

一种适用于HEVC标准中帧内预测的参考像素的硬件片上存储方法

申请号：[201410590925.6](#)

申请日：2014-10-29

申请(专利权)人 [复旦大学](#)

地址 200433 上海市杨浦区邯郸路220号

发明(设计)人 [范益波](#) [黄磊磊](#) [刘聪](#) [程魏](#) [曾晓洋](#)

主分类号 [H04N19/423\(2014.01\)I](#)

分类号 [H04N19/423\(2014.01\)I](#) [H04N19/503\(2014.01\)I](#)

公开(公告)号 104363455A

公开(公告)日 2015-02-18

专利代理机构 [上海正旦专利代理有限公司](#) 31200

代理人 [陆飞](#) [盛志范](#)



(12) 发明专利申请

(10) 申请公布号 CN 104363455 A

(43) 申请公布日 2015. 02. 18

(21) 申请号 201410590925. 6

(22) 申请日 2014. 10. 29

(71) 申请人 复旦大学

地址 200433 上海市杨浦区邯郸路 220 号

(72) 发明人 范益波 黄磊磊 刘聪 程魏

曾晓洋

(74) 专利代理机构 上海正旦专利代理有限公司

31200

代理人 陆飞 盛志范

(51) Int. Cl.

H04N 19/423 (2014. 01)

H04N 19/503 (2014. 01)

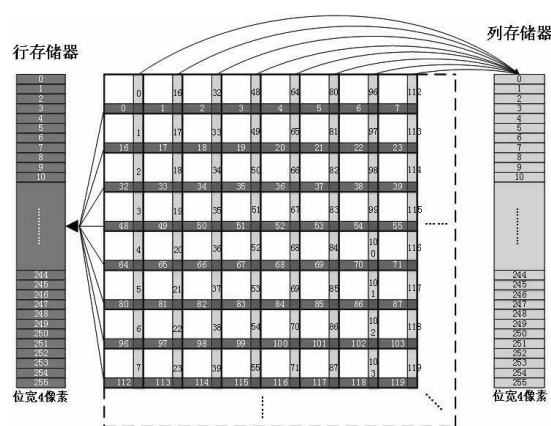
权利要求书1页 说明书3页 附图5页

(54) 发明名称

一种适用于 HEVC 标准中帧内预测的参考像素的硬件片上存储方法

(57) 摘要

本发明属于数字视频技术领域，具体为一种适用于 HEVC 标准中帧内预测的参考像素的片上存储方法。在 HEVC 中，帧内预测是基于块的对象执行的。假设当前编码的最大单位为一个 64×64 块，一个 64×64 块中共有 256 个 4×4 块，每个 4×4 块中共有 7 个预测像素作为参考像素使用，至少需要 1792 个像素的存储空间。本发明将这些参考像素分开存放在两个存储器中：行存储器用来存储所有右上、上方和左上的参考像素，共 1024 个；列存储器用来存储所有左上、左方和左下的参考像素，同样为 1024 个。其存取地址根据对应的 4×4 块在当前 64×64 块内的位置决定。本发明可以在有效地提高参考像素的存取能力，减小处理时间，从而实现高清视频的实时编码。



1. 一种适用于 HEVC 标准中帧内预测的参考像素的硬件片上存储方法,假设当前编码的最大单位为一个 64×64 块,一个 64×64 块中共有 256 个 4×4 块,每个 4×4 块中共有 7 个预测像素将作为参考像素使用,因此至少需要 1792 个像素的存储空间;其中,右上和上方的参考像素共 768 个,左上的参考像素共 256 个,左方和左下的参考像素共 768 个;其特征在于:

将这些参考像素分开存放在两个存储器中,其中,行存储器用来存储所有右上、上方和左上的参考像素,共 $768+256$ 个,即 1024 个;列存储器用来存储所有左上、左方和左下的参考像素,同样为 1024 个;每个存储的位宽设定为 4 个像素,所需的深度为 256;其存取地址根据对应的 4×4 块在当前 64×64 块内的位置决定。

