



## (12)发明专利

(10)授权公告号 CN 104461933 B

(45)授权公告日 2017.10.03

(21)申请号 201410625561.0

(22)申请日 2014.11.07

(65)同一申请的已公布的文献号

申请公布号 CN 104461933 A

(43)申请公布日 2015.03.25

(73)专利权人 珠海全志科技股份有限公司

地址 519080 广东省珠海市香洲区软件园  
路1号生产力加工中心4号楼四层1单元

(72)发明人 张林山

(74)专利代理机构 珠海智专专利商标代理有限公司

公司 44262

代理人 林永协

(51)Int.Cl.

G06F 12/02(2006.01)

(56)对比文件

CN 1359493 A, 2002.07.17,

CN 102169420 A, 2011.08.31,

CN 102495810 A, 2012.06.13,

CN 102023961 A, 2011.04.20,

US 2013339699 A1, 2013.12.19,

审查员 赵海东

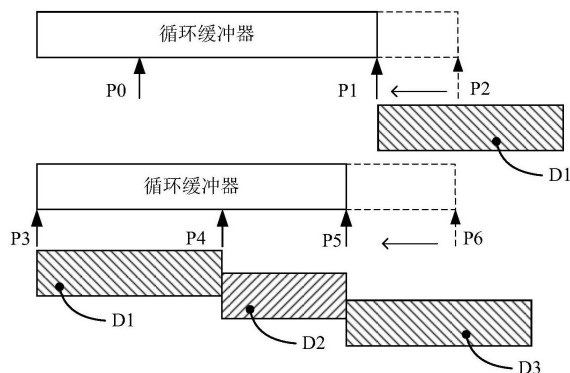
权利要求书2页 说明书5页 附图4页

(54)发明名称

内存管理方法及其装置

(57)摘要

本发明提供一种内存管理方法及其装置,管理方法步骤包括:初始化循环缓冲器的空间;设置数据结束指针指向循环缓冲器的末地址加1的位置上;获取循环缓冲器的剩余空间的剩余容量;判断剩余容量是否大于等于目标数据的长度;如判断剩余容量大于或等于目标数据的长度,目标数据根据剩余空间的首地址写入数据;如判断剩余容量小于目标数据的长度,设置数据结束指针指向剩余空间的首地址。以及应用上述方法的管理装置。通过改变数据结束指针的指向位置有利于节省CPU运行的资源,同时利用对目标数据的长度和剩余空间的容量大小的判断,解决了循环缓冲器在使用过程中数据长度不够用时的的问题,有利于节省内存的占用空间,从而优化运行速度。



### 1. 内存管理方法,包括以下步骤

初始化循环缓冲器的空间,设置剩余空间容量为所述循环缓冲器的整个剩余空间的剩余容量;

其特征在于:

设置数据结束指针指向所述循环缓冲器的末地址加1的位置上;

获取所述剩余空间容量;

判断所述剩余容量是否大于或等于目标数据的一帧数据的长度,如所述剩余容量小于所述目标数据的一帧数据的长度,则等待所述循环缓冲器内的数据被读取,所述循环缓冲器内的数据被读取后返回执行获取所述循环缓冲器的剩余空间的剩余容量的步骤;

如所述剩余容量大于或等于所述目标数据的一帧数据的长度,则执行判断是否越界步骤;

所述判断是否越界的步骤为判断所述数据结束指针与所述目标数据的写入指针之间的容量是否大于或等于所述目标数据的一帧数据的长度;

如所述数据结束指针与所述目标数据的写入指针之间的容量大于或等于所述目标数据的一帧数据的长度,则执行所述目标数据根据剩余空间的首地址写入数据的步骤;

如所述数据结束指针与所述目标数据的写入指针之间的容量小于所述目标数据的一帧数据的长度,则执行所述设置所述数据结束指针指向剩余空间的首地址的步骤;

所述内存管理方法还包括在执行设置所述数据结束指针指向剩余空间的首地址的步骤后,将所述目标数据的写入指针指向所述循环缓冲器的首地址;

更新所述剩余空间容量;

所述目标数据根据剩余空间的首地址写入数据的步骤包括:

将所述剩余空间的首地址发送至解码器;

所述解码器对数据进行解码,并将解码后的数据写入到所述剩余空间。

### 2. 根据权利要求1所述的内存管理方法,其特征在于:

等待所述循环缓冲器内的数据被读取的步骤包括:

获取所需读取数据的长度;

获取循环缓冲器内的数据的长度;

判断所述循环缓冲器内的数据的长度是否大于或等于所述所需读取数据的长度;

