



(12)发明专利申请

(10)申请公布号 CN 109446110 A

(43)申请公布日 2019.03.08

(21)申请号 201811203554.6

(22)申请日 2018.10.16

(71)申请人 西安邮电大学

地址 710121 陕西省西安市长安区韦郭路
西安邮电大学

(72)发明人 曾泽沧 郝武 黄世远 杜慧敏
张丽果

(74)专利代理机构 西安智邦专利商标代理有限
公司 61211

代理人 郑丽红

(51)Int.Cl.

G06F 12/0811(2016.01)

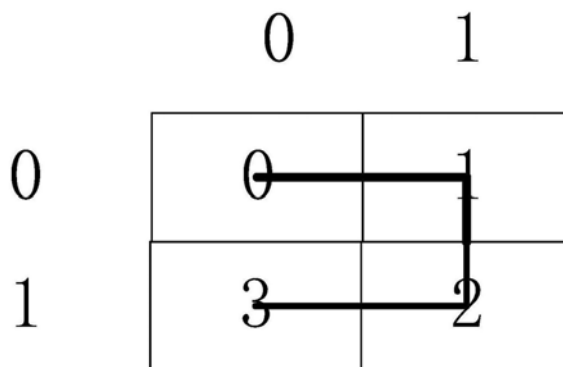
权利要求书2页 说明书4页 附图4页

(54)发明名称

可实现数据地址连续的U型布局tile缓存方法

(57)摘要

本发明涉及数据存储领域,具体涉及一种可实现数据地址连续的U型布局tile缓存方法,解决现有片上寄存器数据存储效率低的问题。本发明的技术解决方案是,一种可实现数据地址连续的U型布局tile缓存方法包括以下步骤:1)选取tile内的四个缓存作为缓存块,四个缓存呈矩形阵列排布;2)以四个缓存中任一缓存为起点,读写顺序沿矩形阵列顺时针或逆时针依次进行。



1. 一种可实现数据地址连续的U型布局tile缓存方法,其特征在於,包括以下步骤:
 - 1) 选取tile内的四个缓存作为缓存块,四个缓存呈矩形阵列排布;
 - 2) 以四个缓存中任一缓存为起点,读写顺序沿矩形阵列顺时针或逆时针依次进行。
2. 根据权利要求1所述的可实现数据地址连续的U型布局tile缓存方法,其特征在於:步骤2)中,以四个缓存中的左上角缓存为起点。
3. 根据权利要求2所述的可实现数据地址连续的U型布局tile缓存方法,其特征在於:步骤2)中,以四个缓存中的左上角缓存为起点,按顺时针依次读写。
4. 一种可实现数据地址连续的U型布局tile缓存方法,其特征在於,包括以下步骤:
 - 1) 将 $2^N \times 2^N$ 的缓存划分为多级缓存块,且一级缓存块包括四个呈矩形阵列排布的缓存,二级缓存块包括四个呈矩形阵列排布的一级缓存块,N级缓存块包括四个呈矩形阵列排布的N-1级缓存块,其中, $N \geq 2$;
 - 2) $2^N \times 2^N$ 缓存的U形读写规则如下:

任一级缓存块读写时,其内部的所有级缓存块须从一级缓存块的任一缓存开始进行读写,

并且,

在每一级缓冲块中,以下一级缓存块中的任一缓存块为起点,按顺时针或逆时针依次读写。
5. 根据权利要求4所述的可实现数据地址连续的U型布局tile缓存方法,其特征在於:步骤2)中,所述的U形读写规则具体包括:

在每一级缓冲块中,以下一级缓存块中的左上角缓存块为起点。
6. 根据权利要求5所述的可实现数据地址连续的U型布局tile缓存方法,其特征在於:步骤2)中,所述的U形读写规则具体包括:

在每一级缓冲块中,以下一级缓存块中的左上角缓存块为起点,按顺时针依次读写。
7. 一种可实现数据地址连续的U型布局tile缓存方法,其特征在於,包括以下步骤:
 - 1) 将 $M \times 2^N \times 2^N$ 的缓存划分为M个 $2^N \times 2^N$ 的缓存;
 - 2) 将 $2^N \times 2^N$ 的缓存划分成多级缓存块,且一级缓存块包括四个呈矩形阵列排布的缓存,二级缓存块包括四个呈矩形阵列排布的一级缓存块,N级缓存块包括四个呈矩形阵列排布的N-1级缓存块,其中, $N \geq 2$;
 - 3) $2^N \times 2^N$ 的缓存的U形读写规则如下:

任一级缓存块读写时,其内部的所有级缓存块必须从一级缓存块的任一缓存开始进行读写,

并且,

在每一级缓冲块中,以下一级缓存块中的任一缓存块为起点,按顺时针或逆时针依次读写;
 - 4) M个 $2^N \times 2^N$ 的缓存依次进行顺序读写。
8. 根据权利要求7所述的可实现数据地址连续的U型布局tile缓存方法,其特征在於:步骤3)中,所述的U形读写规则具体包括:

在每一级缓冲块中,以下一级缓存块中的左上角缓存块为起点。
9. 根据权利要求8所述的可实现数据地址连续的U型布局tile缓存方法,其特征在於:

步骤3)中,所述的U形读写规则具体包括;
在每一级缓冲块中,以下一级缓存块中的左上角缓存块为起点,按顺时针依次读写。

可实现数据地址连续的U型布局tile缓存方法

技术领域

[0001] 本发明涉及数据存储领域,具体涉及一种可实现数据地址连续的U型布局tile缓存方法,该方法为一种U型布局的存储方式和存储地址索引方法。

背景技术

[0002] 随着计算机体系结构的发展,计算机逐渐形成片上寄存器-片上缓存-片外存储器的存储层次结构。片上寄存器最接近处理器,具有数据交换速度最快的特点,同时也具有容量最小的缺陷。因此提升片上寄存器的工作效率,减少片上寄存器的等待时间成为提升数据存储效率的有效方式。

[0003] 在数据存储时,往往需要关注数据的局部性问题。图像、数学矩阵等数据在处理某一区域数据时,相邻地址的数据也有可能在短时间内被用到,产生空间局部性问题。传统的顺序地址访问往往忽视了数据具有的空间局部性,导致片上寄存器产生数据等待,降低了数据的读取效率。

[0004] 随着人们对数据空间局部性认识的提高,相关研究者提出了线性Z型存储,相较于线性存储,线性Z型存储方式在片上寄存器块将每两行变成连续的子地址块,但是在子地址块之间仍会产生较大的地址跨度,故在写回时仍会产生冲突缺失,降低了数据的读取效率。

发明内容

[0005] 本发明的目的是解决现有片上寄存器数据存储效率低的问题,提供了一种可实现数据地址连续的U型布局tile缓存方法。

[0006] 本发明的技术方案是:

[0007] 一种可实现数据地址连续的U型布局tile缓存方法,包括以下步骤:

[0008] 1) 选取tile内的四个缓存作为缓存块,四个缓存呈矩形阵列排布;

[0009] 2) 以四个缓存中任一缓存为起点,读写顺序沿矩形阵列顺时针或逆时针依次进行。

[0010] 进一步地,步骤2)中,以四个缓存中的左上角缓存为起点。

[0011] 进一步地,步骤2)中,以四个缓存中的左上角缓存为起点,按顺时针依次读写。

[0012] 同时,本发明还提供另一种可实现数据地址连续的U型布局tile缓存方法,包括以下步骤:

[0013] 1) 将 $2^N \times 2^N$ 的缓存划分为多级缓存块,且一级缓存块包括四个呈矩形阵列排布的缓存,二级缓存块包括四个呈矩形阵列排布的一级缓存块,依次类推,N级缓存块包括四个呈矩形阵列排布的N-1级缓存块,其中, $N \geq 2$;

[0014] 2) $2^N \times 2^N$ 的缓存的U形读写规则如下;

[0015] 任一级缓存块读写时,其内部的所有级缓存块须从一级缓存块的任一缓存开始进行读写,并且,在每一级缓冲块中,以下一级缓存块中的任一缓存块为起点,按顺时针或逆时针依次读写。

