

[19] 中华人民共和国国家知识产权局



[12] 发明专利申请公布说明书

[21] 申请号 200710051921.0

[51] Int. Cl.

H04N 7/24 (2006.01)

H04N 7/26 (2006.01)

G06T 9/00 (2006.01)

[43] 公开日 2007 年 10 月 17 日

[11] 公开号 CN 101056399A

[22] 申请日 2007.4.19

[21] 申请号 200710051921.0

[71] 申请人 华中科技大学

地址 430074 湖北省武汉市洪山区珞喻路
1037 号

[72] 发明人 熊承义 田金文 柳 健

[74] 专利代理机构 华中科技大学专利中心

代理人 方 放

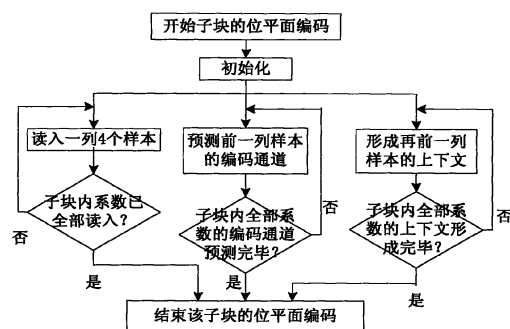
权利要求书 8 页 说明书 21 页 附图 8 页

[54] 发明名称

位平面编码方法及实现该方法的电路

[57] 摘要

位平面编码方法及实现该方法的电路，属于图像数据压缩领域，用于 JPEG 2000 的位平面编码的软件和硬件实现，目的在于减少扫描冗余和复杂度，增强算法的并行性，提高系统数据处理能力。本发明的方法，对每一子块编码包括：(1) 初始化步骤，(2) 并行执行步骤，并行执行读入、编码通道预测和形成上下文子步骤，(3) 结束步骤。本发明的电路包括输入数据缓冲寄存器单元、编码通道预测逻辑电路、编码通道状态寄存器单元、邻域重要性贡献生成电路和上下文形成逻辑电路单元；本发明一次扫描形成子块内所有系数位样本上下文，实现无冗余扫描、在单时钟周期并行形成一个条带列 4 个系数所有位样本的上下文；数据处理能力比同类设计提高 4 倍，性价比更优。



1. 一种位平面编码方法，对每一子块编码包括下述步骤：

(1) 初始化步骤，初始化表示编码子块大小的变量，读入并存储子块内第1条带列的4个系数；

(2) 并行执行步骤，并行同步执行读入子步骤、编码通道预测子步骤和形成上下文子步骤，

(2.1) 读入子步骤：

(2.1.1) 按扫描顺序同时读入一列4个新系数；

(2.1.2) 判定子块内所有列的样本系数是否读入完毕，是则进行步骤(3.1)，否则返回步骤(2.1)；

(2.2) 编码通道预测子步骤：

(2.2.1) 预测前一列样本的编码通道值；

(2.2.2) 判定子块内的所有列的编码通道预测是否进行完毕，是则进行步骤(3.2)，否则返回步骤(2.2)；

(2.3) 形成上下文子步骤：

(2.3.1) 形成更前一列样本的上下文；

(2.3.2) 判定子块内所有样本的上下文是否形成完毕，是则进行步骤(3.3)，否则返回步骤(2.3)；

(3) 结束步骤，包括(3.1)结束读取当前子块的系数；(3.2)结束预测样本的编码通道；(3.3)结束当前子块的上下文形成。

2. 如权利要求1所述的位平面编码方法，其特征在于所述并行执行步骤的编码通道预测子步骤中预测前一列样本的编码通道时，过程为：

(1) 计算当前列样本及其邻域的初始重要性状态；编码块中第*i*个系数在第*p*位样本被编码之前的初始重要性状态 σ_i^{p+1} ：

$$\begin{cases} \sigma_i^{p+1} = v_i^{p+1} | v_i^{p+2} | \dots | v_i^{M-1}, & p = 0, 1, \dots, M-2 \\ \sigma_i^M = 0 \end{cases}$$

式中, v_i^p 表示编码块中第 i 个系数第 p 位的幅度值, 符号 “|” 表示逻辑或运算, M 为幅度位平面的总数目, $M-1$ 为系数幅度值的最高位;

(2) 计算编码通道预测阶段的邻域样本的重要性贡献; 定义 P_z^p 为第 z 系数的第 p 位样本的编码通道值, $P_z^p=2$ 表示该样本应该进入幅度细化通道编码, $P_z^p=1$ 表示该样本应该进入重要性传播通道编码, $P_z^p=3$ 表示该样本应该进入清扫通道编码; 第 p 位平面的第 i 样本的第 j 邻域对第 i 样本的重要性贡献 σ_{ji}^p 计算如下:

(2.1) 如果邻域样本是在第 i 样本之前被扫描, 则对第 i 样本的重要性贡献 $\sigma_{ji}^p = (P_j^p = 1) \& v_j^p | (P_j^p = 2) = (P_j^p = 1) \& v_j^p | \sigma_j^{p+1}$,

定义 m_j 为邻域系数 j 的最高非零幅度位, $P_j^{m_j}$ 为系数 j 最高非零幅度位的编码通道值, 则上式又等价:

$$\sigma_{ji}^p = P_j^{m_j} \& v_j^p | \sigma_j^{p+1},$$

式中“&”表示“与”运算, “|”表示逻辑或运算, $P_j^{m_j}$ 用 1 位二进制数表示, 当 $P_j^{m_j}=1$ 时最高非零幅度位的编码通道为 1, 否则最高非零幅度位的编码通道为 3, 第 i 系数的最高非零幅度位样本的编码通道值

$$P_i^m = \sum_{p=0}^{M-1} v_i^p \& (P_i^p = 1), \text{ “}\Sigma\text{” 在这里也表示逻辑或运算};$$

(2.2) 如果邻域样本是在第 i 样本之后被扫描, 则对第 i 样本的重要性贡献 $\sigma_{ji}^p = \sigma_j^{p+1}$,

而当邻域样本位于当前被编码条带之后且编码模式选为 “vertical causal mode” 时, $\sigma_{ji}^p = 0$;