如所述循环缓冲器内的数据的长度大于或等于所述所需读取数据的长度,则执行判断所述所需读取数据的长度是否大于所述数据结束指针与所述所需读取数据的读取指针之间的容量;

如所述所需读取数据的长度小于或等于所述数据结束指针与所述所需读取数据的读取指针之间的容量,则执行拷贝循环缓冲器内的数据。

### 3. 根据权利要求2所述的内存管理方法,其特征在于:

如所述所需读取数据的大于所述数据结束指针与所述所需读取数据的读取指针之间的容量,则执行拷贝所述所需读取数据的读取指针和所述数据结束指针之间的数据,然后执行从循环缓冲器的首地址拷贝所述所需读取数据的另一部分数据;

设置所述数据结束指针指向所述循环缓冲器的末地址加1的位置上;

更新所述剩余空间容量。

4. 根据权利要求3所述的内存管理方法,其特征在于:

如所述循环缓冲器内的数据的长度小于所述所需读取数据的长度,则等待所述目标数据完全写入到所述循环缓冲器。

5. 内存管理装置,包括

初始化模块,所述初始化模块用于初始化循环缓冲器的空间并输出初始化信号;

其特征在于:

指针设置模块,所述指针设置模块接收所述初始化信号并输出指针设置信号,所述指针设置模块用于设置数据结束指针指向所述循环缓冲器的末地址加1的位置上,所述指针设置模块还用于设置所述数据结束指针指向剩余空间的首地址;

剩余容量获取模块,所述剩余容量获取模块接收所述指针设置信号并输出剩余容量信号,所述剩余容量获取模块用于获取循环缓冲器的剩余空间的剩余容量;

写入模块,用于根据剩余空间的首地址写入目标数据;

等待模块,用于等待数据的写入或读取;

判断模块,所述判断模块接收所述剩余容量信号,所述判断模块向所述指针设置模块、所述写入模块或所述等待模块输出启动信号,所述判断模块用于判断所述剩余容量是否大于等于目标数据的长度;

所述判断模块包括第一判断模块和第二判断模块;

所述第一判断模块用于判断所述剩余容量是否大于或等于所述目标数据的一帧数据的长度,所述第一判断模块向所述等待模块输出所述启动信号;

所述第二判断模块用于判断所述数据结束指针与所述目标数据的写入指针之间的容量是否大于或等于所述目标数据的一帧数据的长度,所述第二判断模块向所述指针设置模块或所述写入模块输出所述启动信号;

所述内存管理装置还包括解码器;

所述写入模块用于将所述剩余空间的首地址发送至所述解码器;

所述解码器还用于数据解码,所述解码器可解码出脉冲编码调制格式的数据,所述解码器可将解码后的脉冲编码调制格式的数据写入到所述剩余空间。

## 内存管理方法及其装置

### 技术领域

[0001] 本发明涉及内存数量处理领域,具体涉及一种内存管理方法及装置。

### 背景技术

[0002] 脉冲编码调制(Pulse Code Modulation),简称PCM,是对连续变化的模拟信号进行抽样、量化和编码产生的数字信号,PCM的优点就是音质好,缺点就是体积大。PCM可以提供用户从2M到155M速率的数字数据专线业务,也可以提供话音、图象传送、远程教学等其他业务。而对压缩数据解码成PCM数据一般的做法是单独在内存上开辟一个空间,通过解码器对PCM数据的一帧数据进行解码,并将该一帧数据写入该空间上,每次进行一次解码一帧数据,就把这个单独开辟的空间复制到循环缓冲器中,这样导致也增加了一次复制。上述的解码过程为内存的可用资源造成浪费,也会大大影响机器的运行速度。另一种做法是为每一帧开辟一个空间,在不确定一帧大小时,只能开辟最大的数据空间,浪费了数据空间,在内存空间受限的时,无法满足要求。

### 发明内容

[0003] 本发明的主要目的是提供一种可节省内存空间和优化运行速度的内存管理方法。

[0004] 本发明的第二目的是提供一种可节省内存空间和优化运行速度的内存管理装置。

[0005] 为了实现本发明的主要目的,本发明提供一种内存管理方法,其步骤包括:初始化循环缓冲器的空间,设置剩余空间容量为循环缓冲器的全部空间容量;设置数据结束指针指向循环缓冲器的末地址加1的位置上;获取剩余空间容量;获取循环缓冲器的剩余空间的剩余容量;判断剩余容量是否大于或等于目标数据的一帧数据的长度,如剩余容量小于目标数据的一帧数据的长度,则等待循环缓冲器内的数据被读取,循环缓冲器内的数据被读取后返回执行获取循环缓冲器的剩余空间的剩余容量的步骤;如剩余容量大于或等于目标数据的一帧数据的长度,则执行判断是否越界步骤;判断是否越界的步骤为判断数据结束指针与目标数据的写入指针之间的容量是否大于或等于目标数据的一帧数据的长度;如数据结束指针与目标数据的写入指针之间的容量大于或等于目标数据的一帧数据的长度,则执行目标数据根据剩余空间的首地址写入数据的步骤;如数据结束指针与目标数据的写入指针之间的容量小于目标数据的一帧数据的长度,则执行设置数据结束指针指向剩余空间的首地址的步骤。