2. 根据权利要求 1 所述的适用于 HEVC 标准中帧内预测的参考像素的硬件片上存储方法,其特征在于:

读取时,所有右上、上方和左上参考像素的存取基地址以当前块左上角 4×4 块的横坐标减一为低位,纵坐标减一为高位;所有左上、左方和左下参考像素的存取基地址以当前块左上角 4×4 块的纵坐标减一为低位,横坐标减一为高位:

$$Row_Addr_{rd} = ((y_{rd} - 1) \ll 4) + (x_{rd} - 1) \quad (1)$$

$$Col_Addr_{rd} = ((x_{rd} - 1) \ll 4) + (y_{rd} - 1) \quad (2)$$

此处, Row_Addr_{rd} 是所需读取的参考像素在行存储器中的基地址, Col_Addr_{rd} 是所需读取的参考像素在列存储器中的基地址, x_{rd} 是当前块左上 4×4 块的横坐标, y_{rd} 是当前块左上 4×4 块的纵坐标, x_{rd} 和 y_{rd} 都以 4×4 块作为计数单位;对于剩下参考像素的读取地址在此基地址的基础上依次加 1;

写入时,所有右上、上方和左上参考像素的存取基地址以当前块右下角 4×4 块的横坐标为低位,纵坐标为高位;所有左上、左方和左下参考像素的存取基地址以当前块左上角 4×4 块的纵坐标为低位,横坐标为高位:

$$Row_Addr_{wr} = (y_{wr} \ll 4) + x_{wr} \quad (3)$$

$$Col_Addr_{wr} = (x_{wr} \ll 4) + y_{wr} \quad (4)$$

此处, Row_Addr_{wr} 是所需写入的参考像素在行存储器中的基地址, Col_Addr_{wr} 是所需写入的参考像素在列存储器中的基地址, x_{wr} 是当前块右下 4×4 块的横坐标, y_{wr} 是当前块右下 4×4 块的纵坐标, x_{wr} 和 y_{wr} 都以 4×4 块作为计数单位;对于剩下参考像素的读取地址在此基地址的基础上依次减 1。

3. 根据权利要求 2 所述的适用于 HEVC 标准中帧内预测参考像素的硬件片上存储方法,其特征在于通过对于行列存储器的同时访问,同时得到右上、上方和左上的参考像素与左上、左方和左下的参考像素。

一种适用于 HEVC 标准中帧内预测的参考像素的硬件片上存储方法

技术领域

[0001] 本发明属于数字视频技术领域,针对 HEVC 视频编解码标准,具体涉及一种适用于 HEVC 标准中帧内预测的参考像素的硬件片上存储方法。

背景技术

[0002] HEVC (High Efficiency Video Coding)是由国际电信组织(ITU)和运动图像专家组(MPEG)联合成立的组织 JCTVC 提出的下一代视频编解码标准。目标是在相同的视觉效果的前提下,相比于上一代标准,即 H. 264/AVC 标准,压缩率提高一倍。

[0003] 基于 HEVC 的视频编码器,主要由以下模块组成:帧内预测、帧间预测、变换、量化、反量化、反变换、重建、去方块滤波器、自适应样点补偿等。其中,帧内预测将利用同一帧图像内相邻像素之间的相关性,采用合适的方法进行预测,以减小空间冗余度,从而达到压缩的效果。

[0004] 在 HEVC 中,帧内预测是基于块的对象执行的,每个方块由若干个排列成正方形的像素组成。为了得到当前块的预测值,即完成对于当前块的帧内预测,需要该方块右上、上方、左上、左方和左下方紧邻的且已经存在的预测值作为参考,如图 1 左上部分所示。这些预测值是通过执行在对于当前块的预测之前的、对于相应块的预测而得到的,并被称为帧内预测的参考像素。另一方面,当前块所产生的预测值亦可能成为其右上、右方、右下、下方和左下方块的参考像素,如图 1 右下部分所示。由于上述原因,帧内预测的执行需要对参考像素进行频繁地读取和存储,而这成为了 HEVC 实时帧内预测的主要瓶颈之一。