[0016] 进一步地,步骤2)中,所述的U形读写规则具体包括:在每一级缓冲块中,以下一级缓存块中的左上角缓存块为起点。

[0017] 进一步地,步骤2)中,所述的U形读写规则具体包括:在每一级缓冲块中,以下一级缓存块中的左上角缓存块为起点,按顺时针依次读写。

[0018] 此外,本发明还提供的其它可实现数据地址连续的U型布局tile缓存方法,包括以下步骤:

[0019] 1) 将 $M \times 2^N \times 2^N$ 的缓存划分成M个 $2^N \times 2^N$ 的缓存;

[0020] 2) 将 $2^N \times 2^N$ 的缓存划分成多级缓存块,且一级缓存块包括四个呈矩形阵列排布的缓存,二级缓存块包括四个呈矩形阵列排布的一级缓存块,……,N级缓存块包括四个呈矩形阵列排布的N-1级缓存块,其中, $N \geq 2$;

[0021] 3) $2^N \times 2^N$ 的缓存的U形读写规则如下:

[0022] 任一级缓存块读写时,其内部的所有级缓存块必须从一级缓存块的任一缓存开始进行读写,并且,在每一级缓冲块中,以下一级缓存块中的任一缓存块为起点,按顺时针或逆时针依次读写;

[0023] 4) M个 $2^N \times 2^N$ 的缓存依次进行顺序读写。

[0024] 进一步地,步骤3)中,所述的U形读写规则具体包括:在每一级缓冲块中,以下一级缓存块中的左上角缓存块为起点。

[0025] 进一步地,步骤3)中,所述的U形读写规则具体包括:在每一级缓冲块中,以下一级缓存块中的左上角缓存块为起点,按顺时针依次读写。

[0026] 本发明与现有技术相比,具有以下技术效果:

[0027] 1. 本发明可将寄存器内的数据地址设置为完全连续的,减少读取地址时较大的地址跨越,从而减少cache的冲突或缺失问题,提高数据传输效率。

[0028] 2. 本发明方法用于改善在块数据读写时由于地址跨度大而导致的cache冲突或缺失问题,提供一种U型存储布局的片上缓存,使整个缓存内所有数据的地址都是连续的。

附图说明

[0029] 图1为缓存大小为 2×2 的U型布局示意图一;

[0030] 图2为缓存大小为 2×2 的U型布局示意图二;

[0031] 图3为缓存大小为 2×2 的U型布局示意图三;

[0032] 图4为缓存大小为 2×2 的U型布局示意图四;

[0033] 图5为缓存大小为 4×4 的U型布局示意图;

[0034] 图6为缓存大小为 8×8 的U型布局示意图;

[0035] 图7为缓存大小为 16×16 的U型布局示意图;

[0036] 图8为多个二级缓存块进行顺序读写示意图。

具体实施方式

[0037] 以下结合附图和具体实施例对本发明的内容作进一步详细描述:

[0038] 一种可实现数据地址连续的U型布局tile缓存方法,包括以下步骤:

[0039] 1) 选取tile内的任意四个缓存作为缓存块,四个缓存呈矩形阵列排布;

[0040] 2) 以四个缓存中任一缓存为起点,读写顺序沿沿矩形阵列的四个拐角顺时针或逆时针依次进行。

[0041] 2×2 tile缓存如图1所示,读写时顺序为左上,右上,右下,左下。

[0042] 2×2 tile缓存如图2所示,读写时顺序为左上,左下,右下,右上。

[0043] 2×2 tile缓存如图3所示,读写时顺序为左下,左上,右上,右下。

[0044] 2×2 tile缓存如图4所示,读写时顺序为右上,左上,左下,右下。

[0045] 同时,本发明还提供第二种可实现数据地址连续的U型布局tile缓存方法,包括以下步骤:

[0046] 首先将 $2^N \times 2^N$ 缓存划分成4个 $2^{N-1} \times 2^{N-1}$ 的缓存,左上子缓存命名为A,右上子缓存命名为B,右下子缓存命名为C,左下子缓存命名为D;分别将ABCD的子缓存以相同方式进行划分,直至分成 2×2 大小的一级缓存块。