[0006] 由上述方案可见,通过改变数据结束指针的指向位置有利于节省CUP运行的资源,同时利用对目标数据的长度和剩余空间的容量大小的判断,解决了循环缓冲器在使用过程中数据长度不够用时的问题,有利于节省内存的占用空间,从而优化运行速度。且对于不固定帧长的数据,或者剩余空间不是帧长的整数倍时,将会出现浪费内存空间,通过判断剩余空间是否大于一帧数据的长度,使得不再需要单独开辟内存空间,直接利用循环缓冲器的空间进行处理即可,亦可以有效地节省内存的占用空间。

[0007] 更进一步的方案是,目标数据根据剩余空间的首地址写入数据的步骤包括:将剩

余空间的首地址发送至解码器;解码器对数据解码成脉冲码编码调制格式的数据,并将解码后的数据写入到剩余空间,更新剩余空间容量。。

[0008] 由上可见,在解码数据时通常是需要要在内存单独在开设另一空间进行解码,随后在拷贝到循环缓冲器内,而本发明提供的内存管理方法采用是将剩余空间的首地址发送至解码器,解码器解码,并将解码后的数据写入到剩余空间中。

[0009] 更进一步的方案是,等待循环缓冲器内的数据被读取的步骤包括:获取所需读取数据的长度;获取循环缓冲器内的数据的长度;判断循环缓冲器内的数据的长度是否大于或等于所需读取数据的长度;如循环缓冲器内的数据的长度大于或等于所需读取数据的长度,则执行判断所需读取数据的长度是否大于或等于数据结束指针与所需读取数据的读取指针之间的容量;如所需读取数据的长度小于或等于数据结束指针与所需读取数据的读取指针之间的容量,则执行拷贝循环缓冲器内的数据。

[0010] 由上可见,CPU或其他程序需要拷贝循环缓冲器内的数据时,则通过判断循环缓冲器内的数据的长度是否大于或等于所需读取数据的长度,从而避免拷贝数据时出现错漏现象。

[0011] 更进一步的方案是,如所需读取数据的长度大于数据结束指针与所需读取数据的读取指针之间的容量,则执行拷贝所需读取数据的读取指针和数据结束指针之间的数据,然后执行从循环缓冲器的首地址拷贝所需读取数据的另一部分数据;设置数据结束指针指向循环缓冲器的末地址加1的位置上,更新剩余空间容量。。

[0012] 更进一步的方案是,如循环缓冲器内的数据的长度小于所需读取数据的长度,则等待目标数据完全写入到循环缓冲器。

[0013] 由上可见,在读取数据是通过复位数据结束指针指向循环缓冲器的末地址加1的位置上,使得循环缓冲器的剩余空间相对于数据写入时的剩余空间较大,这样有利于提高拷贝数据的效率。

[0014] 为了实现本发明的第二目的,本发明提供一种内存管理装置,包括初始化模块、指针设置模块、剩余容量获取模块、判断模块、写入模块和等待模块,初始化模块用于初始化循环缓冲器的空间并输出初始化信号,其中,指针设置模块接收初始化信号并输出指针设置信号,指针设置模块用于设置数据结束指针指向循环缓冲器的末地址加1的位置上,指针设置模块还用于设置数据结束指针指向剩余空间的首地址,剩余容量获取模块接收指针设置信号并输出剩余容量信号,剩余容量获取模块用于获取循环缓冲器的剩余空间的剩余容量,写入模块用于根据剩余空间的首地址写入目标数据,等待模块用于等待数据的写入或读取,判断模块接收剩余容量信号,判断模块向指针设置模块、等待模块或写入模块输出启动信号,判断模块用于判断剩余容量是否大于等于目标数据的长度。

[0015] 由上述方案可见,通过改变数据结束指针的指向位置有利于节省CUP运行的资源,同时利用对目标数据的长度和剩余空间的容量大小的判断,解决了循环缓冲器在使用过程中数据长度不够用时的问题,有利于节省内存的占用空间,从而优化运行速度。