(3) 生成样本的编码通道值, 采用二进制数 $\text{cp}_{pi} = b_1 b_0$ 表示当前位样本的编码通道预测值, $b_1 = \sigma_i^{p+1}$, $b_0 = \sigma_{1i}^p | \sigma_{2i}^p | \sigma_{3i}^p | \sigma_{4i}^p | \sigma_{5i}^p | \sigma_{6i}^p | \sigma_{7i}^p | \sigma_{8i}^p$, (3.1) 如果 $b_1=1$, 则编码块中第 i 系数的第 p 位样本进入幅度细化通道编码, 记

为 $P_i^p=2$; (3.2) 如果 $\text{cpbi}=00$, 则编码块中第 i 系数的第 p 位样本进入清扫通道编码, 记为 $P_i^p=3$; (3.3) 如果 $\text{cpbi}=01$, 编码块中第 i 系数的第 p 位样本进入重要性传播通道编码, 记为 $P_i^p=1$ 。

3. 如权利要求 1 所述的位平面编码方法, 其特征在于所述并行执行步骤的形成上下文子步骤中, 形成更前一系列系数位的上下文时, 过程为:

(1) 在上下文形成阶段, 在线并行计算一系列 4 个系数的所有位样本的邻域的重要性贡献,

(1.1) 如果被编码的当前样本 c 已被预测进入重要性通道编码, 则分别不同情况计算其邻域 j 的重要性贡献 σ_{jc}^p :

A. 如果邻域样本是在当前被编码的第 c 样本之前被扫描, 则对第 c 样本的重要性贡献 $\sigma_{jc}^p = (P_j^p = 1) \& v_j^p \mid (P_j^p = 2) = (P_j^p = 1) \& v_j^p \mid \sigma_j^{p+1}$

定义 m_j 为邻域系数 j 的最高非零幅度位, $P_j^{m_j}$ 为系数 j 最高非零幅度位的编码通道值, 则上式又等价于:

$$\sigma_{jc}^p = P_j^{m_j} \& v_j^p \mid \sigma_j^{p+1}$$

$P_j^{m_j}$ 用 1 位二进制数表示, 当 $P_j^{m_j}=1$ 时最高非零幅度位的编码通道为 1, 否则最高非零幅度位的编码通道为 3, 第 i 系数的最高非零幅度位样本

$$\text{编码通道值 } P_i^{m_i} = \sum_{p=0}^{M-1} v_i^p \& (P_i^p = 1) \quad ;$$

B. 如果邻域样本是在第 c 样本之后被扫描, 则对第 c 样本的重要性贡献 $\sigma_{jc}^p = \sigma_j^{p+1}$,

C. 而当邻域样本位于当前被编码条带之后且编码模式选为 “vertical causal mode” 时, $\sigma_{jc}^p = 0$;

(1.2) 如果当前被编码的第 c 样本已被预测进入幅度细化通道编码, 则其邻域的重要性贡献 $\sigma_{jc}^p = (P_j^p = 1) \& v_j^p \mid \sigma_j^{p+1}$,

但是, 当邻域样本位于当前被编码条带之后且编码模式选为 “vertical

causal mode” 时, $\sigma_{jc}^p = 0$;

(1.3) 如果当前被编码的第 c 样本已被预测进入清扫通道, 则:

对于位于当前被编码样本前的样本, 其邻域的重要性贡献 $\sigma_{jc}^p = v_j^p$,

对于位于当前被编码样本后的样本, 其邻域的重要性贡献

$$\sigma_{jc}^p = (P_j^p = P1) \& v_j^p ,$$

对于位于当前被编码样本后且在下一条带的样本, 编码模式选为
“vertical causal mode” 时其邻域的重要性贡献 $\sigma_{jc}^p = 0$;

(1.4) 在符号编码原语中需要的邻域 j 的符号对当前编码样本 c 进

行符号编码时的重要性贡献 $\sigma_{jc}^x = \sum_{p=0}^{M-1} (\sigma_{jc}^p \& v_c^p \& (P_c^p \neq 2))$,

上式中的符号 “ Σ ” 代表求逻辑或运算;

(2) 并行计算 1 列 4 个样本的上下文环境和判决数据:

(2.1) 对一系列标号分别为 0、1、2 和 3 号的 4 个系数的位样本并行进行零编码 ZC、符号编码 SC 和幅度细化编码 MRC, 以及对当前的一系列系数执行对应幅度位平面的游程编码 RLC, 产生相应的上下文环境和判决数据对, 包括 CXD01、CXD02、CXD03、CXD11、CXD12、CXD13、CXD21、CXD22、CXD23、CXD31、CXD32、CXD33、CXD4、CXD5、CXD6;

(2.2) 样本的上下文输出, 步骤为:

(2.2.1) 判断游程编码的条件是否满足, 满足则进入 (2.2.3), 否则进行 (2.2.2);

(2.2.2) 判断 0 号样本的编码通道预测值是否为 2, 是则输出 CXD03 后进行 (2.2.5), 否则, 输出 CXD01 后进行 (2.2.4);

(2.2.3) 判断游程的长度是否等于 4, 否则输出 CXD4、CXD5 和 CXD6 后进行 (2.2.11), 是则输出 CXD4 后进行 (2.2.14);

(2.2.4) 判断 0 号样本的幅度值是否为 1, 是则输出 CXD02 后进行

(2.2.5), 否则直接进行 (2.2.5);

(2.2.5) 判断 1 号样本的编码通道预测值是否为 2, 是则输出 CXD13 后进行 (2.2.7), 否则输出 CXD11 后进行 (2.2.6);

(2.2.6) 判断 1 号样本的幅度值是否为 1, 是则输出 CXD12 后进行 (2.2.7), 否则直接进行 (2.2.7);

(2.2.7) 判断 2 号样本的编码通道预测值是否为 2, 是则输出 CXD23 后进行 (2.2.9), 否则输出 CXD21 后进行 (2.2.8);

(2.2.8) 判断 2 号样本的幅度值是否为 1, 是则输出 CXD22 后进行 (2.2.9), 否则直接进行 (2.2.9);

(2.2.9) 判断 3 号样本的编码通道预测值是否为 2, 是则输出 CXD33 后进行 (2.2.14), 否则输出 CXD31 后进行 (2.2.10);

(2.2.10) 判断 3 号样本的幅度值是否为 1, 是则输出 CXD32 后进行 (2.2.14), 否则, 直接进行 (2.2.14);

