



(12)发明专利

(10)授权公告号 CN 104737512 B

(45)授权公告日 2019.04.19

(21)申请号 201380053506.2

(22)申请日 2013.10.11

(65)同一申请的已公布的文献号

申请公布号 CN 104737512 A

(43)申请公布日 2015.06.24

(30)优先权数据

10-2012-0113052 2012.10.11 KR

(85)PCT国际申请进入国家阶段日

2015.04.13

(86)PCT国际申请的申请数据

PCT/KR2013/009116 2013.10.11

(87)PCT国际申请的公布数据

WO2014/058276 EN 2014.04.17

(73)专利权人 三星电子株式会社

地址 韩国京畿道

(72)发明人 柳诚烈 朴勍模 宋在涓

(74)专利代理机构 北京市柳沈律师事务所
11105

代理人 邵亚丽 刘虹

(51)Int.Cl.

H04L 29/02(2006.01)

(56)对比文件

US 2011255558 A1, 2011.10.20,

CN 101606354 A, 2009.12.16,

US 2008175148 A1, 2008.07.24,

审查员 侯艳兰

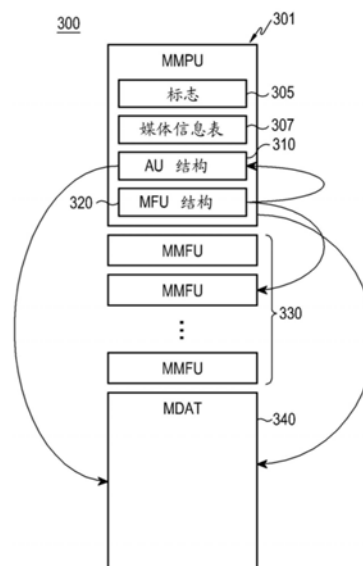
权利要求书1页 说明书13页 附图5页

(54)发明名称

用于在混合网络中传送和接收多媒体数据的装置和方法

(57)摘要

提供一种在能够同时提供广播和因特网通信的混合网络中传送媒体数据的方法。该方法包括：生成媒体处理单元(MPU)以处理媒体数据；以及传送所生成的MPU；其中，所述MPU包括：包括关于所述MPU的信息的MPEG媒体传输(MMT)处理单元(MMPU)、包括从所述媒体数据生成的一个或多个媒体数据片段的媒体数据盒(MDAT)、以及包括关于每个媒体数据片段的的信息的MMFU盒。



1. 一种用于在混合网络中发送媒体数据的方法,该方法包括:

生成媒体处理单元,其包括媒体数据部分和信息部分,所述媒体数据部分包括所述媒体数据的至少一个片段,而所述信息部分包括关于所述媒体处理单元的信息以及关于所述媒体数据的至少一个片段的信息;以及

发送所述媒体处理单元,

其中所述信息部分包括指示所述媒体处理单元是否包括与初始生成的媒体处理单元的片段结构对应的所有片段的第一信息。

2. 如权利要求1所述的方法,其中,

所述信息部分包括指示所述媒体处理单元是否被分段为所述媒体数据的至少两个片段的第二信息。

3. 如权利要求1所述的方法,其中,所述信息部分包括指示所述媒体处理单元中的所述媒体数据的至少一个片段是定时数据还是非定时数据的第三信息。

4. 如权利要求1所述的方法,其中,

所述信息部分包括顺序号、偏移信息和长度信息,所述顺序号指示所述媒体数据的至少一个片段在所述媒体处理单元之内的排序次序,所述偏移信息指示所述媒体数据的至少一个片段在所述媒体处理单元之内的位置,并且所述长度信息指示所述媒体数据的至少一个片段的长度。

5. 如权利要求4所述的方法,其中,所述媒体数据的至少一个片段是根据所述顺序号、偏移信息和长度信息来放置的。

用于在混合网络中传送和接收多媒体数据的装置和方法

技术领域

[0001] 本公开涉及一种用于在混合网络中传送和接收多媒体数据的装置和方法。更具体地,本公开涉及一种当MPEG媒体传输(MMT)不是所支持的数据类型时用于再现MMT数据的装置和方法。

背景技术

[0002] 最近,在多媒体服务中,已开发了:其中同时连接广播网络和通信网络的混合网络,以及提供包括多媒体数据、应用和文件的混合内容的MPEG媒体传输(MMT)技术。

[0003] MMT流以媒体处理单元(MPU)为单位被划分并发送,而且媒体再现设备也以MPU为单位来接收和再现该MMT流。当MMT按MPU来划分时,MMT流并不是无条件地按预定时间或预定大小来划分,而是按照如下单位来划分:该单位足以描述相应MPU中包括的媒体的结构并足以执行相应MMT流需要的操作。

发明内容

[0004] 技术问题

[0005] 相比之下,在根据相关技术的MPU配置中,不支持MMT的再现装置不能接收在MPU中包括的媒体数据。此外,根据相关技术的MPU配置,对于在预定时间处再现的定时的媒体数据(timed media data)的传送是高效率的,而对于非定时的媒体数据的传送是低效率的。

[0006] 以上信息被提供作为背景信息,仅仅用于帮助理解本发明。关于上述任何内容就本公开而言是否可适用为相关技术,没有作出判定,也没有作出断言。

[0007] 技术方案

[0008] 本公开的各方面要解决至少上述问题和/或缺点并提供至少下述优点。相应地,本公开的一方面是为了提供一种媒体处理单元(MPU)结构,其中不支持MPEG媒体传输(MMT)的媒体再现装置能够接收和再现MMT媒体数据。

[0009] 本公开的另一方面是为了提供能够传送非定时的媒体数据的、用于传送多媒体数据的装置和方法。

[0010] 本公开的又一方面提供一种用于传送多媒体数据的装置和方法,其中,即使网络节点删除或修改媒体片段单元(MFU),接收侧也不需要再次接收对MPU的更新,并且该装置和方法能够考虑到覆盖不同访问单元(AU)的MFU的关系而以MFU为单位来删除MPU。

[0011] 根据本公开的一方面,提供一种在能够同时提供广播和因特网通信的混合网络中传送媒体数据的方法。该方法包括:生成MPU以处理媒体数据;以及传送所生成的MPU;其中,所述MPU包括:包括关于所述MPU的信息的MPEG媒体传输(MMT)处理单元(MMPU)盒(box)、包括从所述媒体数据生成的一个或多个媒体数据片段的媒体数据盒(MDAT)、以及包括关于每个媒体数据片段的信息的MMFU盒。

[0012] 根据本公开的一方面,该MMPU盒包括:指示媒体数据是与定时数据还是与非定时数据相对应的第一标志(flag)、指示MPU是否包括MFU的第二标志、以及指示MPU是与原始