[0005] 为了提高预测的准确性,HEVC 引入了基于四叉树的块结构,具体地,图像处理块的最大单位可以是一个 64×64 块,而该 64×64 块可以被划分成 4 个 32×32 块,每个 32×32 块又可以被划分为 4 个 16×16 块,依次类推直到 4×4 块的层次。可以说,帧内预测的过程就是选择最佳划分的过程,而在这样的遍历过程中,参考像素将被频繁地产生、存储、读取和更新。因此,对于这些参考像素的存储方法将直接影响对于资源的占用和编码的效率。

发明内容

[0006] 本发明的目的在于提出一种可以克服现有技术不足的、适用于 HEVC 标准中帧内预测的参考像素的硬件片上存储方法,以便在有效地提高芯片处理时,提高参考像素的存取能力,从而减小处理时间,高效地实现高清视频的实时编码。

[0007] 假设当前编码的最大单位为一个 64×64 块,那么对于当前的处理单位,所需要存储的参考像素如图 2 所示,其中灰色部分是将作为参考像素使用的预测值,白色部分是其预测值。由于最小块可以是 4×4 块,所有 4×4 块所涉及的参考像素都需要被存储和更新。具体地,一个 64×64 块中共有 256 个 4×4 块,每个 4×4 块中共有 7 个预测像素将作为参考像素使用,因此至少需要 1792 个像素的存储空间。其中,右上和上方的参考像素共 768 个,左上的参考像素共 256 个,左方和左下的参考像素共 768 个。

[0008] 为了降低读取的难度、加快读取的速度,本发明将这些参考像素分开存放在两个存储器中,如图 3 所示。其中,取名为行存储器的存储器用来存储所有右上、上方和左上的参考像素,共 768+256 个,即 1024 个;取名为列存储器的存储器用来存储所有左上、左方和左下的参考像素,同样为 1024 个。将每个存储的位宽设定为 4 个像素,那么所需的深度为 256。其存取地址根据对应的 4×4 块在当前 64×64 块内的位置决定。

[0009] 读取时,具体地,所有右上、上方和左上参考像素的存取基地址以当前块左上角 4×4 块的横坐标(记为 x)减一为低位,纵坐标(记为 y)减一为高位;所有左上、左方和左下参考像素的存取基地址以当前块左上角 4×4 块的纵坐标减一为低位,横坐标减一为高位:

$$Row_Addr_{rd} = ((y_{rd} - 1) \ll 4) + (x_{rd} - 1) \quad (1)$$

$$Col_Addr_{rd} = ((x_{rd} - 1) \ll 4) + (y_{rd} - 1) \quad (2)$$

此处, Row_Addr_{rd} 是所需读取的参考像素在行存储器中的基地址, Col_Addr_{rd} 是所需读取的参考像素在列存储器中的基地址, x_{rd} 是当前块左上 4×4 块的横坐标, y_{rd} 是当前块左上 4×4 块的纵坐标, x_{rd} 和 y_{rd} 都以 4×4 块作为计数单位。对于剩下参考像素的读取地址在此基地址的基础上依次加 1 即可。

[0010] 写入时,具体地,所有右上、上方和左上参考像素的存取基地址以当前块右下角 4×4 块的横坐标(记为 x)为低位,纵坐标(记为 y)为高位;所有左上、左方和左下参考像素的存取基地址以当前块左上角 4×4 块的纵坐标为低位,横坐标为高位:

$$Row_Addr_{wr} = (y_{wr} \ll 4) + x_{wr} \quad (3)$$

$$Col_Addr_{wr} = (x_{wr} \ll 4) + y_{wr} \quad (4)$$

此处, Row_Addr_{wr} 是所需写入的参考像素在行存储器中的基地址, Col_Addr_{wr} 是所需写入的参考像素在列存储器中的基地址, x_{wr} 是当前块右下 4×4 块的横坐标, y_{wr} 是当前块右下 4×4 块的纵坐标, x_{wr} 和 y_{wr} 都以 4×4 块作为计数单位。对于剩下参考像素的读取地址在此基地址的基础上依次减 1 即可。