(2.2.11) 判断游程长度是否等于 0, 是则进行 (2.2.4), 否则进行 (2.2.12);

(2.2.12) 判断游程长度是否等于 1, 是则进行 (2.2.6), 否则进行 (2.2.13);

(2.2.13) 判断游程长度是否等于 2, 是则进行 (2.2.8), 否则进行 (2.2.10);

(2.2.14) 结束样本的上下文输出。

4. 一种实现位平面编码方法的位平面编码电路, 包括输入数据缓冲寄存器单元、编码通道预测逻辑电路、编码通道状态寄存器单元、邻域重要性贡献生成电路和上下文形成逻辑电路单元; (1) 输入数据缓冲寄存器单元按照规定的扫描方式存放最近输入的当前条带的 4 行×4 列个系数, 以及前一条带第 4 行的有用系数值和当前条带第 4 行移出的系数值;

(2) 编码通道预测逻辑电路按照所述并行执行步骤的编码通道预测子步骤中预测前一行样本的编码通道方法产生相应列系数的编码通道值；(3) 编码通道状态寄存器单元用来存储当前条带的当前上下文形成列及其前一行共 8 个系数的编码通道预测值，以及前一条带第 4 行的有用系数和当前条带第 4 行移出的系数的最高非零幅度位的编码通道预测值；(4) 邻域重要性贡献生成电路根据所述形成上下文子步骤中，形成更前一行样本的上下文时所述的方法，在每个时钟周期并行产生 4 个系数的用以形成上下文的邻域重要性贡献；(5) 上下文形成逻辑电路单元根据所述形成上下文子步骤中，形成更前一行样本的上下文时的上下文形成方法，在每个时钟周期并行产生 4 个系数的上下文环境和判决数据对。

5. 如权利要求 4 所述的位平面编码电路，其特征在于所述输入数据缓冲寄存器单元包括 4 个 4 单位延时移位寄存器组和一个 N 单位延时移位寄存器组，每个 4 单位延时移位寄存器组由 4 个寄存器串联构成，用来存储输入的当前条带的 1 行 \times 4 列系数；N 单位延时移位寄存器组由 N 个寄存器串联构成，用来存储前一条带第 4 行的有用系数与当前条带的第 4 行移出的系数。

6. 如权利要求 4 所述的位平面编码电路，其特征在于所述编码通道预测逻辑电路包括并行工作的 4 个处理器单元，每个处理器单元输入当前被预测编码通道的系数及其 8 邻域系数、8 邻域系数中已预测的编码通道值，产生被预测编码通道系数的编码通道预测值，4 个处理器单元在单时钟周期并行产生一行 4 个系数除最高非零位平面位外的所有幅度位的编码通道预测值。

7. 如权利要求 4 所述的位平面编码电路，其特征在于所述编码通道

状态寄存器单元包括 4 组 2 单位延时寄存器组和一个 $N+2$ 单位延时移位寄存器组；每组 2 单位延时寄存器组由 2 个寄存器串联构成，用来存储当前被编码条带列与其前一行条带列的 1 行 2 个系数除最高非零位平面位外的所有幅度位的编码通道值； $N+2$ 单位延时移位寄存器组由 $N+2$ 个寄存器串联构成，分别用来存储当前被编码条带列的第 4 系数、其前一行条带列的第 4 系数以及缓存在数据缓冲寄存器单元 N 单位延时移位寄存器的 N 个系数的最高非零幅度位的编码通道值。

8. 如权利要求 4 所述的位平面编码电路，其特征在于所述邻域重要性贡献生成电路包括并行工作的 4 个处理器单元，每个处理器单元输入当前被编码系数及其 8 邻域系数、以及它们的编码通道值，产生被编码系数的 8 邻域重要性贡献，4 个处理器单元在单时钟周期内并行地生成当前被编码条带列 4 个系数的所有位样本的上下文形成时需要的邻域重要性贡献。

9. 如权利要求 4 所述的位平面编码电路，其特征在于所述上下文形成逻辑电路单元包括 4 组单字编码模块、1 个游程编码原语模块、输出端口以及控制电路，每组单字编码模块由零编码原语模块、幅度细化编码原语模块和符号编码原语模块组成；(1) 单字编码模块中零编码原语模块和幅度细化编码原语模块的输入为当前被编码系数的幅度值、及当前被编码系数邻域样本在水平方向、垂直方向和对角线方向的重要性贡献，产生幅度位的上下文环境和判决数据对；符号编码原语模块的输入为当前被编码系数的符号和邻域系数的符号在水平方向、垂直方向的重要性贡献，产生当前被编码系数符号位的上下文环境和判决数据对；(2) 游程编码原语模块的输入为当前被编码条带列 4 个系数的幅度值和游程编码标志信号，产生一个条带列的上下文环境和判决数据对；(3) 4 组单字

编码模块和游程编码原语模块在每个内部时钟周期，并行产生每个位平面同一条带列4个系数每个幅度位平面样本的15个上下文环境和判决数据对到输出端口，输出端口输出其中1到10个有效的上下文环境和判决数据对；（4）控制电路判定对应条带列是否进行游程编码并产生游程编码标志信号到游程编码原语模块，同时产生相应的输出控制信号到输出端口，控制输出端口选择有效的上下文环境和判决数据对的输出。

位平面编码方法及实现该方法的电路

技术领域

本发明属于图像数据压缩领域，特别涉及一种位平面编码方法及实现该方法的电路，用于 JPEG 2000 的位平面编码的软件和硬件实现。

背景技术

嵌入式优化截断块编码(Embedded Block Coding Optimal Truncation)是一种已被新的静止图像编码标准 JPEG2000 采用的最新编码技术，其实现主要包含位平面编码(Bit Plane Coding)和自适应二元算术编码(Binary Arithmetic Coding)两部分。其位平面编码为一种基于分数位平面的多通道编码技术，以产生待编码样本的上下文环境和判决数据对，进一步用于后级的自适应二元算术编码。