MPU还是与已修改的MPU相对应的第三标志。

[0013] 根据本公开的一方面,MPU盒包括具有关于AU的信息的AU结构盒,而且MPU包括具有多个媒体数据片段的一个或多个AU。

[0014] 根据本公开的一方面,MPU盒包括MFU结构盒,该MFU结构盒包括至少一个媒体数据片段以及关于与该媒体数据片段相对应的MMFU盒的信息。

[0015] 根据本公开的一方面,MFU结构盒包括关于媒体数据片段和AU之间的关系的的信息,而且MPU包括具有多个媒体数据片段的一个或多个AU。

[0016] 根据本公开的一方面,MFU结构盒包括以下各项中的至少一者:媒体数据片段之间的引用信息、优先级信息、以及时间和空间的混合层信息,而且这些信息是考虑到属于不同AU的媒体数据片段而确定的。

[0017] 根据本公开的一方面,关于媒体数据片段的信息包括以下各项中的至少一者:每个媒体数据片段的顺序号、大小和偏移。

[0018] 根据本公开的一方面,所述第三标记被修改为足以指示:媒体数据片段是由被删除实体修改的MPU,而且在该MPU中包括的媒体数据片段中的至少一个被删除。

[0019] 根据本公开的另一方面,提供一种在能够同时提供广播和因特网通信的混合网络中传送媒体数据的装置。该装置包括:生成单元,其生成MPU以处理媒体数据;以及,传送单元,其传送所生成的MPU;其中,所述MPU包括:包括关于所述MPU的信息的MMPU盒、包括从所述媒体数据生成的一个或多个媒体数据片段的MDAT盒、以及包括关于每个媒体数据片段的信息的MMFU盒。

[0020] 根据本公开的另一方面,提供一种在能够同时提供广播和因特网通信的混合网络中接收媒体数据的方法。该方法包括:接收MPU以处理媒体数据;以及通过使用所生成的MPU来恢复所述媒体数据;其中,所述MPU包括:包括关于所述MPU的信息的MMPU盒、包括从所述媒体数据生成的一个或多个媒体数据片段的MDAT盒、以及包括关于每个媒体数据片段的信息的MMFU盒。

[0021] 根据本公开的另一方面,提供一种在能够同时提供广播和因特网通信的混合网络中接收媒体数据的装置。该装置包括:接收单元,其接收MPU以处理媒体数据;以及,恢复单元,其通过使用所生成的MPU来恢复所述媒体数据;其中,所述MPU包括:包括关于所述MPU的信息的MMPU盒、包括从所述媒体数据生成的一个或多个媒体数据片段的MDAT盒、以及包括关于每个媒体数据片段的信息的MMFU盒。

[0022] 根据本公开的另一方面,提供一种用于传送多媒体数据的装置,其能够通过使用MPU来传送传统媒体格式的数据。该装置包括具有与传统再现装置兼容的MPU结构的非定时的媒体数据。此外,即使MFU被删除,也不需要执行对MPU的更新。此外,能够考虑到覆盖不同AU的MFU的关系而以MFU为单位来删除MPU。

[0023] 对于本领域的技术人员来说,从以下结合附图进行的本公开的各种实施例的详细描述中本公开的其它方面、优点和显著特性将变得明显。

附图说明

[0024] 通过以下结合附图的描述,本公开的某些实施例的上述和其他方面、特征和优点将会更加清楚,附图中:

- [0025] 图1是示出根据相关技术的媒体处理单元(MPU)的结构的框图;
- [0026] 图2A和图2B是示出根据本公开的实施例的、在网络节点中移除媒体片段单元(MFU)的过程的视图;
- [0027] 图3是示出根据本公开的实施例的MPU的结构的框图。
- [0028] 图4A和图4B是示出根据本公开的实施例的、在网络节点或接收端接收MPU之后修改或恢复MPU的过程的视图;以及
- [0029] 图5是示出根据本公开的实施例的传送多媒体数据的方法的流程图。
- [0030] 贯穿附图,相同的附图参考标号将被理解为指代相同的部件、组件、和结构。

具体实施方式

[0031] 提供以下参照附图的描述来帮助全面理解权利要求及其等效物所限定的本发明的各种实施例。以上描述包括各种具体细节来帮助理解,但这些具体细节应被看作仅仅是示例性的。因此,本领域普通技术人员将会认识到,在不脱离本公开的范围和精神的情况下,可以对本文描述的各种实施例进行各种改变和修改。另外,为了清楚和简洁,可能省略对公知功能和结构的描述。

[0032] 在以下描述和权利要求中使用的术语和字词不受限于字面含义,而只是被发明人用来使得能够对于本公开有清楚且一致的理解。从而,本领域技术人员应当清楚,提供以下对本公开的各种实施例的描述只是为了说明,而不是为了限制如所附权利要求及其等同物所限定的本公开。

[0033] 要理解,单数形式“一”、“一个”和“所述”包括复数所指物,除非上下文明确地另有规定。从而,例如,对“一组件表面”的提及包括对一个或多个这样的表面的提及。

[0034] 在本公开的各种实施例的描述中使用的术语将基于MPEG媒体传输(MMT)标准来描述。

[0035] 媒体片段单元(MFU)为独立于特定媒体编解码器而且包括媒体解码器能够独立地消费的编码的媒体数据的通用容器。MFU等于或小于访问单元(AU),而且包括能够被传送层所利用的信息。

[0036] AU是能够将定时(timing)信息作为属性的最小数据实体。对于不具有针对解码和与解码相关联的呈现而指定的定时信息的、编码的媒体数据,AU不被定义。

[0037] 媒体处理单元(MPU)是独立于任何特定媒体编解码器而且包含一个或多个AU以及另外的与传送和消费有关的信息的通用容器。对于非定时数据,MPU包括部分的没有AU边界的数据。MPU是能够完整地、独立地处理的、编码的媒体数据单位。在这个上下文中,处理意指封装入MMT包或用于传送的分组。

[0038] 非定时数据是在非特定时间被消费的数据元素。该非定时数据可以具有其中数据被执行的、有效的定时范围。

[0039] 定时数据被定义为与用于解码和呈现的特定时间相关联的数据元素。

[0040] 在下文中,本公开的各种实施例将被详细地描述。

[0041] 图1是示出根据相关技术的MPU结构的框图。

[0042] 参照图1,MPU 100包括MMT处理单元(MMPU) 105和一个或多个MFU 130、140和150。MMPU包括关于MPU结构的信息,而且还包括盒,诸如图1中所示的轨道片段报头(TFHD) 107、

轨道片段运行 (TRUN) 110和子样本信息 (SUBS) 120。相应的盒的详细描述将被省略。各个MFU 130、140和150都包括MMFU和媒体数据盒 (MDAT)。每个MMFU包括关于相应的MFU的结构的信息,并且MDAT包括实际的媒体数据。相比之下,MFU中包括的媒体数据是在该描述中的定时的媒体数据。在非定时的媒体数据的情况下,由于非定时的媒体数据的划分并传送是不必要的,因此非定时的媒体数据在MPU中传送,但不以MFU的形式被划分并传送。