[0016] 更进一步的方案是,判断模块包括第一判断模块和第二判断模块;第一判断模块用于判断剩余容量是否大于或等于目标数据的一帧数据的长度,第一判断模块向等待模块输出启动信号;第二判断模块用于判断数据结束指针与目标数据的写入指针之间的容量是否大于或等于目标数据的一帧数据的长度,第二判断模块向指针设置模块或写入模块输出

启动信号。

[0017] 由上可见,对于不固定帧长的数据,或者剩余空间不是帧长的整数倍时,将会出现浪费内存空间,通过判断模块去判断剩余空间是否大于一帧数据的长度,使得不再需要单独开辟内存空间,直接利用循环缓冲器的空间进行处理即可,可以有效地节省内存的占用空间。

## 附图说明

[0018] 图1是本发明内存管理方法实施例中写入数据时的流程图。

[0019] 图2是本发明内存管理方法实施例写入数据时的原理图。

[0020] 图3是本发明内存管理方法实施例中读取数据时的流程图。

[0021] 图4是本发明内存管理装置实施例的系统框图。

[0022] 以下结合附图及实施例对本发明作进一步说明。

## 具体实施方式

[0023] 内存管理方法实施例:

[0024] 参照图1,图1是内存管理方法实施例中写入数据时的流程图。对内存进行管理时,首先执行步骤S11,即首先初始化循环缓冲器的空间,循环缓冲器为开设在内存上的循环缓冲器(buffer),同时设置剩余空间容量为循环缓冲器的全部空间容量,将写指针和读指针指向循环缓冲器的首地址。然后执行步骤S12,即设置数据结束指针(pEnd)为循环缓冲器的结束位置,也就是将数据结束指针指向循环缓冲器的末地址加1的位置上。随后执行步骤S13,即获取循环缓冲器的剩余空间的剩余容量大小,再执行步骤S14获取目标数据的长度,目标数据是指所需要写入的数据。

[0025] 随后执行判断步骤S15,即判断剩余容量是否大于或等于目标数据的一帧数据的长度,当剩余容量小于目标数据的一帧数据的长度时,则执行步骤S151,即等待循环缓冲器内的数据被读取后,释放循环缓冲器内空间。执行完毕步骤S151后,则返回步骤S13,重新获取剩余空间的剩余容量。当剩余容量大于或等于目标数据的一帧数据的长度时,则执行步骤S16,即判断是否越界步骤。判断是否越界步骤是判断数据结束指针与目标数据的写入指针之间的容量是否大于或等于目标数据的一帧数据的长度。

[0026] 如数据结束指针与目标数据的写入指针之间的容量大于或等于目标数据的一帧数据的长度,则目标数据将根据剩余空间的首地址写入所需要写入的数据,该步骤具体包括依次执行的步骤S171、步骤S172和步骤S173。首先执行步骤S171,即将当前剩余空间的首地址发送至解码器,随后执行步骤S172,解码器进行解码,并将解码后的数据从解码器的写入指针开始写入到剩余空间中。然后执行步骤S173,更新PCM数据空间,使得下一次的解码或写入从未写区域开始进行。更新完毕PCM数据空间后则返回步骤S13,从而继续进行下一步的写入程序。

[0027] 当在执行判断步骤S16时,数据结束指针与目标数据的写入指针之间的容量小于目标数据的一帧数据的大小时,则执行步骤S181,设置数据结束指针指向剩余空间的首地址。随后再执行步骤S182,将写入指针指向循环缓冲器的首地址,再执行步骤S183,更新剩余空间容量大小。最后返回步骤S13,从而继续进行下一步的写入程序。

[0028] 参照图2,图2是内存管理方法实施例中写入数据时的原理图。当其他外部数据从位于P0的位置写入到循环缓冲器中,并且写入完成后当前指针位于P1的位置,此时需要写入PCM数据的一帧数据D1时,由于此时数据结束指针位于P2,同时由于循环缓冲器中P1与P2之间的剩余空间容量小于数据D1的长度,所以将设置数据结束指针指向剩余空间的首地址即P1处的位置,同时将写入指针更新到下一循环的循环缓冲器的首地址P3处。更新剩余空间长度,其方法是:剩余空间长度等于原来剩余空间长度减去P1到P2之间的长度。