在 EBCOT 编码算法的位平面编码部分，采用四种不同的编码原语之一或之二来形成系数样本的上下文环境与对判决数据对 (CXD)：(1) 零编码原语，利用当前编码样本的在水平方向邻域的重要性贡献 σ_h 、垂直方向邻域的重要性贡献 σ_v 和对角方向邻域的重要性贡献 σ_d ，产生待编码系数幅度样本的上下文环境，JPEG2000 使用的待编码样本的上下文环境和与判决数据与其邻域的重要性状态的关系如表 1 所示，表中的 x 可以为任意的 0 或 1 值， D 表示编码样本的幅度值；(2) 符号编码原语，利用当前编码系数的在平方向邻域系数的符号重要性贡献 σ_h^s 、垂直方向邻域的符号重要性贡献 σ_v^s ，产生当前编码系数符号位样本的上下文，JPEG2000 使用的待编码系数符号样本的上下文环境和判决数据与其邻域系数的符号重要性贡献的关系如表 2 所示，表中的 S 代表编码系数的

符号值, \bar{s} 代表编码系数的符号值的反码; (3) 幅度细化编码原语, 利用当前编码样本的在平方向邻域的重要性贡献 σ_h 、垂直方向邻域的重要性贡献 σ_v 和对角方向邻域样本的重要性贡献 σ_d 的和 σ_Σ , 以及与当前编码样本的幅度细化状态信息 $\bar{\sigma}$, 产生待编码系数幅度样本的上下文环境和判决数据, JPEG2000 使用的待编码样本的上下文环境和判决数据与其邻域的重要性贡献的关系如表 3 所示, 表中的 x 可以为可取的任意值, D 表示编码样本的幅度值; (4) 游程编码原语, 在检测到当前编码条带列满足执行游程编码的条件下, 进一步依据游程的长度形成 1~3 对上下文环境和判决数据, JPEG2000 使用的待编码条带列执行游程编码产生的上下文如表 4 示。

表1 零编码产生样本的上下文与其邻域的重要性状态的关系

| LL和LH子带 (垂直高通) | | | HL子带 (水平高通) | | | HH子带 (对角高通) | | 上下文标签 | 判决数据 |
|-------------------|------------|------------|----------------|------------|------------|-----------------------|------------|-------|------|
| σ_h | σ_v | σ_d | σ_h | σ_v | σ_d | $\sigma_h + \sigma_v$ | σ_d | | |
| 2 | x | x | x | 2 | x | x | >2 | 8 | D |
| 1 | >0 | x | >0 | 1 | x | >0 | 2 | 7 | D |
| 1 | 0 | >0 | 0 | 1 | >0 | 0 | 2 | 6 | D |
| 1 | 0 | 0 | 0 | 1 | 0 | >1 | 1 | 5 | D |
| 0 | 2 | x | 2 | 0 | x | 1 | 1 | 4 | D |
| 0 | 1 | x | 1 | 0 | x | 0 | 1 | 3 | D |
| 0 | 0 | >1 | 0 | 0 | >1 | >1 | 0 | 2 | D |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | D |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | D |

表2 符号编码产生样本的上下文环境和判决数据

| σ_h^x | σ_v^x | 判决数据 | 上下文环境 |
|--------------|--------------|-----------|-------|
| 1 | 1 | S | 13 |
| 1 | 0 | S | 12 |
| 1 | -1 | S | 11 |
| 0 | 1 | S | 10 |
| 0 | 0 | S | 9 |
| 0 | -1 | \bar{S} | 10 |
| -1 | 1 | \bar{S} | 11 |
| -1 | 0 | \bar{S} | 12 |
| -1 | -1 | \bar{S} | 13 |

表3 幅度细化编码产生样本的上下文环境和判决数据

| $\bar{\sigma}$ | σ_Σ | 上下文环境 | 判决数据 |
|----------------|-----------------|-------|------|
| 1 | x | 16 | D |
| 0 | >0 | 15 | D |
| 0 | 0 | 14 | D |

表4 游程编码产生样本的上下文环境及判决数据

| 游程长度 | 0 | | | 1 | | | 2 | | | 3 | | | 4 |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 上下文环境 | 17 | 18 | 18 | 17 | 18 | 18 | 17 | 18 | 18 | 17 | 18 | 18 | 17 |
| 判决数据 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

基于传统的分数位平面编码算法的实现需要对原始数据进行全部共 $3n-2$ 通道编码(n 为样本幅度值的最高位), 因此需要对原始数据进行全部共 $3n-2$ 次扫描, 其软硬件实现具有很大的扫描冗余和很高的计算复杂度。根据 JPEG 2000 标准, 每个样本的上下文形成需要已知该样本及其 8 邻域的 5 个状态变量的值, 因此需要较大规模的存储器, 缓存状态变量的中间值, 从而要求系统具有较高的硬件实现成本和具有较大的功率消耗。针对上述问题, C. J. Lian 等人在文献“Analysis and architecture design of block-coding engine for ebcot in JPEG2000,” IEEE Trans. on Circuits and Systems for Video Technology, vol.13, pp.219-230, 2003 提出了一种采用跳样本、跳列等优化串行编码(optimal serial coding)的技术减少通道扫描的冗余性, 从而减少扫描时间。J. S. Chiang 等人在文献“Efficient pass parallel architecture for EBCOT in JPEG2000,” Proceeding of IEEE ISCAS-2002, May 2002 提出了采用通道并行编码(pass-parallel coding)的技术减少对存储器的访问和增加编码速度。朱悦心等人提出了一种采用多级查询表的位平面扫描技术减少通道扫描的冗余性和增加通道扫描和状态变量的并行性。刘凯等人在文献“Bit plane parallel coder for EBCOT and its VLSI architecture,” Chinese Journal of Computers, vol.27, no.7, pp.928-935, 2004 提出了一种基于位平面并行的编码技术, 实现位平面编码的并行完成。Fang 等人在文献“Novel word-level algorithm of embedded block coding in JPEG2000,” Proceeding of IEEE ICME2003, vol.1, pp.123-139, 2003 提出了一种基于字级的位平面编码技术, 但是其实现需要经过两次扫描, 因此需要较大规模的存储器需求而只适合于软件实现。Xu 等人在文献“Bit-plane and pass dual parallel architecture for coefficient bit modeling in JPEG2000,” Proceeding of IEEE ICASSP'2004, vol.5, pp.85-88, 2004 提出了一种通道和位平面编码双并行的编码思想以进一步加速了位平面建模的速度和减少存储器需求。上述方法在一定程度上减少了冗余扫描时间,