[0043] 此外,虽然未在附图中示出,但AU可以包括多个MFU。例如,若在MPU中存在五个MFU,那么五个MFU当中的头两个MFU被包括在第一AU中,而剩余三个MFU可以被包括在第二AU中。在这种情况下,MMPU包括具有关于AU结构的信息的AU结构。具体地,该AU结构被包括在MMPU中的TRUN 110的盒中。

[0044] 表1示出用于图1的定时的媒体数据的AU的结构。

[0045] 【表1】

```
aligned(8) class TrackRunBox
extend Full Box( 'trun' , version, tf_flags) {
    unsigned int(32) sample_count;
    {
        unsigned int(32) sample_duration;
        if (version == 0) {
            unsigned int(32) sample_composition_time_offset;
        }
[0046]     else {
        signed int(32) sample_composition_time_offset;
    }
    unsigned int(32) decoding_order; //flag=0x1000
    unsigned int(32) subsample_start_id; //flag=0x2000
    unsigned int(32) subsample_end_id; //flag=0x2000
    unsigned int(8) rap_flag; //flag=0x4000
    }[sample_count]
}
```

[0047] 如表1中所指示的、关于AU的信息被包括在TRUN盒110中。TRUN盒110包括相应的定时的媒体数据样本的基本长度以及数目、当各个样本具有不同长度时的样本的长度、样本的再现时间和解码次序、关于相应样本是否为随机访问点 (RAP) 的确定、以及关于相应的MFU的开始和终止编号的信息。

[0048] 然而,在非定时数据的情况下,因为MPU 100可以不包括MFU,故MPU可以被要求包括关于MFU的开始索引和终止索引的信息。例如,在其中MFU包括非定时的媒体数据的情况下,MPU 100不被要求具有AU结构和时间信息。因此,不必要的信息变成开销。根据本公开的各种实施例,AU不被要求包括AU结构和时间信息,然而,MFU可以被要求包括这种信息。根据本公开的各种实施例,关于MPU是否包括MFU的指示应被提供。

[0049] 相比之下,不支持MMT格式的、根据相关技术的定时的媒体数据不能由MMT格式的MFU来封装。相应地,在其中MPU 100包括传统媒体数据的情况下,MPU不被要求包括关于MFU的信息。此外,因为根据相关技术的媒体再现装置不支持MMT格式以至于从所接收的媒体数据中识别不出按照MMT提供的MPU中的盒,因此,根据相关技术的媒体再现装置跳过MMPU盒或MMFU盒。相应地,根据相关技术的媒体再现装置不能访问任何盒。因此,根据相关技术的

媒体再现装置识别不出MMFU中包括的媒体数据。相应地,根据相关技术的MPU结构不能与不支持MMT的、根据相关技术的媒体再现装置兼容。

[0050] 根据本公开的各种实施例,提供一种MPU结构,使该MPU结构能够与不支持MMT的媒体再现装置兼容。根据本公开的各种实施例,MMFU包括MFU的序号、以及媒体数据的大小信息和偏移。

[0051] 如上所述,另一方面,MMT定时媒体数据以MFU为单位被传送并接收。虽然在多个MFU的传送期间一些MFU可能丢失,但某些网络节点可能考虑到信道条件而不传送MFU。例如,在其中信道条件已突然恶化的情况下,若数据传输的速率要降低一半,那么将要发送的全部MFU的一半不被传送。在这种情况下,当任意一个MFU丢失时,接收侧再次接收关于全部MPU的信息。例如,全部媒体数据可以通过更新其中反映或指示MFU的丢失的MPU来恢复。根据本公开的各种实施例,当一些MFU丢失时,尽管不再接收全部MPU,但MPU结构允许全部媒体数据被恢复。为此,根据本公开的各种实施例,相应的MPU能够包括指示原始MPU具有被修改的部分的信息,以便接收侧能够通过相应的信息识别那个部分被修改或删除。

[0052] 根据本公开的各种实施例的MPU的结构将被更详细地描述。

[0053] 根据本公开的各种实施例,MMPU包括以下各项中的至少一者:“标志(flag)”字段、“媒体信息表(media information table)”字段、“AU结构(AU structure)”字段和“MPU结构(MPU structure)”字段。

[0054] 首先,“标志”字段将被描述。根据本公开的各种实施例,“标志”字段包括三个标志。“标志1”指示在相应的MPU中包括的数据是定时数据还是非定时数据。“标志2”指示相应的MPU是否包括MFU。“标志3”指示相应的MPU是原始MPU还是经修改的MPU。

[0055] 标志1将被详细地描述。若标志1为“1(真)”时,那么标志1指示MPU 100具有定时的媒体数据。若标志1为“0(假)”时,那么标志1指示MPU 100具有非定时的媒体数据。本公开的各种实施例不限于此。例如,其相反情况是可能的。此外,若标志1为“1(真)”时,那么标志1指示关于定时的媒体数据的时间信息被提供。若标志1为“0(假)”时,那么标志1指示没有提供时间信息。关于定时的媒体数据的时间信息是指再现定时的媒体数据的时间。

[0056] “标志2”将被详细地描述。若标志2为“1(真)”时,那么标志2指示MPU具有用MFU封装的媒体数据。若标志2为“0(假)”时,那么标志2指示MPU具有不是用MFU封装的媒体数据。本公开的各种实施例不限于此。例如,其相反情况是可能的。作为参考,具有不是用MFU封装的媒体格式的媒体数据可以由,例如,MPEG-传输流(TS)或ISOFF来表达。此外,若标志2为“真”时,那么标志2指示关于MFU结构的信息被提供。若标志2为“假”时,那么标志2指示关于MFU结构的信息没有被提供。以上提及的标志2指示MPU是否包括用MFU封装的媒体数据。换言之,标志2指示MPU是否包括支持MMT的媒体数据。

[0057] 之后,“标志3”将被详细地描述。

[0058] 在下文中,MFU结构将被描述。

[0059] 假设MFU结构独立地存在,不论AU存在或不存在。在其中AU存在并包括MFU的情况下,MFU结构包括关于AU和MFU之间的关系的信息。相比之下,在其中AU不存在并不包括MFU的情况下,MFU结构不包括关于AU和MFU之间的关系的信息。短语“关于AU和MFU之间的关系的信息”是指关于属于每个AU的MFU的数目的信息。相应地,使用映射信息来将AU表与MFU表映射是可能的。

[0060] 此外,在其中MFU包括定时的媒体数据的情况下,因为MFU包括“定时的媒体数据”的类型,所以将MFU中包括的媒体数据的类型指示为预定基本值是可能的而且描述MFU中包括的媒体的类型是可能的。因此,媒体类型能够从媒体信息表中的媒体类型集合中选择。相比之下,在其中MFU包括非定时的媒体数据的情况下,一个MFU可以具有彼此不同的媒体类型。因此,指示媒体信息表的哪个媒体类型与每个MFU映射是可能的。