[0011] 本发明中,可以通过对于行列存储器的同时访问,同时得到右上、上方和左上的参考像素与左上、左方和左下的参考像素。

[0012] 通过本发明的存储方法,任意大小的块的存储方式都是统一的,任意大小的块的读取都是容易的、迅速的。除此之外,任意大小的块的右上、上方和左上的参考像素与左上、左方和左下的参考像素都是能够同时读取的。

[0013] 本发明提出的存储方法,可以大大降低读取的难度,加快读取的速度,从而减小处理时间,实现高清视频的实时编码。

附图说明

[0014] 图 1:HEVC 参考像素说明图。

[0015] 图 2:一个 64×64 块内所需存储的参考像素。

- [0016] 图 3 : 本发明所提出的存储方法。
- [0017] 图 4 : 本存储方法的读示例一。
- [0018] 图 5 : 本存储方法的读示例二。
- [0019] 图 6 : 本存储方法的写示例一。
- [0020] 图 7 : 本存储方法的写示例二。

具体实施方式

[0021] 下面通过实例并结合附图, 进一步具体描述本发明方法。

[0022] 如对于图 4 中, 标记为 A 的 4×4 块而言, 其横坐标 x_{rd} 为 2, 纵坐标 y_{rd} 为 1。那么, 根据公式 (1), 得到所需读取的参考像素在行存储器中的首地址 Row_Addr_{rd} 为 1, 剩余地址为 2、3; 根据公式 (2), 得到所需读取的参考像素在行存储器中的首地址 Col_Addr_{rd} 为 16, 剩余地址为 17、18。

[0023] 又如对图 5 中, 标记为 A 的 8×8 块而言, 其左上角 4×4 块的横坐标 x_{rd} 为 2, 纵坐标 y_{rd} 为 2。那么根据公式 (1), 得到所需参考像素在行存储器中的首地址 Row_Addr_{rd} 为 17, 剩余地址为 18、19、20、21; 根据公式 (2), 得到所需参考像素在行存储器中的首地址 Col_Addr_{rd} 为 17, 剩余地址为 18、19、20、21。

[0024] 如对于图 6 中, 标记为 A 的 4×4 块而言, 其横坐标 x_{rd} 为 2, 纵坐标 y_{rd} 为 1。那么, 根据公式 (3), 得到所需写入的参考像素在行存储器中的首地址 Row_Addr_{wr} 为 18; 根据公式 (4), 得到所需写入的参考像素在行存储器中的首地址 Col_Addr_{wr} 为 33。

[0025] 又如对图 7 中, 标记为 A 的 8×8 块而言, 其右下角 4×4 块的横坐标 x 为 3, 纵坐标为 3。那么根据公式 (1), 得到所需写入的参考像素在行存储器中的首地址 Row_Addr_{wr} 为 51, 剩余地址为 50; 根据公式 (2), 得到所需参考像素在行存储器中的首地址 Col_Addr_{wr} 为 51, 剩余地址为 50。