增加了数据处理能力，但他们的实现仍具有较低的数据处理能力，并具有较高的硬件实现成本和控制复杂度。

发明内容

本发明提出一种位平面编码方法及实现该方法的电路，针对现有技术存在的缺陷，提出基于字级顺序扫描和多字并行编码的技术，目的在于有效地减少扫描冗余和控制复杂度，有效地增强算法实现的并行性，进一步有效地提高系统的数据处理能力。

本发明的一种位平面编码方法，对每一子块编码包括下述步骤：

(1) 初始化步骤，初始化表示编码子块大小的变量，读入并存储子块内第1条带列的4个系数；

(2) 并行执行步骤，并行同步执行读入子步骤、编码通道预测子步骤和形成上下文子步骤，

(2.1) 读入子步骤：

(2.1.1) 按扫描顺序同时读入一列4个新系数；

(2.1.2) 判定子块内所有列的样本系数是否读入完毕，是则进行步骤(3.1)，否则返回步骤(2.1)；

(2.2) 编码通道预测子步骤：

(2.2.1) 预测前一列样本的编码通道值；

(2.2.2) 判定子块内的所有列的编码通道预测是否进行完毕，是则进行步骤(3.2)，否则返回步骤(2.2)；

(2.3) 形成上下文子步骤：

(2.3.1) 形成更前一列样本的上下文；

(2.3.2) 判定子块内所有样本的上下文是否形成完毕，是则进行步骤(3.3)，否则返回步骤(2.3)；

(3) 结束步骤，包括(3.1)结束读取当前子块的系数；(3.2)结束

预测样本的编码通道；(3.3) 结束当前子块的上下文形成。

所述的位平面编码方法，其特征在于所述并行执行步骤的编码通道预测子步骤中预测前一列样本的编码通道时，过程为：

(1) 计算当前列样本及其邻域的初始重要性状态；编码块中第 i 个系数在第 p 位样本被编码之前的初始重要性状态 σ_i^{p+1} ：

$$\begin{cases} \sigma_i^{p+1} = v_i^{p+1} | v_i^{p+2} | \dots | v_i^{M-1}, & p = 0, 1, \dots, M-2 \\ \sigma_i^M = 0 \end{cases}$$

式中， v_i^p 表示编码块中第 i 个系数第 p 位的幅度值，符号“|”表示逻辑或运算， M 为幅度位平面的总数目， $M-1$ 为系数幅度值的最高位；

(2) 计算编码通道预测阶段的邻域样本的重要性贡献；定义 P_z^p 为第 z 系数的第 p 位样本的编码通道值， $P_z^p=2$ 表示该样本应该进入幅度细化通道编码， $P_z^p=1$ 表示该样本应该进入重要性传播通道编码， $P_z^p=3$ 表示该样本应该进入清扫通道编码；第 p 位平面的第 i 样本的第 j 邻域对第 i 样本的重要性贡献 σ_{ji}^p 计算如下：

(2.1) 如果邻域样本是在第 i 样本之前被扫描，则对第 i 样本的重要性贡献 $\sigma_{ji}^p = (P_j^p = 1) \& v_j^p | (P_j^p = 2) = (P_j^p = 1) \& v_j^p | \sigma_j^{p+1}$ ，

定义 m_j 为邻域系数 j 的最高非零幅度位， $P_j^{m_j}$ 为系数 j 最高非零幅度位的编码通道值，则上式又等价于：

$$\sigma_{ji}^p = P_j^{m_j} \& v_j^p | \sigma_j^{p+1},$$

式中“&”表示“与”运算，“|”表示逻辑或运算， $P_j^{m_j}$ 用 1 位二进制数表示，当 $P_j^{m_j}=1$ 时最高非零幅度位的编码通道为 1，否则编码通道为 3，第 i 系数的最高非零幅度位样本的编码通道值 $P_i^{m_i} = \sum_{p=0}^{M-1} v_i^p \& (P_i^p = 1)$ ，“ Σ ”在这里也表示逻辑或运算；

(2.2) 如果邻域样本是在第 i 样本之后被扫描，则对第 i 样本的重要

性贡献 $\sigma_{ji}^p = \sigma_j^{p+1}$,

而当邻域样本位于当前被编码条带之后且编码模式选为“vertical causal mode”时, $\sigma_{ji}^p = 0$;

(3) 生成样本的编码通道值, 采用二进制数 $\text{cpbi} = b_1 b_0$ 表示当前位样本的编码通道预测值, $b_1 = \sigma_i^{p+1}$, $b_0 = \sigma_{1i}^p | \sigma_{2i}^p | \sigma_{3i}^p | \sigma_{4i}^p | \sigma_{5i}^p | \sigma_{6i}^p | \sigma_{7i}^p | \sigma_{8i}^p$, (3.1) 如果 $b_1 = 1$, 则编码块中第 i 系数的第 p 位样本进入幅度细化通道编码, 记为 $P_i^p = 2$; (3.2) 如果 $\text{cpbi} = 00$, 则编码块中第 i 系数的第 p 位样本进入清扫通道编码, 记为 $P_i^p = 3$; (3.3) 如果 $\text{cpbi} = 01$, 编码块中第 i 系数的第 p 位样本进入重要性传播通道编码, 记为 $P_i^p = 1$ 。

所述的位平面编码方法, 其特征在于所述并行执行步骤的形成上下文子步骤中, 形成更前一系列系数位的上下文时, 过程为:

(1) 在上下文形成阶段, 在线并行计算一系列 4 个系数的所有位样本的邻域的重要性贡献,

(1.1) 如果当前被编码的样本 c 已被预测进入重要性通道编码, 则分别不同情况计算其邻域 j 的重要性贡献 σ_{jc}^p :

A. 如果邻域样本是在第 c 样本之前被扫描, 则对第 c 样本的重要性贡献 $\sigma_{jc}^p = (P_j^p = 1) \& v_j^p | (P_j^p = 2) = (P_j^p = 1) \& v_j^p | \sigma_j^{p+1}$