[0029] 进入下一循环缓冲器后,当读数据流程读取到P1处时则判断数据结束,并将数据结束指针指向位置P6处,即将数据结束指针指向循环缓冲器的末地址加1的位置上,读数据流程从循环空间首地址开始继续读取数据。随后将对一帧数据D1进行写入,由于循环缓冲器中P3与P6之间的剩余空间容量大于数据D1的大小,则将循环缓冲器的剩余空间首地址即P3发送至解码器,解码器根据地址P3写入数据。随后如将要写入数据D2时,则判断循环缓冲器中剩余空间的容量,即判断当前指针的位置P4与P6之间的剩余空间容量是否大于数据D2的大小,由于当前剩余空间的容量大于数据D2的大小,则数据D2根据当前剩余空间的首地址即P4进行写入数据。随后如再需要对PCM数据的另一帧数据D3进行写入时,由于此时数据结束指针位于P6,同时由于循环缓冲器中P5与P6之间的剩余空间容量小于数据D3的大小,所以将设置数据结束指针指向当前剩余空间的首地址即P5处的位置。由上可见,通过改变数据结束指针的指向位置有利于节省CUP运行的资源,从而优化运行速度。

[0030] 参照图3,图3是内存管理方法实施例中读取数据时的流程图。在目标数据根据剩余空间的首地址写入数据的步骤或设置数据结束指针指向剩余空间的首地址的步骤执行完毕后,首先执行步骤S21,即读取PCM数据,随后执行步骤S22,获取所需读取数据的长度,然后再执行步骤S23,获取循环缓冲器内数据的长度。随后在执行步骤S24,判断循环缓冲器内的数据的长度是否大于或等于所需读取数据的长度。

[0031] 如循环缓冲器内的数据的长度小于所需读取数据的长度,则执行步骤S241,即等待所需的数据完全写入到循环缓冲器内,待所需的数据完全写入到循环缓冲器后,则返回执行步骤S21。

[0032] 如循环缓冲器内的数据的长度大于或等于所需读取数据的长度,则执行判断步骤S25,即判断所需读取数据的长度是否大于或等于数据结束指针与所需读取数据的读取指针之间的容量。

[0033] 如所需读取数据的长度等于或小于数据结束指针与所需读取数据的读取指针之间的容量,则执行步骤S26,即拷贝当前数据到内存的其他位置上,再执行步骤S27,即更新PCM数据空间。

[0034] 如所需读取数据的长度大于数据结束指针与所需读取数据的读取指针之间的容量,则执行步骤S28,即拷贝所需读取数据的读取指针和数据结束指针之间的数据,再执行步骤S29,从循环缓冲器的首地址拷贝所需读取数据的另一部分数据,最后在执行步骤S30,将数据结束指针指向循环缓冲器的末地址加1的位置上。

[0035] 内存管理装置实施例:

[0036] 参照图4,图4是内存管理装置3的系统框图,内存管理装置3包括初始化模块31、指针设置模块32、剩余容量获取模块33、判断模块34、写入模块35、解码器36和等待模块38。初始化模块31用于初始化循环缓冲器的空间向指针设置模块32输出初始化信号。指针设置模

块32接收到初始化信号后,指针设置模块32除了可将数据结束指针指向循环缓冲器的末地址加1的位置上,指针设置模块32还可将数据结束指针指向剩余空间的首地址,在指针设置模块32将数据结束指针设置完毕后则向剩余容量模块33输出指针设置信号。

[0037] 剩余容量获取模块33接收指针设置信号后,剩余容量获取模块33将获取循环缓冲器的剩余空间的剩余容量,剩余容量获取模块33向判断模块34输出剩余容量信号。

[0038] 判断模块34包括第一判断模块341和第二判断模块342。判断模块34接收到剩余容量信号后,判断模块34将判断剩余容量是否大于等于目标数据的大小。具体地,首先第一判断模块341判断剩余容量是否大于或等于目标数据的一帧数据的大小,如剩余容量小于目标数据的一帧数据的大小,则第一判断模块341向等待模块38输出启动信号,等待模块38则等待循环缓冲器内的数据被读取或等待所需数据完全写入到循环缓冲器内。如剩余容量大于或等于目标数据的一帧数据的大小,则第二判断模块判断数据结束指针与目标数据的写入指针之间的容量是否大于或等于目标数据的一帧数据的大小。

[0039] 如数据结束指针与目标数据的写入指针之间的容量小于目标数据的一帧数据的大小,则第二判断模块向指针设置模块输出启动信号。如数据结束指针与目标数据的写入指针之间的容量大于或等于目标数据的一帧数据的大小,则第二判断模块向写入模块输出启动信号。

[0040] 在指针设置模块32接收到启动信号后,则根据启动信号的不同将数据结束指针指向不同的位置。写入模块35在接收到启动信号后,写入模块35根据剩余空间的首地址写入目标数据。具体地,写入模块35用于将剩余空间的首地址发送至解码器36,解码器36解码,将解码后脉冲码编码调制格式的数据即PCM数据写入到剩余空间中。