[0061] 此外,MFU结构提供关于MPU中的各种MFU之间的关联关系的信息。在图1所示的传统MPU结构中包括的MFU具有:示出不同MFU彼此引用的频率的“依赖性计数(dependency_counter)”字段、示出MFU当中的优先级的“优先级(priority)”字段、以及示出关于空间和时间的混合层的信息的“多层信息(multilayer information)”字段。信息的引用范围被限制于属于一个AU的不同MFU。例如,假设AU 1包括MFU 1、MFU 2和MFU 3,并且AU 2包括MFU 3和MFU 4,在根据相关技术的MFU中,“依赖性计数(dependency_counter)”、“优先级(priority)”、“多层信息(multilayer information)”中的每个仅仅示出属于AU 1的MFU 1、MFU 2和MFU 3之间的引用关系。类似地,“依赖性计数(dependency_counter)”、“优先级(priority)”、“多层信息(multilayer information)”中的每个仅仅示出属于AU 2的MFU 3和MFU 4之间的引用关系。然而,属于彼此不同的AU 1和AU 2的MFU之间的引用关系可能存在。根据相关技术,由于关于AU中包括的MFU的信息被限制于在同一AU中包括MFU,若属于不同AU的MFU之间的引用关系存在,那么当仅考虑到AU中包括的MFU之间的关系而示出所述信息时,网络节点可能不能有效地删除或修改MFU。

[0062] 根据本公开的各种实施例,“依赖性计数(dependency_counter)”、“优先级(priority)”、“多层信息(multilayer information)”是考虑到在不同AU中存在的MFU之间的关系而被示出的。因此,网络节点具有处于混乱状态的信道条件。当预计某个MFU被移除时,移除MFU的效率可能增加。MFU的移除将参照图2A和图2B来描述。

[0063] 图2A和图2B是示出根据本公开的实施例的、在网络节点中移除MFU的过程的视图。

[0064] 例如,图2A和图2B示出根据本公开的实施例的、当MFU属于不同AU时在网络节点中移除MFU的过程。

[0065] 参照图2A,相互引用在位于不同AU的MFU之间被保持(maintain),其中相互引用通过实线来示出。参照图2B,相互引用在位于不同AU中的MFU之间没有被保持,其中相互引用通过虚线来示出。

[0066] 如图2A中所示的,不同的MFU 211、213和215在不同的AU 201、203和205之间彼此引用。假设第二AU 203的第二MFU 213在网络节点中被删除,被删除的第二MFU 213引用的、第一AU 201中的第一MFU 211和第三AU 205中的第三MFU 215可以被删除。

[0067] 相比之下,如图2B中所示的,由于第二AU 223中的第二MFU 233没有引用第一AU 221中的第一MFU 231和第三AU 225中的第三MFU 235,因此,尽管第二MFU 233被删除,第一MFU 231和第三MFU 235也可以不被删除。

[0068] 在如图2A中所示的、其中在不同AU中的MFU之间的相互引用被保持的状态中,若MFU中的“依赖性计数(dependency_counter)”、“优先级(priority)”、“多层信息(multilayer information)”是考虑到关于一个AU中的MFU之间的关系的的信息而示出的,那么网络节点选择将被删除的MFU,而不考虑到引用不同AU中的MFU的相应MFU。相应地,因为当网络节点删除该MFU时考虑一个AU中包括的MFU的优先级等等,因此有效地删除MFU是很

难的。

[0069] 例如,若在一个AU中存在五个MFU并且网络节点考虑到这五个MFU之间的关系来确定将被删除的MFU,那么具有最低优先级的MFU将从五个MFU中被删除。然而,若被删除的MFU引用不同AU中包括的MFU,那么相应MFU的优先级可能被改变。因此,为了指示MFU的优先级,该相应MFU与不同AU以及该相应MFU所属的AU的关系应被考虑。在采用不同的多条的信息的情况下,若相应信息是考虑到不同AU而被指示的,那么当MFU在网络节点中被删除时,通过使用这几条信息来有效地删除MFU是可能的。

[0070] 本公开的各种实施例提出考虑到以上描述的、在表2中指示的AU结构和在表3中指示的MFU结构。

[0071] 【表2】

```
aligned(8) class AccessUnitStructureBox extends Box( 'ausb' ){
    unsigned int(32) default_au_duration;
    unsigned int(32) number_of_au;
{
    unsigned int(32) au_duration; //optional
    if (version == 0) {
        unsigned int(32) au_composition_time_offset;
[0072]     }
        else {
            signed int(32) au_composition_time_offset;
        }
    unsigned int(32) decoding_order; //optional
    unsigned int(8) rap_flag;          //optional
    }[ number_of_au ]
}
```

[0073] 表2的AU结构被包括在根据本公开的各种实施例的MPU中。

[0074] 图3是示出根据本公开的实施例的MPU的结构的框图。

[0075] 参照图3,在MPU中包括AU结构由参考标号310来表示。

[0076] 【表3】

```

aligned(8) class MediaFragmentUnitStructureBox extends Box( 'mfus' ){
    unsigned int(32) default_media_type_id;
    unsigned int(32) item_count;
    {
        if(has_timed_media){
            unsigned int(32) number_of_mfu_in_au;
        }
[0077]    unsigned int(32) media_type_id; //optional
        unsigned int(32) size;
        unsigned int(8) dependency_counter;
        unsigned int(8) priority;
        unsigned int(32) multilayerInfor;
    }[ item_count ]
}

```

[0078] 表3的MFU结构被包括在根据本公开的各种实施例的MPU中。该MPU中包括MFU结构由图3中的参考标号320来表示。具体地,如表3中所提供的,“依赖性计数”、“优先级”和“多层信息”引用的MFU的范围,扩展到属于不同AU以及该相应MFU所属的当前AU的MFU。此外,表3的MFU结构包括:MFU在AU中的顺序号(number_of_mfu_in_au)、以及关于位于MDAT中的媒体数据的实际大小的信息。此外,虽然未在表3中示出,但媒体数据可以包括偏移,以便在必要时告知网络节点它在MDAT中的位置。根据本公开的各种实施例,媒体数据实际上被布置在MDAT中,不同于根据相关技术的MPU结构。

[0079] 在下文中,保持与不支持MMT的、根据相关技术的媒体格式再现装置的兼容的方案将被描述。

[0080] 对于传统MPU结构,MMFU在其中包括定时的媒体数据。然而,由于不支持MMT的、根据相关技术的媒体格式再现装置不能识别以上描述的MMFU,故在MPU中不存在定时的媒体数据能够被访问的盒。相比之下,关于使用基于组块(chunk)和样本来编址(addressing)的、根据相关技术的媒体格式,存在如下问题:定时的媒体数据被放置在不连续的空间中。相应地,根据相关技术的媒体格式再现装置与MMT不兼容。

[0081] 根据本公开的各种实施例,MFU的顺序号和媒体数据盒的大小、以及偏移,在必要时,被另外描述,而且实际媒体数据的片段被顺序地放置在MDAT中。因此,网络设备或存储设备读取MPU然后以MFU为单位删除该MPU中包括的媒体数据,由此有效地传送或存储MPU。例如,若接收由服务提供者初始生成的MPU的网络节点,被假设为在很差的网络条件中删除媒体数据的特定片段并传送MPU。根据本公开的各种实施例,网络节点删除MPU中包括的多个MMFU盒当中的某个MMFU盒,并通过使用相应MMFU指示的偏移和大小信息来删除MDAT中的相应的媒体数据,从而减小MPU的大小。