定义 m_j 为邻域系数 j 的最高非零幅度位, $P_j^{m_j}$ 为系数 j 最高非零幅度位的编码通道值, 则上式又等价于:

$$\sigma_{jc}^p = P_j^{m_j} \& v_j^p | \sigma_j^{p+1}$$

$P_j^{m_j}$ 用 1 位二进制数表示, 当 $P_j^{m_j} = 1$ 时最高非零幅度位的编码通道为 1, 否则为编码通道 3, 第 i 系数的最高非零幅度位样本编码通道值

$$P_i^{m_i} = \sum_{p=0}^{M-1} v_i^p \& (P_i^p = 1) \quad ;$$

B.如果邻域样本是在第 c 样本之后被扫描,则对第 c 样本的重要性贡献 $\sigma_{jc}^p = \sigma_j^{p+1}$,

C.而当邻域样本位于当前被编码条带之后且编码模式选为“vertical causal mode”时, $\sigma_{jc}^p = 0$;

(1.2) 如果当前被编码的样本 c 已被预测进入幅度细化通道编码,则其邻域的重要性贡献 $\sigma_{jc}^p = (P_j^p = 1) \& v_j^p | \sigma_j^{p+1}$,

但是,当邻域样本位于当前被编码条带之后且编码模式选为“vertical causal mode”时, $\sigma_{jc}^p = 0$;

(1.3) 如果被编码的当前样本已被预测进入清扫通道,则:

对于位于当前被编码样本前的样本,其邻域的重要性贡献 $\sigma_{jc}^p = v_j^p$,

对于位于当前被编码样本后的样本,其邻域的重要性贡献

$$\sigma_{jc}^p = (P_j^p = P1) \& v_j^p ,$$

对于位于当前被编码样本后且在下一条带的样本,编码模式选为“vertical causal mode”时其邻域的重要性贡献 $\sigma_{jc}^p = 0$;

(1.4) 在符号编码原语中需要的邻域 j 的符号对当前编码样本 c 进行符号编码时的重要性贡献 $\sigma_{jc}^x = \sum_{p=0}^{M-1} (\sigma_{jc}^p \& v_c^p \& (P_c^p \neq 2))$,

上式中的符号“ Σ ”代表求逻辑或运算;

(2) 并行计算 1 列 4 个样本的上下文环境和判决数据:

(2.1) 对一系列标号分别为 0、1、2 和 3 号的 4 个系数的位样本并行进行零编码 ZC、符号编码 SC 和幅度细化编码 MRC,以及对当前的一系列系数执行对应幅度位平面的游程编码 RLC,产生相应的上下文环境和判决数据对,包括 CXD01、CXD02、CXD03、CXD11、CXD12、CXD13、CXD21、CXD22、CXD23、CXD31、CXD32、CXD33、CXD4、CXD5、CXD6;

(2.2) 样本的上下文输出,步骤为:

(2.2.1) 判断游程编码的条件是否满足，满足则进入 (2.2.3)，否则进行 (2.2.2)；

(2.2.2) 判断 0 号样本的编码通道预测值是否为 2，是则输出 CXD03 后进行 (2.2.5)，否则，输出 CXD01 后进行 (2.2.4)；

(2.2.3) 判断游程的长度是否等于 4，否则输出 CXD4、CXD5 和 CXD6 后进行 (2.2.11)，是则输出 CXD4 后进行 (2.2.14)；

(2.2.4) 判断 0 号样本的幅度值是否为 1，是则输出 CXD02 后进行 (2.2.5)，否则直接进行 (2.2.5)；

(2.2.5) 判断 1 号样本的编码通道预测值是否为 2，是则输出 CXD13 后进行 (2.2.7)，否则输出 CXD11 后进行 (2.2.6)；

(2.2.6) 判断 1 号样本的幅度值是否为 1，是则输出 CXD12 后进行 (2.2.7)，否则直接进行 (2.2.7)；

(2.2.7) 判断 2 号样本的编码通道预测值是否为 2，是则输出 CXD23 后进行 (2.2.9)，否则输出 CXD21 后进行 (2.2.8)；

(2.2.8) 判断 2 号样本的幅度值是否为 1，是则输出 CXD22 后进行 (2.2.9)，否则直接进行 (2.2.9)；

(2.2.9) 判断 3 号样本的编码通道预测值是否为 2，是则输出 CXD33 后进行 (2.2.14)，否则输出 CXD31 后进行 (2.2.10)；

(2.2.10) 判断 3 号样本的幅度值是否为 1，是则输出 CXD32 后进行 (2.2.14)，否则，直接进行 (2.2.14)；

(2.2.11) 判断游程长度是否等于 0，是则进行 (2.2.4)，否则进行 (2.2.12)；

(2.2.12) 判断游程长度是否等于 1，是则进行 (2.2.6)，否则进行 (2.2.13)；

(2.2.13) 判断游程长度是否等于 2，是则进行 (2.2.8)，否则进行 (2.2.10)；

(2.2.14) 结束样本的上下文输出。

本发明的一种位平面编码电路，包括输入数据缓冲寄存器单元、编码通道预测逻辑电路、编码通道状态寄存器单元、邻域重要性贡献生成电路和上下文形成逻辑电路单元；（1）输入数据缓冲寄存器单元按照规定的扫描方式存放最近输入的当前条带的4行×4列个系数，以及前一条带第4行的有用系数值和当前条带第4行移出的系数值；（2）编码通道预测逻辑电路按照所述并行执行步骤的编码通道预测子步骤中预测前一列样本的编码通道方法产生相应列系数的编码通道值；（3）编码通道状态寄存器单元用来存储当前条带的当前上下文形成列及其前一列共8个系数的编码通道预测值，以及前一条带第4行的有用系数和当前条带第4行移出的系数的最高非零幅度位的编码通道预测值；（4）邻域重要性贡献生成电路根据所述形成上下文子步骤中，形成更前一列样本的上下文时所述的方法，在每个时钟周期并行产生4个系数的用以形成上下文的邻域重要性贡献；（5）上下文形成逻辑电路单元根据所述形成上下文子步骤中，形成更前一列样本的上下文时的上下文形成方法，在每个时钟周期并行产生4个系数的上下文环境和判决数据对。

所述的位平面编码电路，其特征在于所述输入数据缓冲寄存器单元包括4个4单位延时移位寄存器组和一个N单位延时移位寄存器组，每个4单位延时移位寄存器组由4个寄存器串联构成，用来存储输入的当前条带的1行×4列系数；N单位延时移位寄存器组由N个寄存器串联构成，用来存储前一条带第4行的有用系数与当前条带的第4行移出的系数。