[0047] 这四个子缓存写回顺序为ABCD,分别将ABCD的子缓存以相同方式进行读写,且任一级缓存块读写时,其内部的所有级缓存块须从一级缓存块的任一缓存开始进行读写,并且,在每一级缓存块中,以下一级缓存块中的任一缓存块为起点,按顺时针或逆时针依次读写。

[0048] 同时,本发明还提供第三种可实现数据地址连续的U型布局tile缓存方法,包括以下步骤:

[0049] 对 $2^N \times 2^N$ tile缓存,将其划分成 $2^{N/M}$ 个 $2^N \times 2^M$ 矩形子缓存,其中 $M < N$,将每个 $2^N \times 2^M$ 划分为 $2^{N/M}$ 个 $2^M \times 2^M$ 的子缓存, $2^{N/M}$ 个 $2^M \times 2^M$ 的缓存块按照线性的顺序进行写回,所有 $2^M \times 2^M$ 的缓存以上述第二种方法中的方式进行划分和读写,即左上子缓存命名为A,右上子缓存命名为B,右下子缓存命名为C,左下子缓存命名为D,这四个子缓存写回顺序为ABCD,直至划分成一级缓存块 2×2 的子缓存块,子缓存块的写回顺序也为U型读写。

[0050] U型存储布局的tile缓存,数据缓存地址计算,定义数据读写顺序为Umap,tile内横坐标为tileX,纵坐标为tileY,tileX、tileY和Umap均为二进制数;依据Umap计算tile内对应的横坐标tileX和纵坐标tileY,数据缓存地址为Tileaddr,Umap[m]代表Umap的第m位;用于U型布局的存储地址索引方法,包括以下步骤:

[0051] $\text{TileY} = \{\text{Umap}[7], \text{Umap}[5], \text{Umap}[3], \text{Umap}[1]\}$ 。

[0052] $\text{TileX} = \{(\text{Umap}[6] \wedge \text{Umap}[7]), (\text{Umap}[4] \wedge \text{Umap}[5]), (\text{Umap}[2] \wedge \text{Umap}[3]), (\text{Umap}[0] \wedge \text{Umap}[1])\}$ 。

[0053] $\text{Tileaddr} = \{\text{TileY}, \text{TileX}\}$ 。

[0054] 地址索引的计算方法可依据tile的尺寸进行扩展。

[0055] 定义tile为一个片上数据缓存,其地址为Addr,横坐标为tileX,纵坐标为tileY,存储一个数据,其序号为Umap,则在读写过程中,以U型读写数据包括以下步骤:

[0056] 1) 读写连续地址的数据,第一个数据的Umap为0,此后每读写取一个数据Umap加1;

[0057] 2) 利用Umap计算tileX和tileY的值;

[0058] $\text{TileY} = \{\text{Umap}[7], \text{Umap}[5], \text{Umap}[3], \text{Umap}[1]\}$ 。

[0059] $\text{TileX} = \{(\text{Umap}[6] \wedge \text{Umap}[7]), (\text{Umap}[4] \wedge \text{Umap}[5]), (\text{Umap}[2] \wedge \text{Umap}[3]), (\text{Umap}[0] \wedge \text{Umap}[1])\}$ 。

[0060] 3) 将tileY与tileX组合在一起得到Addr, $\text{Addr} = \{\text{tileY}, \text{tileX}\}$,将数据存入tile

的对应地址中或者将数据从对应地址中取出进行写回。

[0061] 实施例1

[0062] 如图1, tile缓存大小为 2×2 , U型布局读写顺序采用首先读写左上, 其次读写右上, 然后读写右下, 最后读写左下的U型读写顺序。

[0063] 实施例2

[0064] 如图5, tile缓存大小为 4×4 , U型布局读写顺序采用两层U型存储方法, 在单个 2×2 的存储块中采用实施例1中的存储方式; 2×2 的块外采用先存储左上 2×2 块, 其次存储右上 2×2 块, 然后存储右下 2×2 块, 最后存储左下 2×2 块的U型读写方式。