[0041] 内存管理装置3包括读取模块37,读取模块37用于读取PCM数据,并获取所需读取数据的长度和循环缓冲器内数据的长度。读取模块37向判断模块34输出剩余容量信号,使得判断模块34判断循环缓冲器内的数据的长度是否大于或等于所需读取数据的长度。

[0042] 如循环缓冲器内的数据的长度大于或等于所需读取数据的长度,则拷贝当前数据到内存的其他位置上,再更新PCM数据空间。

[0043] 如循环缓冲器内的数据的长度小于所需读取数据的长度,则拷贝所需读取数据的读取指针和数据结束指针之间的数据,再从循环缓冲器的首地址拷贝所需读取数据的另一部分数据,最后向指针设置模块32输出启动信号,并使数据结束指针指向循环缓冲器的末地址加1的位置上。

[0044] 由上可见,通过改变数据结束指针的指向位置有利于节省CUP运行的资源,同时利用对目标数据的长度和剩余空间的容量大小的判断,解决了循环缓冲器在使用过程中数据长度不够用时的问题。在对于不固定帧长的数据,或者剩余空间不是帧长的整数倍时,通过判断模块去判断剩余空间是否大于一帧数据的长度,不再需要单独开辟内存空间,直接利用循环缓冲器的空间进行处理即可,可以有效地节省内存的占用空间,从而优化运行速度。

[0045] 需要说明的是,本发明采用音频PCM数据进行说明,并不局限在音频PCM数据的方面,只要是该装置需要数据空间进行管理数据输入输出,并且该装置需要连续的空间来存放输出的数据时,本发明均可以解决上述情况,并实现本发明的主要目的,即减少数据拷贝或者空间的开辟,从而提高速度和在空间紧张的情况下的提供一种有效地解决方法。