所述的位平面编码电路，其特征在于所述编码通道预测逻辑电路包

括并行工作的 4 个处理器单元，每个处理器单元输入当前被预测编码通道的系数及其 8 邻域系数、8 邻域系数中已预测的编码通道值，产生被预测编码通道系数的编码通道预测值，4 个处理器单元在单时钟周期并行产生一列 4 个系数除最高非零位平面位外的所有幅度位的编码通道预测值。

所述的位平面编码电路，其特征在于所述编码通道状态寄存器单元包括 4 组 2 单位延时寄存器组和一个 $N+2$ 单位延时移位寄存器组；每组 2 单位延时寄存器组由 2 个寄存器串联构成，用来存储当前被编码条带列与其前一系列条带列的 1 行 2 个系数除最高非零位平面位外的所有幅度位的编码通道值； $N+2$ 单位延时移位寄存器组由 $N+2$ 个寄存器串联构成，分别用来存储当前被编码条带列的第 4 系数、其前一系列条带列的第 4 系数以及缓存在数据缓冲寄存器单元 N 单位延时移位寄存器的 N 个系数的最高非零位的编码通道值。

所述的位平面编码电路，其特征在于所述邻域重要性贡献生成电路包括并行工作的 4 个处理器单元，每个处理器单元输入当前被编码系数及其 8 邻域系数、以及它们的编码通道值，产生被编码系数的 8 邻域重要性贡献，4 个处理器单元在单时钟周期内并行地生成当前被编码条带列 4 个系数的所有位样本的上下文形成时需要的邻域重要性贡献。

所述的位平面编码电路，其特征在于所述上下文形成逻辑电路单元包括 4 组单字编码模块、1 个游程编码原语模块、输出端口以及控制电路，每组单字编码模块由零编码原语模块、幅度细化编码原语模块和符号编码原语模块组成；（1）单字编码模块中零编码原语模块和幅度细化编码原语模块的输入为当前被编码系数的幅度值、及当前被编码系数邻域样本在水平方向、垂直方向和对角线方向的重要性贡献，产生幅度位的上

下文环境和判决数据对；符号编码原语模块的输入为当前被编码系数的符号和邻域系数的符号在水平方向、垂直方向的重要性贡献，产生当前被编码系数符号位的上下文环境和判决数据对；（2）游程编码原语模块的输入为当前被编码条带列 4 个系数的幅度值和游程编码标志信号，产生一个条带列的上下文环境和判决数据对；（3）4 组单字编码模块和游程编码原语模块在每个内部时钟周期，并行产生每个位平面同一条带列 4 个系数每个幅度位平面样本的 15 个上下文环境和判决数据对到输出端口，输出端口输出其中 1 到 10 个有效的上下文环境和判决数据对；（4）控制电路判定对应条带列是否进行游程编码并产生游程编码标志信号到游程编码原语模块，同时产生相应的输出控制信号到输出端口，控制输出端口选择有效的上下文环境和判决数据对的输出。

本发明实现了仅通过一次扫描过程即可完成对子块内所有系数的位样本的上下文形成，实现了对编码样本扫描的真正无冗余；实现了在单时钟周期对一个条带列 4 个系数的所有位样本的上下文形成并行执行；比较同类最好设计，本发明的数据处理能力提高了 4 倍，以及具有更高的数据吞吐能力比硬件成本的性能。

附图说明

图 1 是 JPEG2000 标准中采用的位平面编码的扫描模式示意图；

图 2 是 JPEG2000 标准中定义的当前被编码样本 X 及其 8 邻域示意图；

图 3 是本发明的编码通道预测及上下文形成的时间关系示意图；

图 4 是本发明方法的流程示意图；

图 5 是本发明中并行执行步骤的形成上下文子步骤中，并行计算 1 列 4 个样本的上下文环境和实现流程图；

图 6 是本发明图 5 中的样本的上下文输出的方法实现流程图；

- 图 7 是本发明的位平面编码电路示意图；
图 8 是图 7 中输入数据缓冲寄存器单元电路示意图；
图 9 是图 7 中编码通道预测逻辑电路示意图；
图 10 是图 7 中编码通道状态寄存器单元示意图；
图 11 是图 7 中邻域重要性贡献生成电路示意图；
图 12 是图 7 中上下文形成逻辑电路单元示意图。

具体实施方式

下面结合附图和实施例对本发明进行详细说明。

图 1 给出了 JPEG2000 标准中采用的位平面编码数据扫描示意图。对于每个子块的位平面编码按照每 4 行样本定义为一个条带(stripe)，一个条带的一列 4 个样本称为一个条带列(stripe column)。对样本进行扫描的顺序是：首先依次从上往下扫描最高位平面的第一个条带的最左边一个条带列的元素，然后依次从上往下扫描最高位平面的第一个条带的左边第二个条带列的元素，直到该条带的所有样本被全部扫描；接下来将按照上面的方式扫描最高位平面的第二个条带，一直到最高位平面的所有条带的元素被扫描完成。接下来以同样的方式进行对次高位平面和更低位平面的扫描。除了最高位平面只需进行清扫通道的扫描和编码外，其它低位的每个位平面要求进行三个通道(重要性传播通道、幅度细化通道、清扫通道)的扫描和编码。根据此编码规则，对于低位的每个位平面将要求进行 3 次扫描。所以，对于最大幅度值为 M 位表示的子块，完成位平面编码将需要对原始数据进行 $3M-2$ 次扫描。

图 2 给出了 JPEG2000 标准中定义的当前样本与其 8 邻域的示意图。图中 X 表示的是当前样本， $V0$ 和 $V1$ 为 X 的垂直方向邻域， $H0$ 和 $H1$ 为 X 的水平方向邻域， $D0\sim D4$ 为 X 的对角方向邻域。当 X 处在一个条带列的第一个元素时，其邻域 $D0$ 、 $D1$ 和 $V0$ 位于当前编码条带的上一条