[0065] 实施例3

[0066] 如图6, tile缓存大小为 8×8 , U型布局存储顺序采用三层U型存储方法, 在单个 2×2 的存储块中采用实施例1中的存储方式; 4×4 的存储块采用实施例2中的存储方式; 4×4 存储块外采用先存储左上 4×4 块, 其次存储右上 4×4 块, 然后存储右下 4×4 块, 最后存储左下 4×4 块的存储方式。

[0067] 实施例4

[0068] 如图7, tile缓存大小为 16×16 , U型布局存储顺序采用四层U型存储方法, 在单个 2×2 的存储块中采用实施例1中的存储方式; 4×4 的存储块采用实施例2中的存储方式; 8×8 的存储块采用实施例3中的存储方式; 8×8 存储块外采用先存储左上 8×8 块, 其次存储右上 8×8 块, 然后存储右下 8×8 块, 最后存储左下 8×8 块的存储方式。

[0069] 实施例5

[0070] 如图8所示, 将 16×16 的tile缓存划分成4个 4×16 的矩形块, 这四个矩形块的写回顺序为自上而下, 在 4×16 的矩形块内的写回顺序为每个小矩形的标号, 4×16 的矩形块为4个 4×4 的存储块, 4×4 的存储块采用实施例2中的存储方式, 然后依次对4个 4×4 的存储块进行顺序存储。