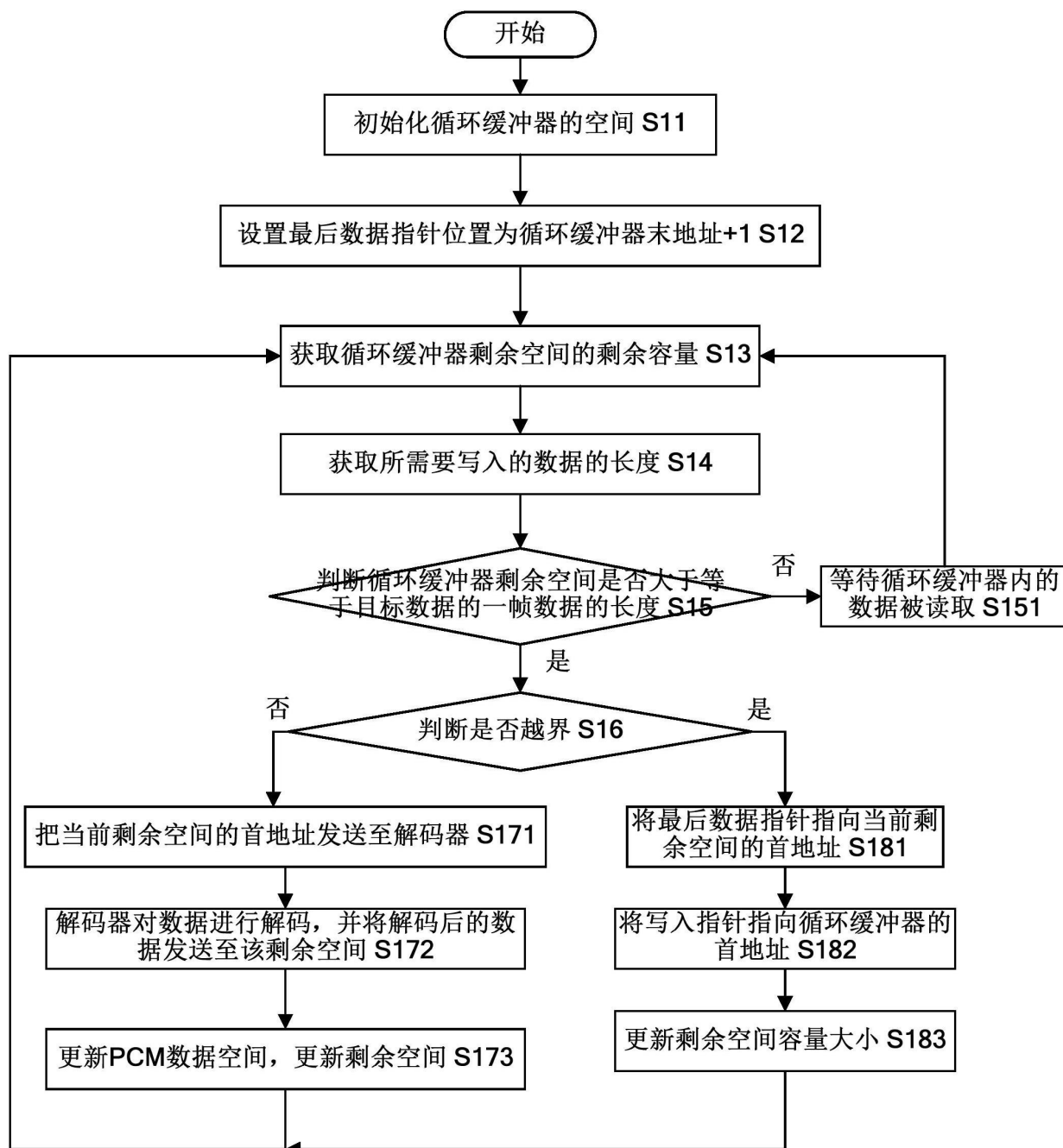


图1

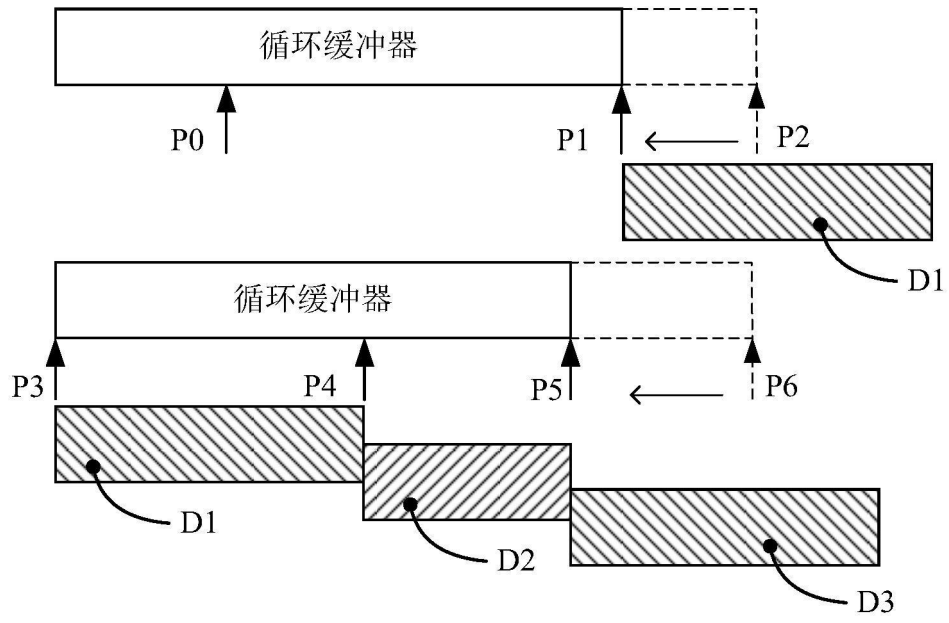


图2