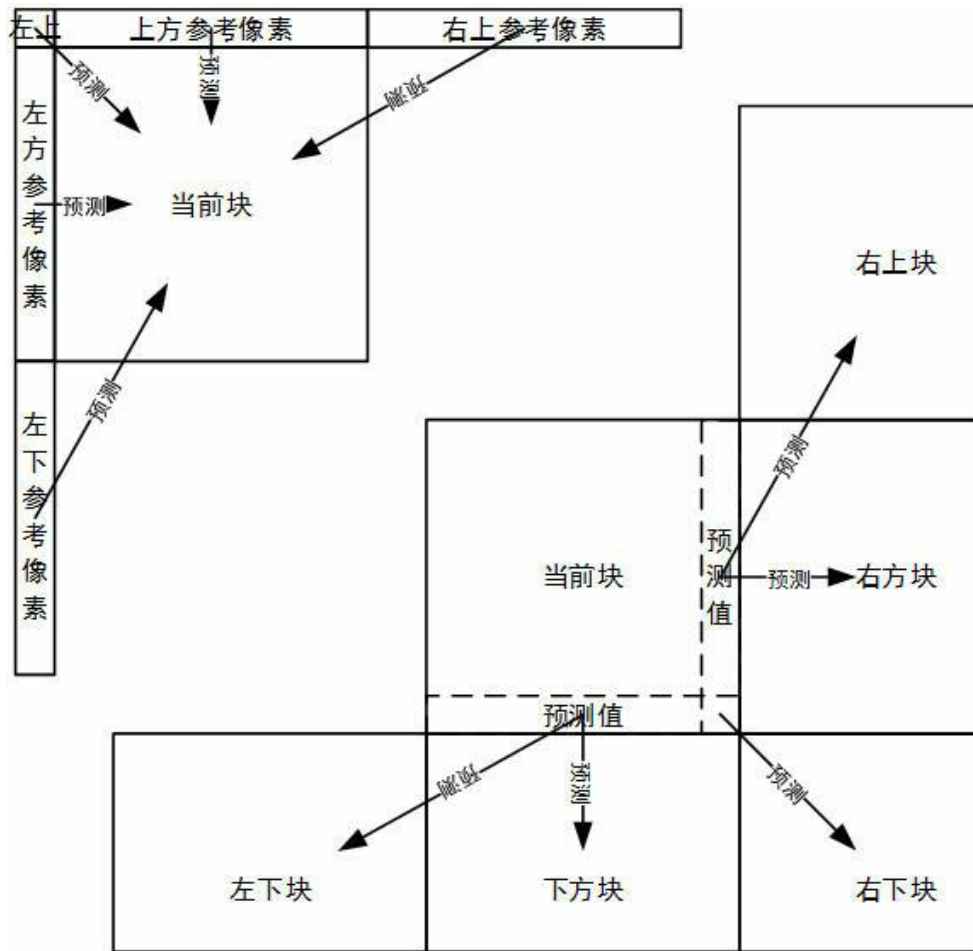


图 1

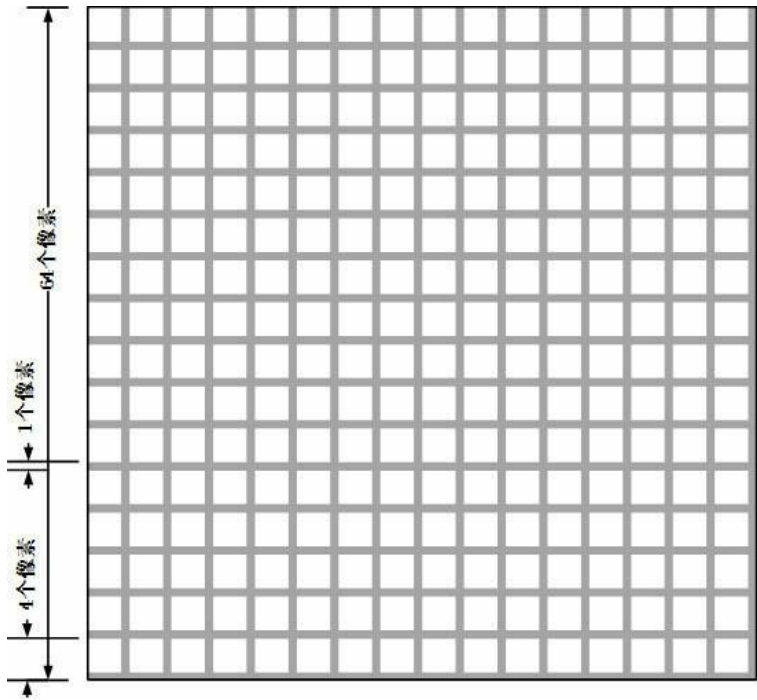