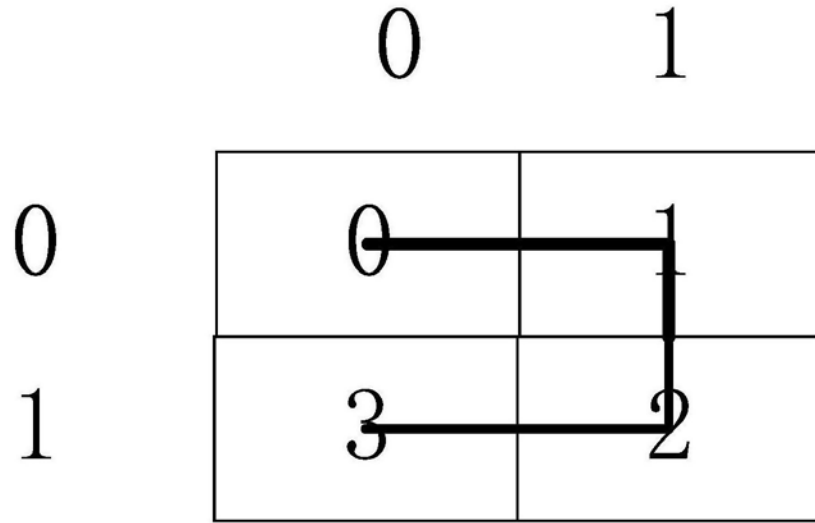


图1

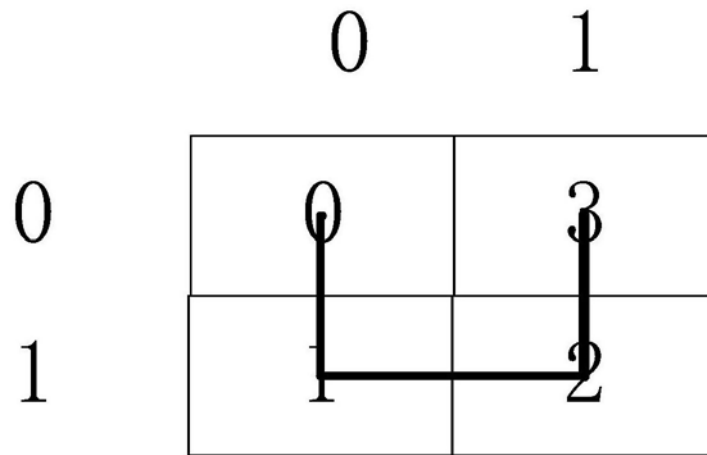


图2

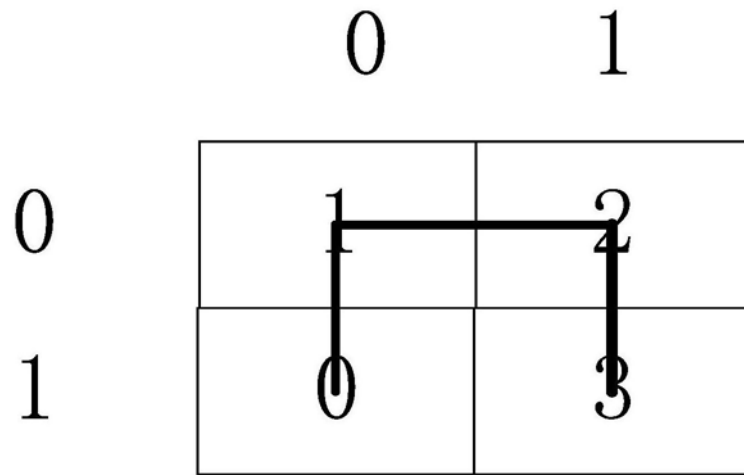