[0082] 即使如上所述一些媒体数据在网络节点中被删除,接收侧也不再次接收其中反映被删除的媒体数据的MPU。更准确地说,接收侧可能被要求识别特定媒体数据从相应的MPU被删除。在下文中,其中接收侧识别媒体数据被删除的方案将被描述。

[0083] 在包括接收侧的媒体再现装置的特定网络节点接收和读出MPU的结果中,若该特

定网络节点搜索MPU中的MMFU并发现一些MMFU丢失了,那么媒体格式再现装置能够识别出所接收的MPU不是传送侧传送的原始MPU,而是被修改的MPU。然而,因为在媒体格式再现装置读出所有MMFU之后,该媒体格式再现装置确定所接收的MPU不是原始MPU,故对于该确定是需要时间的。根据本公开的各种实施例,为了快速地识别网络实体接收的MPU是否为原始MPU,“标志3”字段被包括在MPU中。

[0084] 若“标志3”为“1(真)”,那么标志3指示相应的MPU为完全原始的MPU并包括所有原始生成的MFU。若“标志3”为“0(假)”,那么标志3指示相应的MPU为“被修改的MPU”,其中一些MFU被删除或修改。本公开的各种实施例不限于此。例如,其相反情况是可能的。例如,因为标志3指示相应的MPU为被修改的MPU,故相应的MPU可以在之后用原始MPU来代替,以便再现和存储相应的MPU。

[0085] “标志3”可以被放置在MPU前面的固定的偏移的位置。此外,在网络实体删除MFU之后,“标志3”的值被从“1(真)”改变为“0(假)”,以便指示相应的MPU被修改。

[0086] 以上描述的根据本公开的各种实施例的MMFU结构在表4中作为示例被指示。

[0087] 【表4】

```
aligned(8) class MediaFragmentUnitBox
    extends FullBox( 'mmfu' , version, 0){
    if (version == 1){
        unsigned int(32) mfu_sequence_number;
    }
[0088]     else {unsigned int(16) mfu_sequence_number;
    }
    unsigned int(32) size;
    unsigned int(32) offset;
}
```

[0089] 应理解,表4的MMFU被包括在根据本公开的各种实施例的MPU中,并且包括MFU的序号、大小和偏移。作为参考,MPU中包括的MMFU由图3中的参考标号330来表示,如之后描述的。

[0090] MPU 301包括标志305、媒体信息表307、AU结构310和MFU结构320,而且通过使用它们具有与该MPU有关的信息。此外,MPU 300包括至少一个MMFU 330,而且每个MMFU包括关于每个媒体数据片段的信息。MDAT 340包括一条或多条媒体数据(如一个或多个媒体数据片段),而且各个媒体数据片段按照顺序号的次序被顺序地排列,并具有MMFU中包括的偏移和大小。

[0091] 根据本公开的各种实施例,标志305可以包括以上所述的“标志1”、“标志2”和“标志3”。标志的内容可以与以上提及的描述相同。

[0092] 媒体信息表307是其中MFU具有的各种媒体类型被预先设置的表,而且当前MFU中包括的媒体类型可以通过该表来指示。

[0093] AU结构310可以与之前表2中指示的那个相同。AU结构具有如下结构:其中关于AU和MFU的信息从图1中示出的AU结构中排除。

[0094] AU结构320可以与之前表3中指示的那个相同。如上所述,根据本公开的各种实施例,MFU结构320的“依赖性计数”、“优先级”和“多层信息”引用的MFU的范围,扩展到属于不同AU以及属于包括该相应MFU的当前AU的MFU。

[0095] 相比之下,若MPU 300具有AU结构310,那么AU结构310包括关于该相应AU中包括的媒体数据的信息。此外,若MFU 300具有MFU结构320,那么该MFU结构320提供关于AU和MFU之间的关系的信息。

[0096] 此外,在其中特定MMFU被删除的情况下,接收MPU的诸如媒体再现装置的实体能够通过使用该MFU结构320,来识别被删除的MMFU和从MDAT 340中删除的媒体数据的偏移和大小。例如,因为MFU结构320包括每个MMFU的标识符和大小,以及关于MMFU 330在MDAT 340中的位置的偏移信息,因此,通过使该信息来识别出被删除的MMFU和从MDAT中删除的媒体数据的偏移和大小是可能的。

[0097] 图4A和图4B是示出根据本公开的实施例的、在网络节点或接收端接收MPU之后修改或恢复MPU的过程的视图。

[0098] 参照图4,三个MPU是基于如之前图3中所示的根据本公开的各种实施例的MPU结构的。因此,因为这些MPU的详细结构已在之前被描述,故详细结构的描述将被省略。

[0099] 原始MPU 400指示由内容提供者所生成的初始MPU。假设原始MPU 400包括MMPU 411、四个MMFU 412,以及分别与四个MMFU相对应的四条媒体数据413。因为原始MPU 400包括初始传输所需的所有MMFU以及分别与这些MMFU相对应的媒体数据,故原始MPU 400被称为“原始MPU”并且“is_complete (是完整的)”的值变成“1”。“is_complete”是可以等同于之前描述的“标志3”的标志,并被命名为直观地表达目的或功能。

[0100] 被修改的MPU 410意指:能够修改MPU的诸如网络节点的网络实体从原始MPU 400删除或修改一些MFU,从而传送该MPU或有效地存储该MPU。因为一些MMFU和相应的媒体数据从MPU中被删除或修改,故被删除或被修改的MPU 410被称为“被修改的MPU”并且“is_complete”被设置为“0”。“is_complete”是等同于之前描述的“标志3”的标志。此时,根据需要修改原始MPU 400并删除MMFU和媒体数据的实体可以将“is_complete”的值从“1”修改为“0”。例如,当网络实体接收到原始MPU 400时,“is_complete”的值为“1”。然而,因为相应的实体修改了MPU,所以“is_complete”的值被设置为“0”以便指示被修改的MPU。

[0101] 被恢复的MPU 420意指:存储所接收的媒体格式再现装置或终端恢复所接收的MPU,以便生成“被恢复的MPU”。接收“is_complete”的值为“0”的MPU的终端能够通过使用MMFU中包括的媒体数据的大小和偏移,来确定被删除的媒体数据的大小和偏移。

[0102] 接着,相应媒体数据从MDAT被删除的位置,被填充具有与被删除媒体数据的大小相同大小的“虚拟(dummy)”数据。该虚拟数据是没有特定含义的数据,并且具有如下属性:即使解码器解码虚拟数据,解码错误也不会发生。