图 2

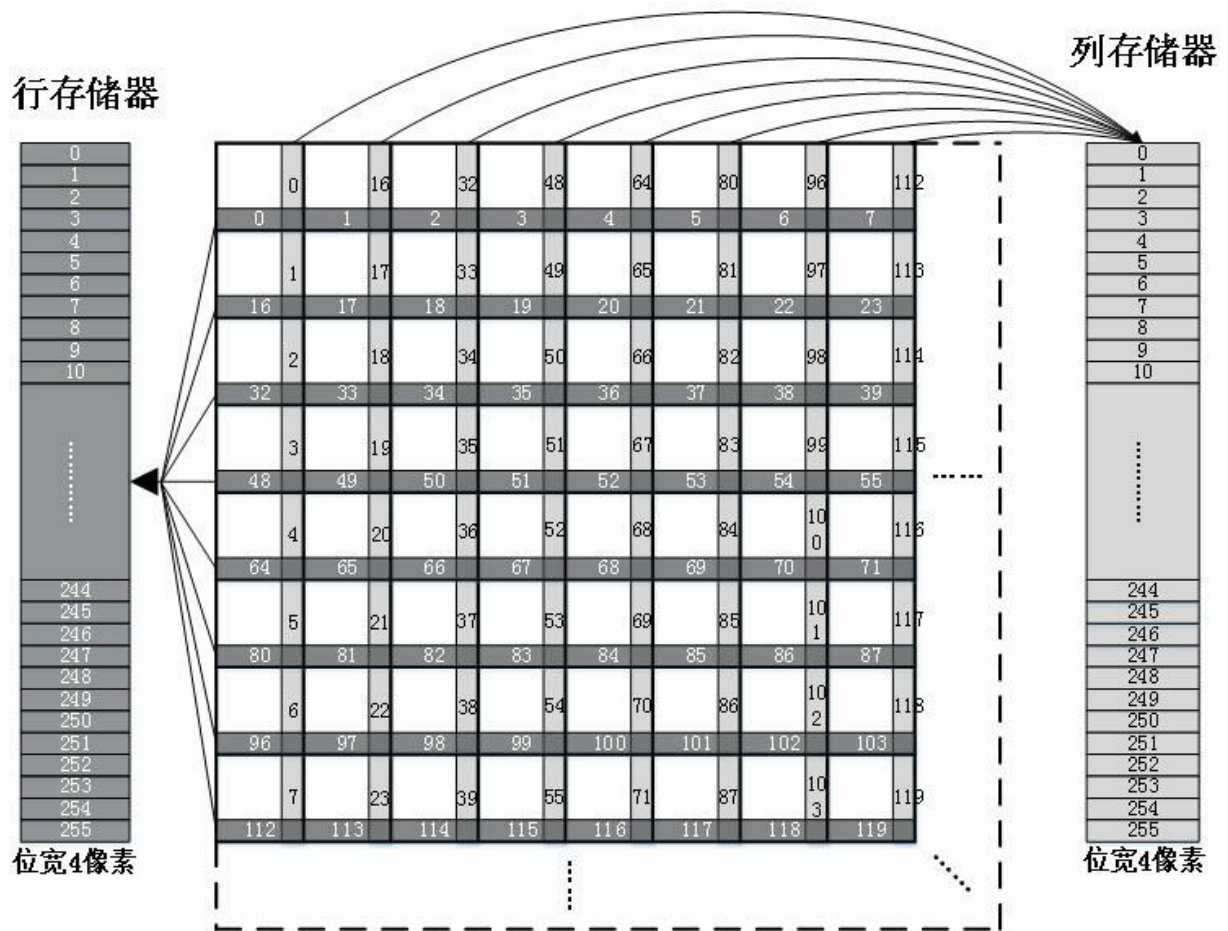


图 3

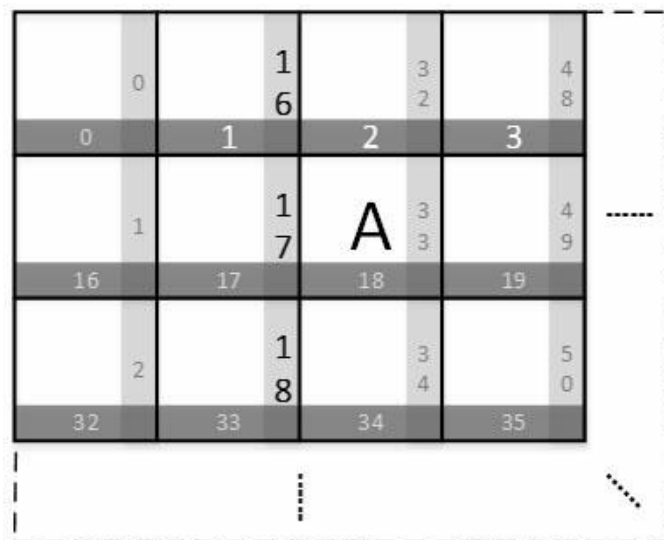


