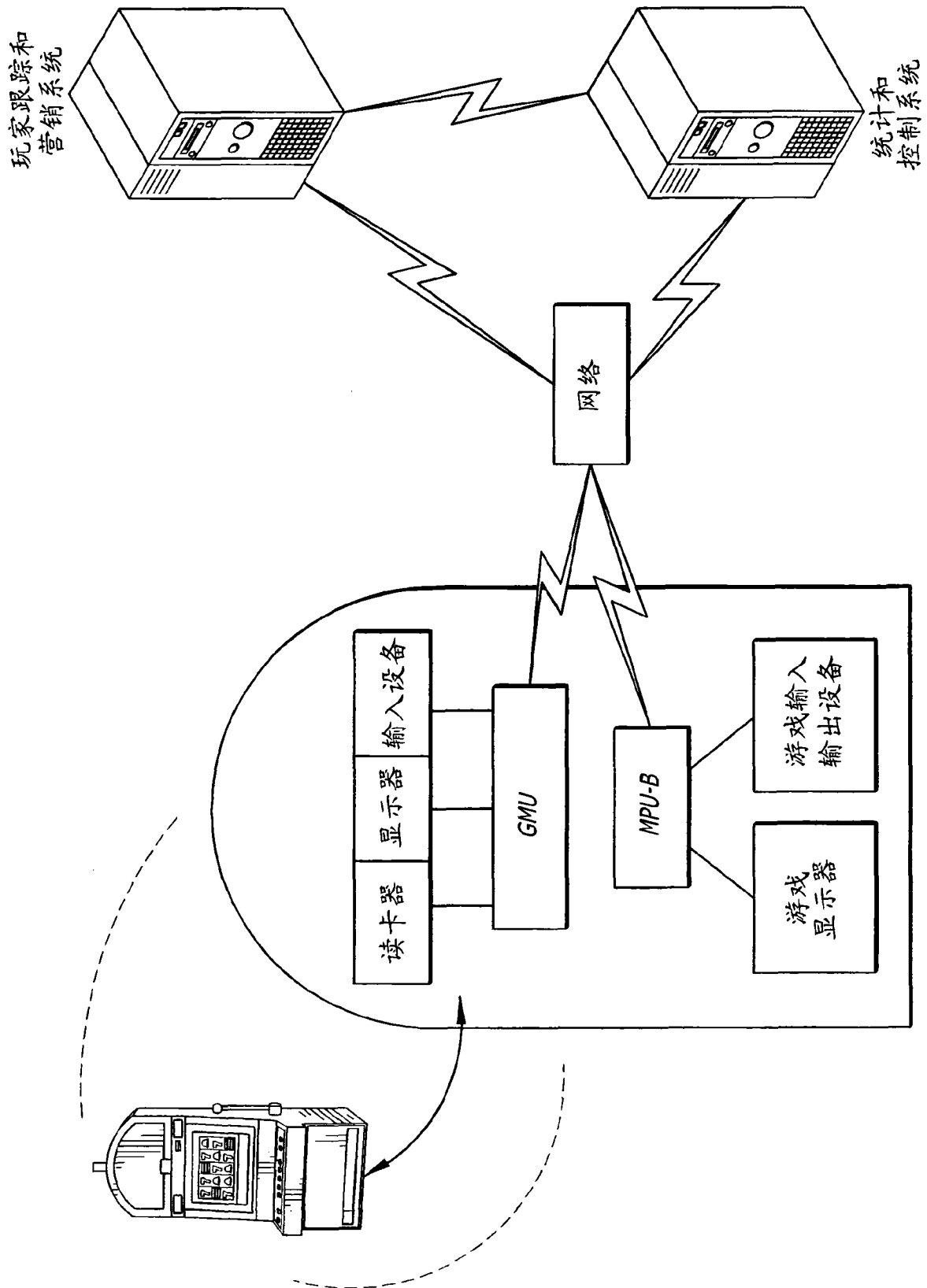


图 20