图3

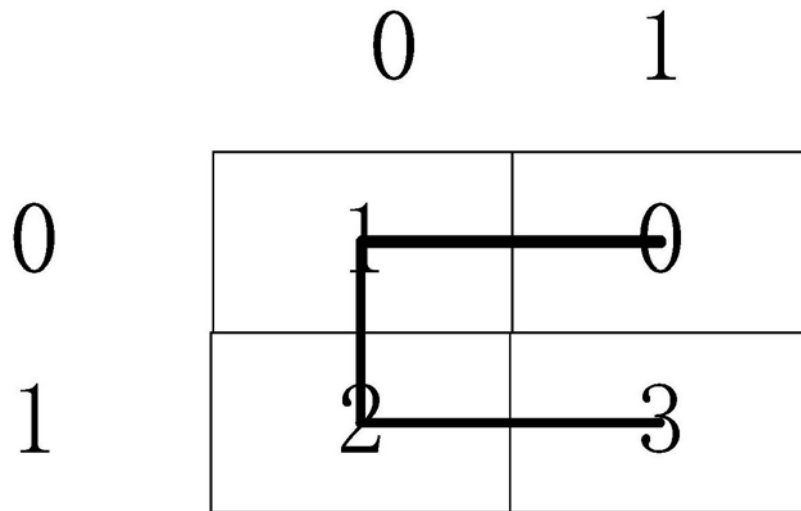


图4

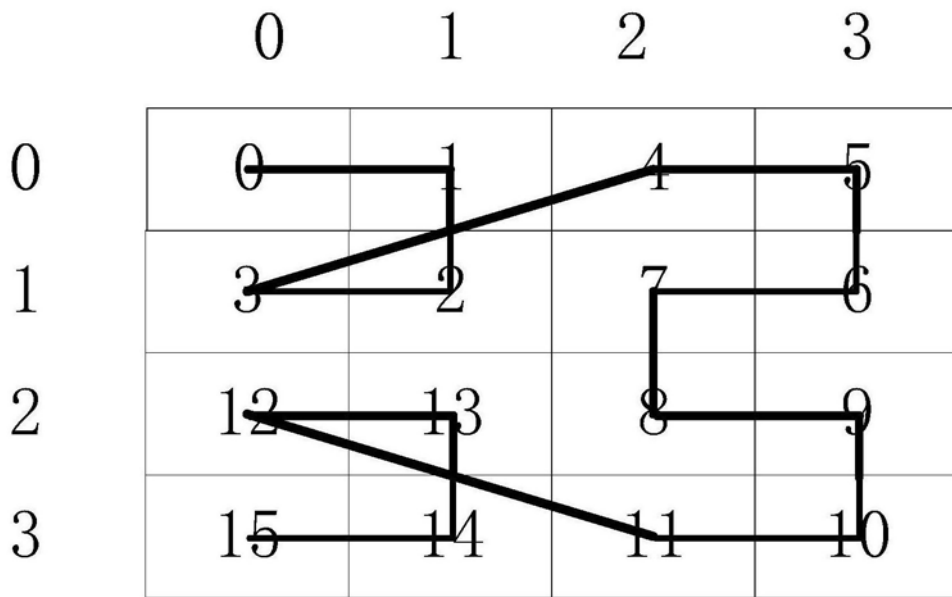


图5