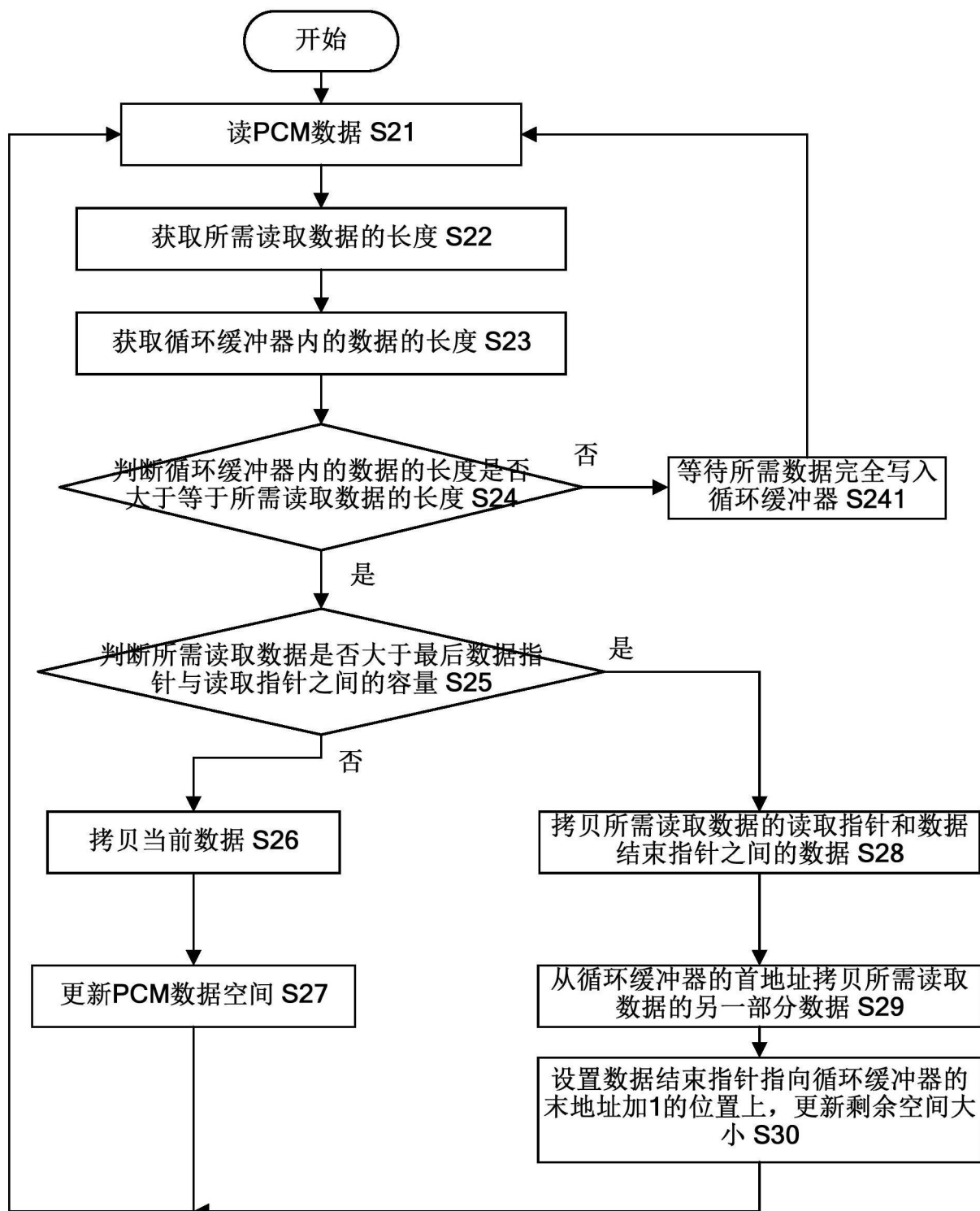


图3

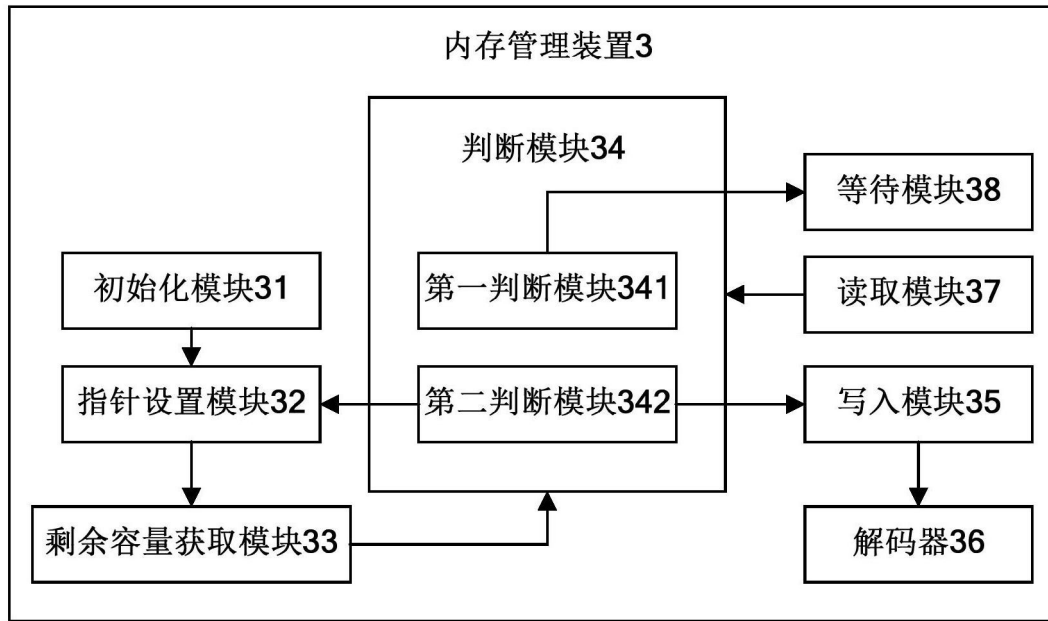


图4