[0103] 若被删除的MPU通过使用以上所述的虚拟数据来恢复,那么被恢复的MPU 420被生成。该MPU 420不完全等同于原始MPU,但被称为“被恢复的MPU”,因为该数据位于如下的位置处:在该位置指示包括不支持所述MPU和MMT的传统媒体格式的信息。具有传统媒体格式的信息可以包括作为示例的、ISO 14496-12的MOOV盒、MOOF盒等等。

[0104] 参照图4B,示出MPU(诸如,例如,根据本公开的实施例的图4A中所示的MPU)中数据的位置的视图被提供。

[0105] 原始MPU 430包括MMPU 431、四个MMFU 432,以及分别与四个MMFU相对应的四条媒体数据433。原始MPU 430中的实心箭头435指示初始媒体数据的位置。

[0106] 在被修改的MPU 440中,id=3的MMFU 441被删除。因此,与id=3的MMFU 441相对应的媒体数据从MDAT被删除。与id=1和id=2的MMFU相对应的媒体数据存在于:与该媒体数据在原始MPU 430中的位置相同的位置。两个实心箭头443指示:存在于相应位置处的媒体数据中没有错误。因为由第三虚线箭头445从上侧指示的媒体数据是与id=4的实际MMFU相对应的媒体数据,故该媒体数据具有与该媒体数据在原始MPU中的位置不同的位置。两个虚线箭头447指示:因为媒体数据存在于与该媒体数据在原始MPU 430中的位置不同的位置处,故存在错误。

[0107] 根据本公开的各种实施例,当媒体再现设备或终端通过使以上所述的虚拟数据来生成被恢复的MPU 450时,所生成的数据存在于与该数据在原始MPU中的位置相同的位置处。因此,因为被恢复的MPU被确保与原始MPU一致,故确保被恢复的MPU与传统媒体格式的兼容性是可能的。

[0108] 在下文中,在本公开的各种实施例中使用的各种类型的媒体将被描述。

[0109] 根据本公开的各种实施例,所使用的媒体类型提供适用于定时的媒体数据和非定时的媒体数据二者的MIME类型、媒体初始化信息和基本信息。

[0110] 该定时的媒体数据和非定时的媒体数据被提供有一个或多个媒体类型表,并且根据每个MIME类型提供初始化相应的媒体数据所需要的适合的信息。

[0111] 定时的媒体数据的代表性初始化信息包括诸如序列参数集合和画面参数集合的参数集合。

[0112] 再现MPU的再现装置开始再现在MMT资源中连续的MPU中的特定的一个,在定时的媒体数据的情况下根据媒体类型通过读取参数集合来初始化解码器;并且将定时的媒体数据应用于该解码器从而完成MPU的再现。作为参考,MMT资源是指一个或多个MPU的集合。

[0113] 文件可以是非定时的媒体数据的代表性示例。用于文件的初始化的信息可以包括诸如文件的名称、大小、循环冗余校验等的信息。因为该文件被提供为被映射到一个MFU上而且某一文件的位置的信息在MFU结构中被描述,因此,当MPU中存在多个文件时,再现设备能够通过映射信息和该文件的位置来提取所期望的文件。

[0114] 以上提及的媒体类型表的示例在表5中被指示,而且在该媒体类型表中MMPU在表6中被指示。

[0115] 【表5】

```
aligned(8) class MediaTypeInformationBox extends Box( 'mtib' ){
    unsigned int(32) number_of_media;
    {
        unsigned int(32) mime_type;
[0116]         unsigned int(32) length;
        bit(8*length) initializationData;
    }[ number_of_media ]
}
```

[0117] 表5中的媒体类型表被包括在MMPU中并由图3中所示的参考标号307来表示。相应

的媒体类型表提供关于MPU中媒体的类型的信息。例如,媒体类型表提供关于相应媒体使用哪种编解码和压缩方法的信息。

[0118] 【表6】

```
aligned(8) class AsseltIdentifierBox extends Box( 'asid' ){
    unsigned int(32) asset_id_scheme;
    unsigned int(32) asset_id_length;
    unsigned int(8) asset_id_value[asset_id_length];
}
aligned(8) class MMTProcessingUnitBox
    extends FullBox( 'mmpu' , version, 0){
    unsigned int(1) has_mfu;
    unsigned int(1) is_complete;
    unsigned int(1) has_timed_media;
    unsigned int(5) reserved;
```

[0119]

```
AssetIdentifierBox( );
unsigned int(32) mpu_sequence_number;

MediaTypeInfoBox( );

    if(has_timed_media){
        AccessUnitStructureBox( );
    }
    if(has_mfu){
        MediaFragmentUnitStructureBox( );
    }
}
```

[0120] 在表6中,根据本公开的实施例的、资源(asset)信息、asid(资源id)信息和MPU盒被描述。

[0121] 在表6中,如以上所述的,“has timed media(具有定时的媒体)”是等同于“标志1”的标志,并指示MPU是定时的或非定时的媒体数据。此外,“has MFU(具有MFU)”相应于“标志2”并指示MPU是否具有MFU结构。“is complete(是完整的)”相应于“标志3”并指示MPU是“original flag(原始标记)”还是具有被删除的MFU的“modified flag(被修改的标记)”。

[0122] 图5是示出根据本公开的实施例的传送多媒体数据的方法的流程图。

[0123] 参照图5,在操作500处,媒体数据传送装置生成媒体数据的MFU。

[0124] 之后,在操作510处,媒体数据传送装置生成包括被用于再现媒体数据的信息的MPU。接着,媒体数据传送装置传送所生成的MPU。

[0125] 根据本公开的各种实施例,该MPU具有诸如图3中所示的结构的MPU结构。

[0126] 然而,本领域技术人员将理解的是,用于接收媒体数据的装置和方法可以相应于

用于传送媒体数据的装置和方法而被实现。因此,用于接收媒体数据的装置和方法的描述将被省略。

[0127] 虽然已参考本公开的各种实施例示出和描述了本公开,但本领域技术人员应理解,在不脱离如权利要求及其等同物所限定的本公开的精神和范围的情况下,可对本公开进行形式和细节上的各种改变。