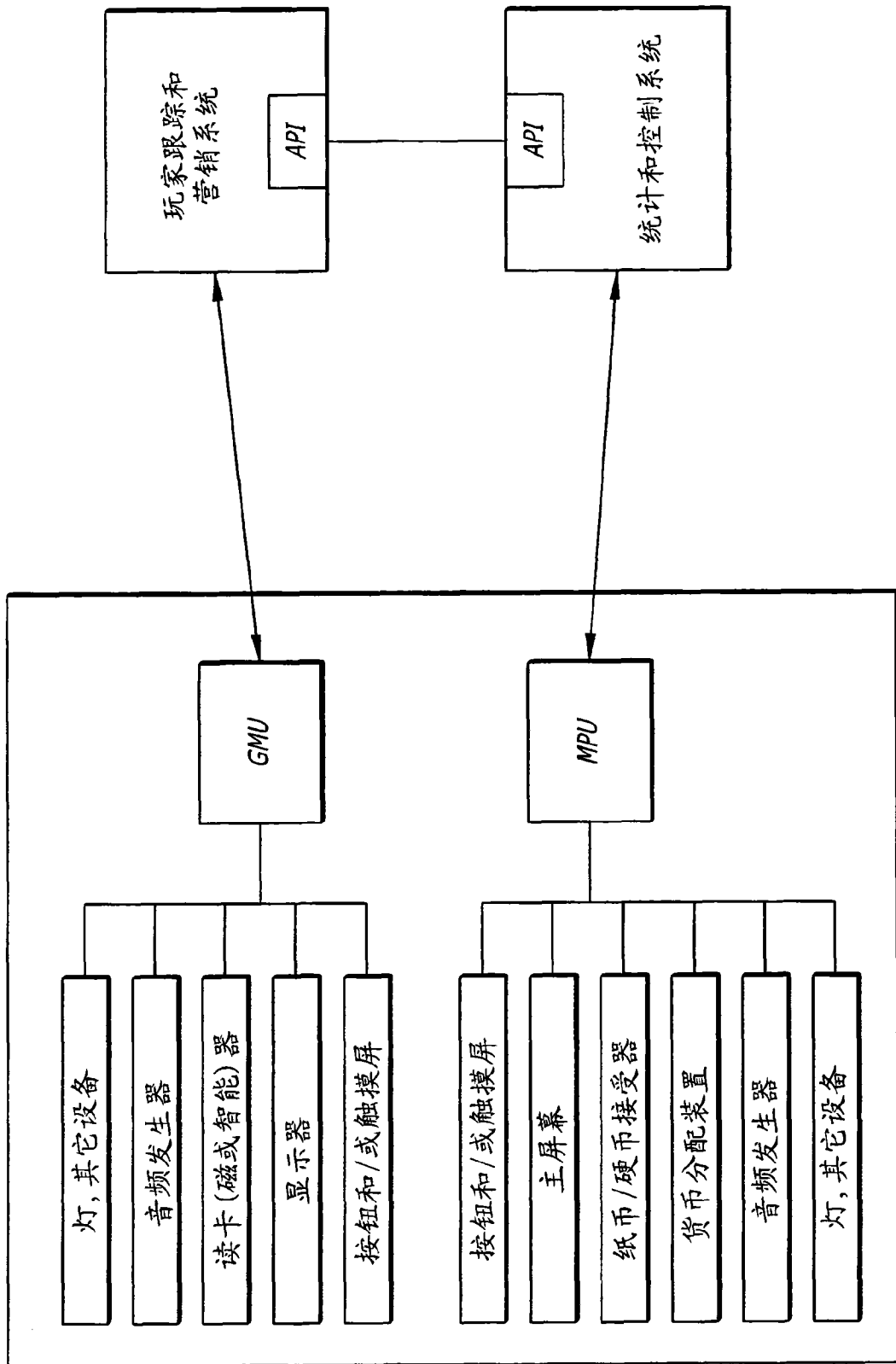


图 21