图 4

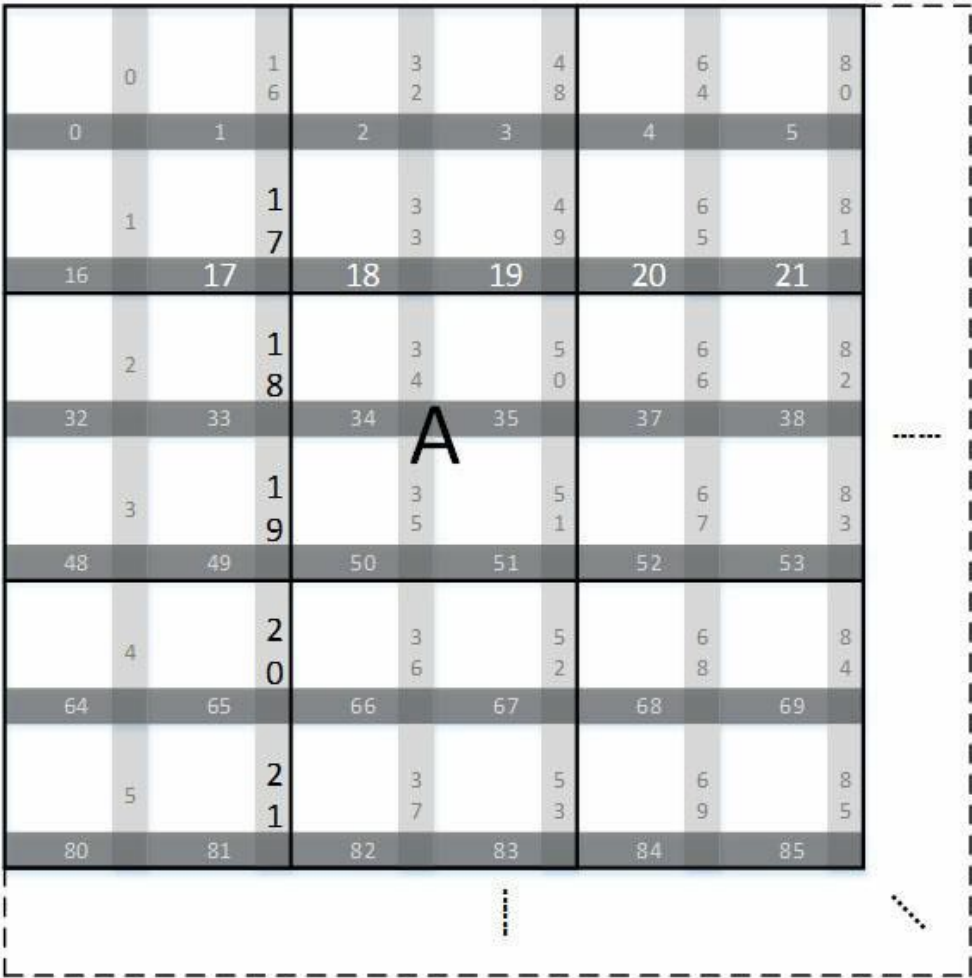


图 5