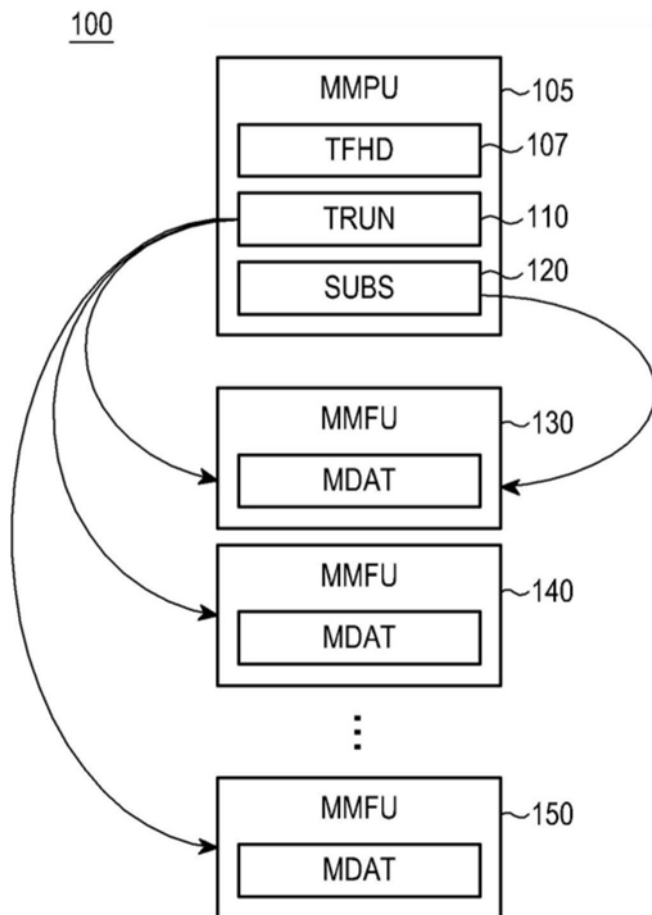


图1

200

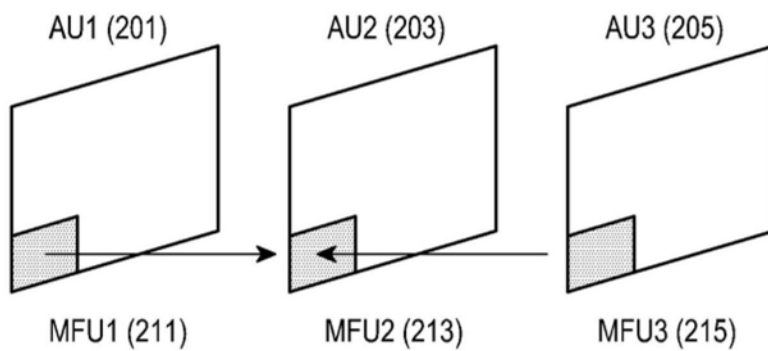


图2a

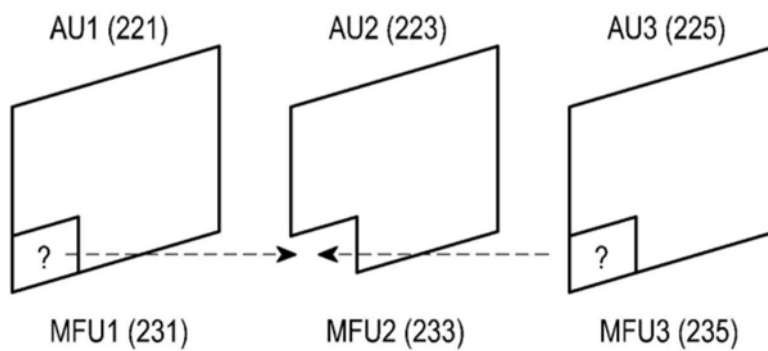


图2b