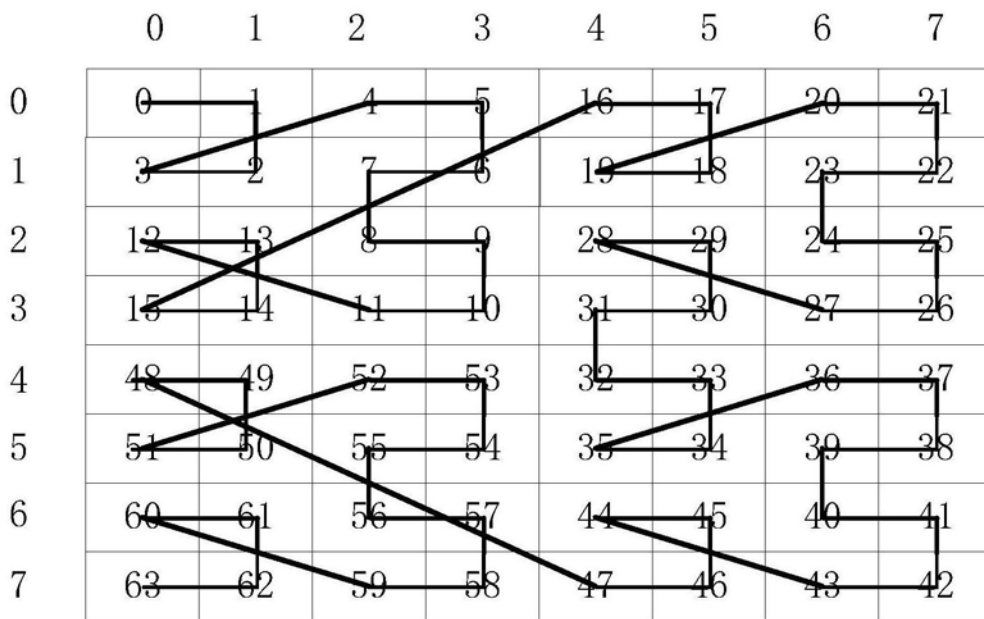


图6

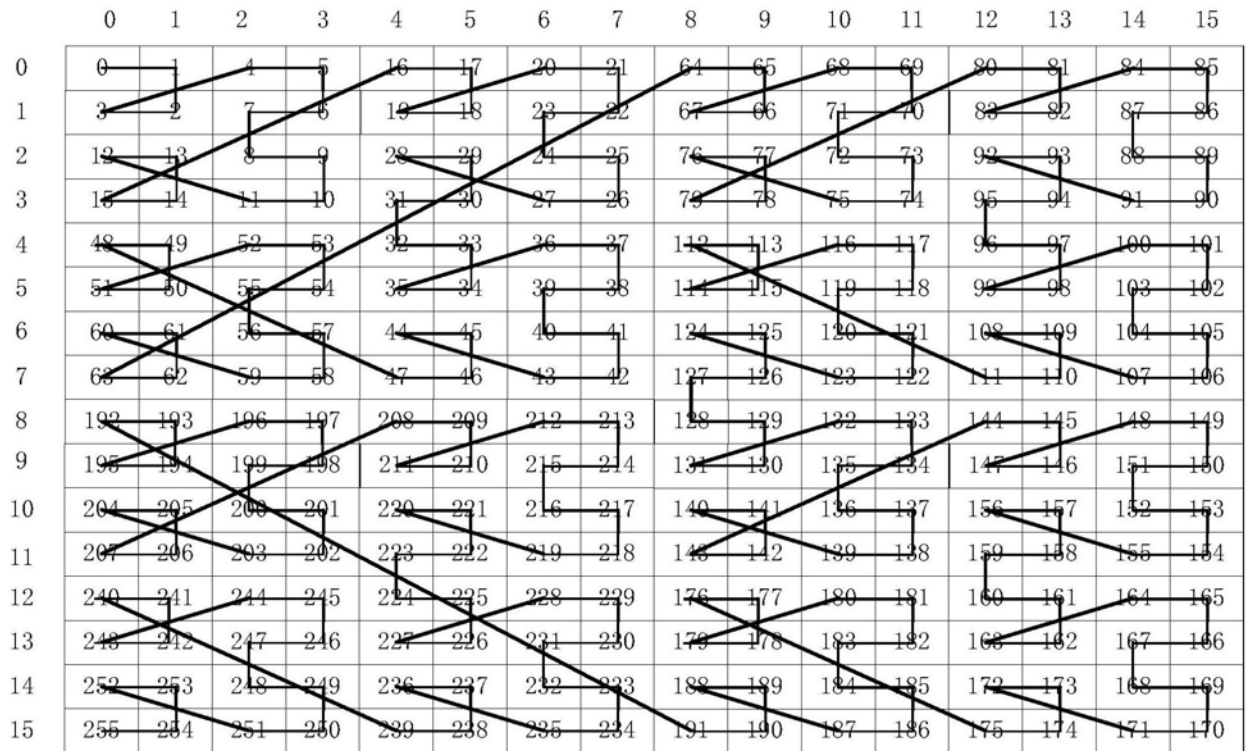


图7

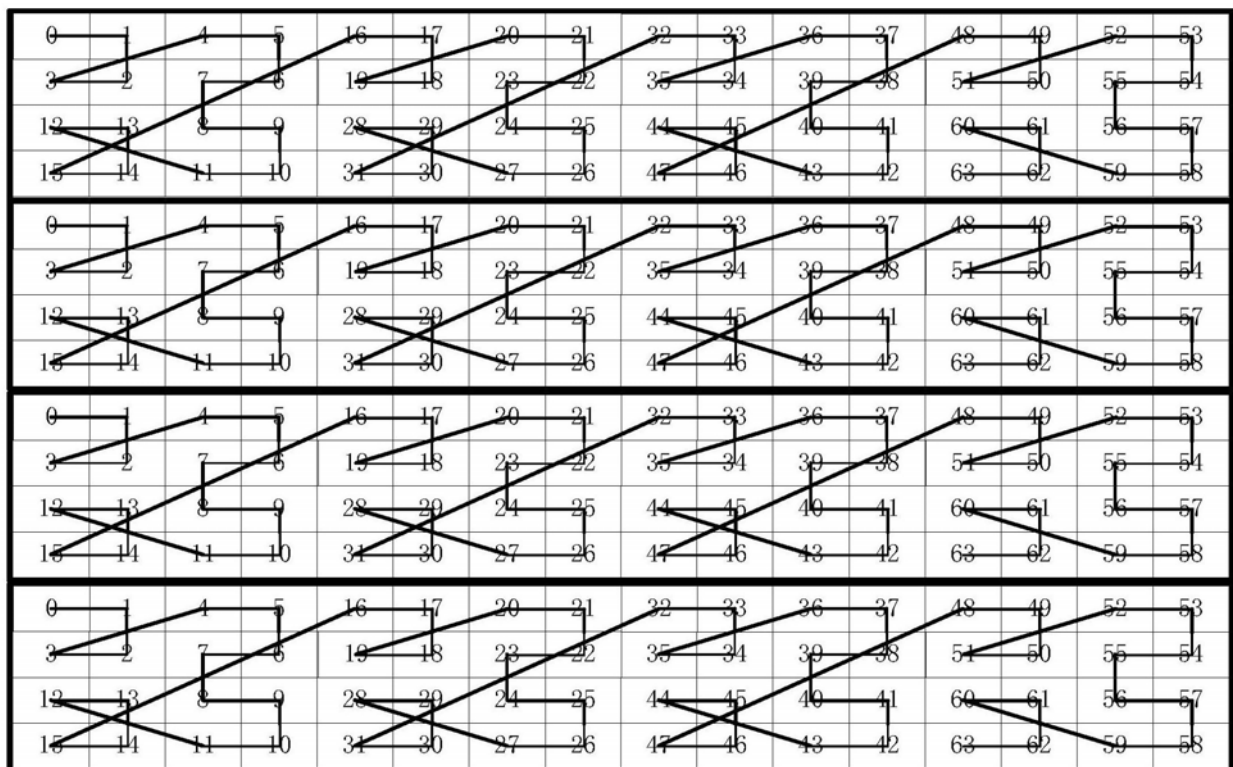


图8