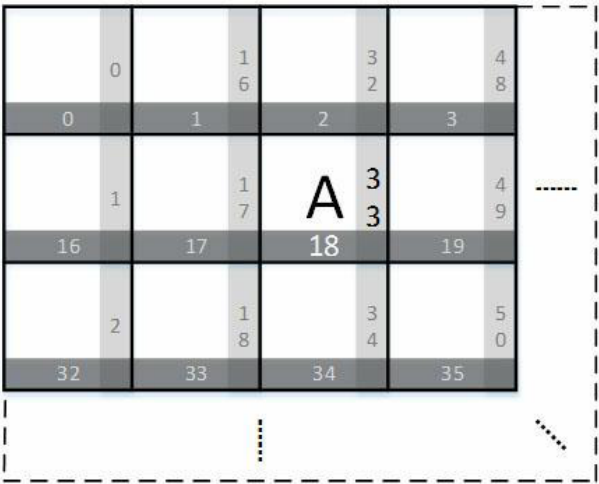


图 6

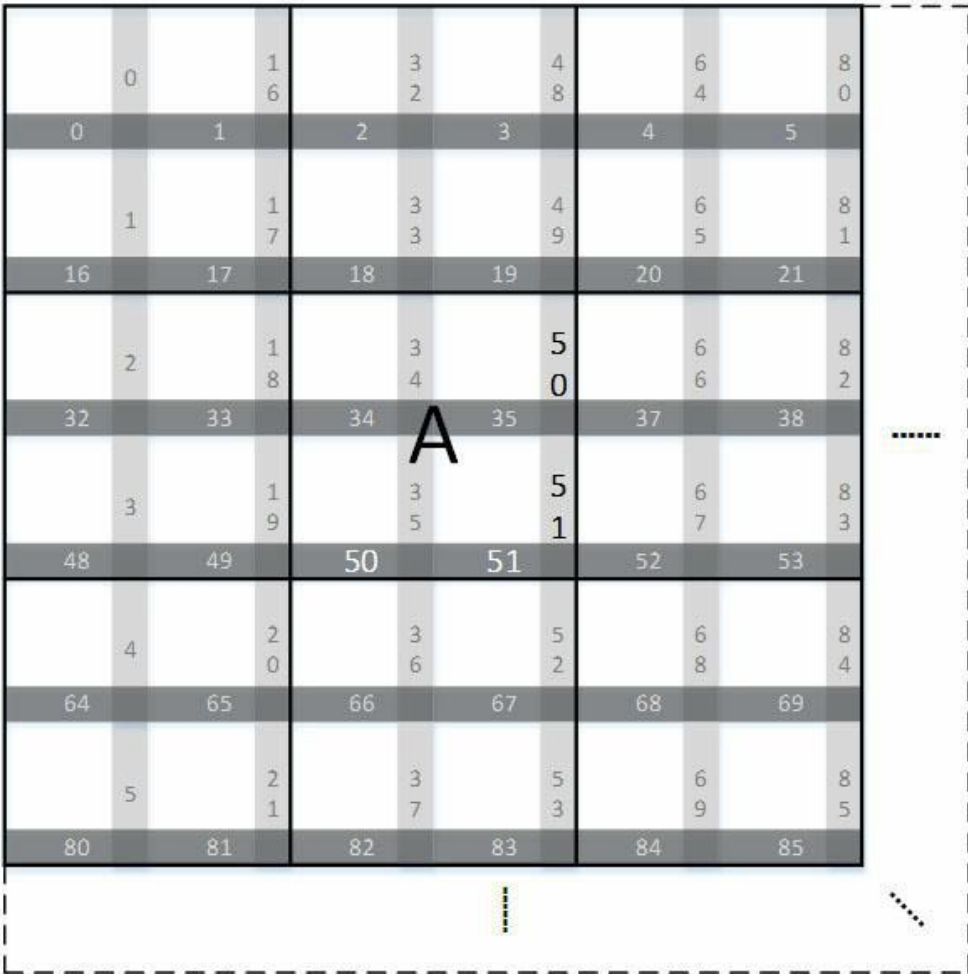


图 7