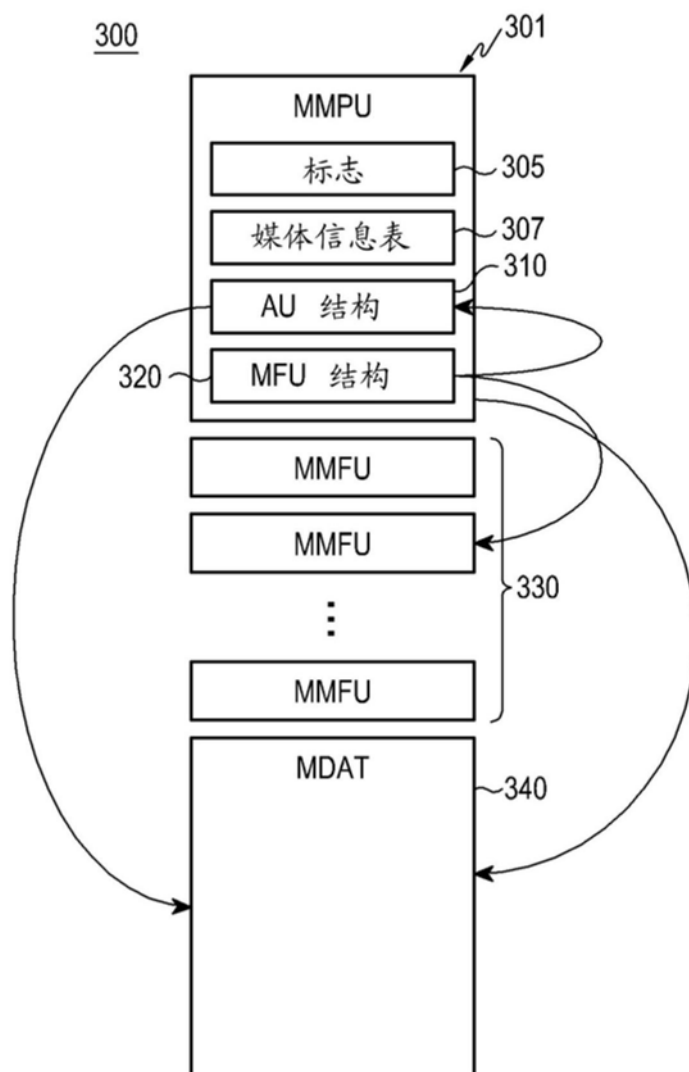


图3

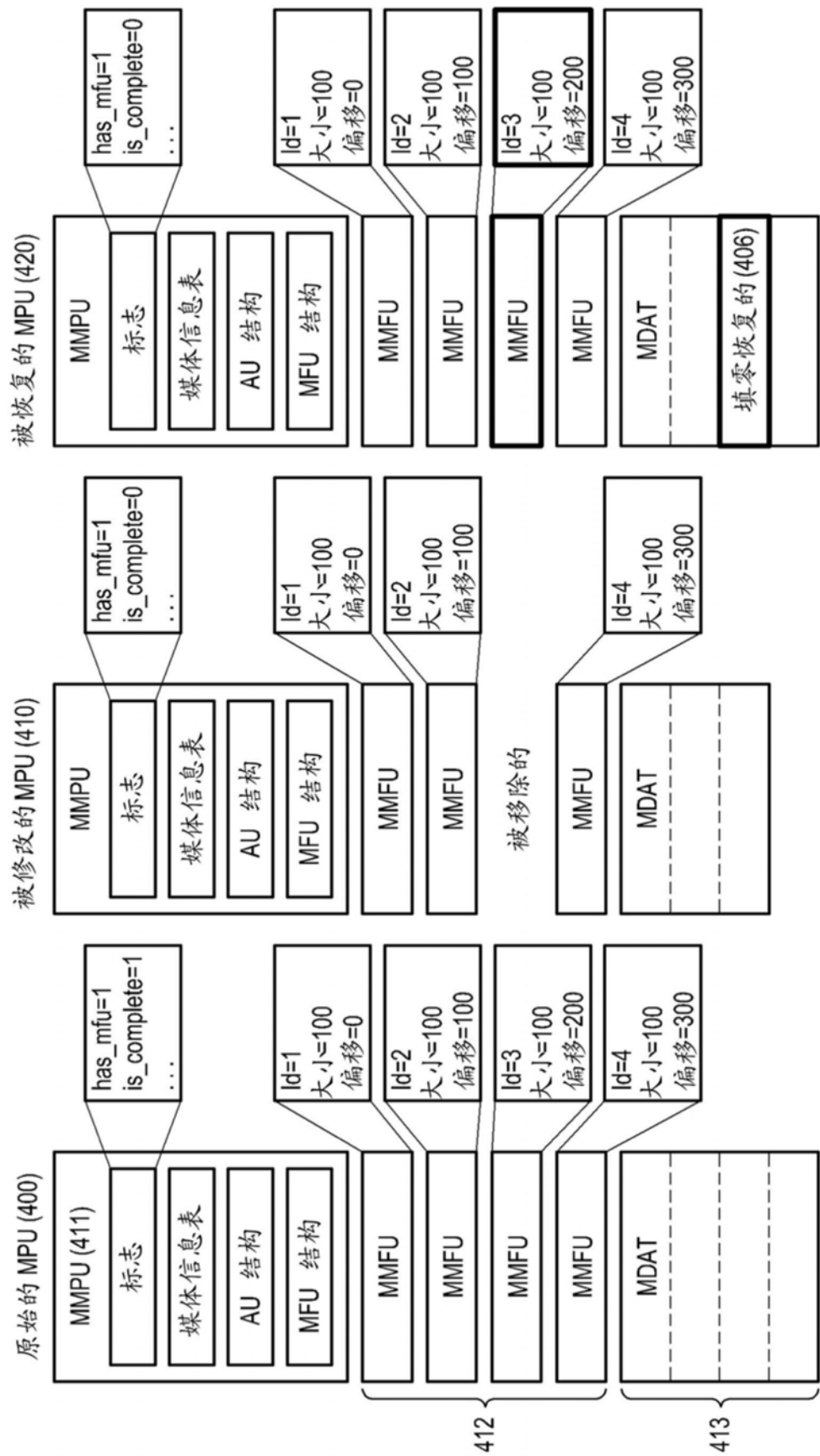


图4a

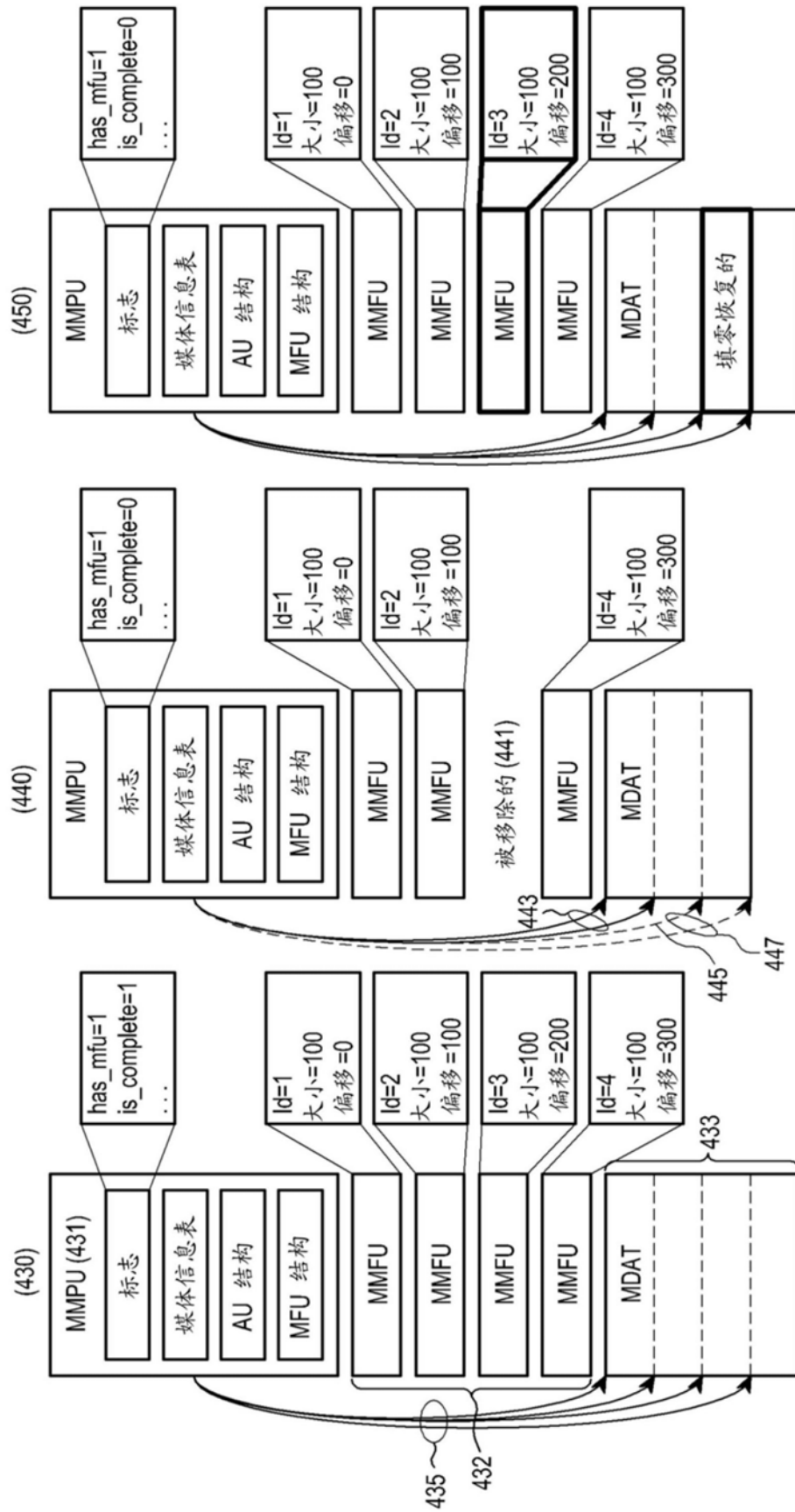


图4b

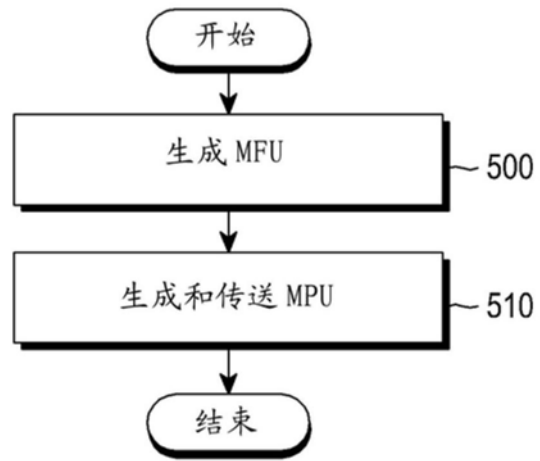


图5