带。而当 X 处在一个条带列的第四个元素时，其邻域 D2、D3 和 V1 位于当前编码条带的下一条带。

图 3 给出了本发明方法的编码通道预测及上下文形成的时间关系示意图，图 3 中的 P00~P33 代表依次读入某条带的 4 列数据样本，当读入一个条带列的数据 P30~P33 后，根据上面介绍的编码通道预测方法先进行对于样本 P20~P23 的编码通道的预测，然后根据上面介绍的顺序形成样本上下文的方法形成样本 P10~P13 的上下文。图 3 结构说明本发明的位平面编码过程分为两个阶段：样本的编码通道预测阶段和样本的上下文形成阶段，两个阶段的实现可并行流水线完成，前一系列样本编码通道的预测和后一系列样本的上下文形成流水线并行完成，此过程重复执行，直到对所有样本的位平面编码的完成。本实施例对于在垂直因果模式（vertical causal mode）下的位平面编码流程进行详细说明，但是，其方法可推广适用于在 JPEG 2000 缺省模式（regular mode）的位平面编码。

图 4 具体表示了本发明的方法实现流程图。对每一子块编码的开始首先进行表示编码子块大小的变量的初始化，并预先读入和存储子块内第 1 条带列的 4 个系数。接下来一次并行读入一个条带列的 4 个系数，同时预测前一系列 4 个系数的所有幅度样本的编码通道值，以及形成更前一系列 4 个系数的所有位样本的上下文。不断重复上面的过程，直到完成对子块内的所有系数的位样本的上下文形成。

图 5 给出了本发明中并行执行步骤的形成上下文子步骤中，并行计算 1 列 4 个样本的上下文环境和判决数据的实现流程；并行形成一系列 4 个样本的上下文的实现流程包括两个阶段：首先并行地对一系列分别标号为 0 号、1 号、2 号和 3 号的 4 个样本进行零编码、符号编码和幅度细化编码，以及对一系列样本执行游程编码，产生相应样本的上下文环境和判决数据对，分别表示为 CXD01、CXD02、CXD03、CXD11、CXD12、CXD13、CXD21、CXD22、CXD23、CXD31、CXD32、CXD33、CXD4、

CXD5、CXD6，然后对产生样本的有效上下文环境和判决数据对进行输出。对输入系数的每一个幅度位平面并行地进行相同的操作。

图 6 给出图 5 中的执行对样本的上下文输出实现流程图。定义 $rlc=1$ 表示一列 4 个样本满足游程编码的条件，定义 RL 表示游程的长度，定义 $pp0$ 、 $pp1$ 、 $pp2$ 和 $pp3$ 分别代表同一列中的第 0 号、第 1 号、第 2 号和第 3 号样本的编码通道预测值， $F0$ 、 $F1$ 、 $F2$ 和 $F3$ 分别代表同一列中的第 0 号、第 1 号、第 2 号和第 3 号样本的幅度值。输出过程顺序包括以下步骤：

- (1) 如果满足 $rlc=1$ ，进行 (3)，否则进行 (2)；
- (2) 如果 $pp0=2$ ，则输出 CXD03 后进行 (5)，否则输出 CXD01 后进行 (4)；
- (3) 如果 $RL \neq 4$ ，输出 CXD4、CXD5 和 CXD6 后进行 (11)，否则输出 CXD4 后进行 (14)；
- (4) 如果 $F0=1$ ，输出 CXD02 后进行 (5)，否则直接进行 (5)；
- (5) 如果 $pp1=2$ ，输出 CXD13 后进行 (7)，否则输出 CXD11 后进行 (6)；
- (6) 如果 $F1=1$ ，输出 CXD12 后进行 (7)，否则直接进行 (7)；
- (7) 如果 $pp2=2$ ，输出 CXD23 后进行 (9)，否则输出 CXD21 后进行 (8)；
- (8) 如果 $F2=1$ ，输出 CXD22 后进行 (9)，否则直接进行 (9)；
- (9) 如果 $pp3=2$ ，输出 CXD33 后进行 (14)，否则输出 CXD31 后进行 (10)；
- (10) 如果 $F3=1$ ，输出 CXD32 后进行 (14)，否则直接进行 (14)；
- (11) 如果 $RL=0$ ，进行 (4)，否则进行 (12)；
- (12) 如果 $RL=1$ ，进行 (6)，否则进行 (13)；
- (13) 如果 $RL=2$ ，进行 (8)，否则进行 (10)；

(14) 结束样本的上下文输出。

图 7 给出了本发明的位平面编码电路。按照图 1 给出的扫描方式在每个时钟周期并行输入小波系数一个条带列的 4 个字到输入数据缓冲寄存器单元,输入数据缓冲寄存器单元用来存放当前条带当前输入的如图 3 所示的 4 行 \times 4 列个系数以及前一条带第 4 行和当前条带第 4 行移出的系数值,每个时钟周期并行输入 4 个新系数。编码通道预测逻辑电路按照上面介绍的预测编码通道的方法产生相应列系数的编码通道值。编码通道状态寄存器单元用来存储当前条带的当前上下文形成列及其前一系列共 8 个系数的编码通道值,以及前一条带第 4 行的系数和当前条带第 4 行移出的系数的最高非零幅度位的编码通道值。邻域重要性贡献生成电路根据前面介绍的方法,在每个时钟周期并行产生当前被编码的 4 个系数上下文所需要的邻域重要性贡献。

图 8 给出了图 7 中输入数据缓冲寄存器单元的内部电路,包括 4 个 4 单位延时移位寄存器组和一个 N 单位延时移位寄存器组,每个 4 单位延时移位寄存器组由 4 个寄存器串联构成,N 单位延时移位寄存器组由 N 个寄存器串联构成,分别用来存储输入的当前条带的 4 行 \times 4 列系数和前一条带第 4 行的有用系数与当前条带的第 4 行移出的系数;输入 In1 连接到第一 4 单位延时移位寄存器组的寄存器 R11 的输入端,R11 的输出 x11 连接到寄存器 R12 的输入端,R12 的输出 x12 连接到寄存器 R13 的输入端,R13 的输出 x13 连接到寄存器 R14 的输入端,R14 的输出为 x14;输入 In2 连接到第二 4 单位延时移位寄存器组的寄存器 R21 的输入端,R21 的输出 x21 连接到寄存器 R22 的输入端,R22 的输出 x22 连接到寄存器 R23 的输入端,R23 的输出 x23 连接到寄存器 R24 的输入端,R24 的输出为 x24;输入 In3 连接到第三 4 单位延时移位寄存器组的寄存器 R31 的输入端,R31 的输出 x31 连接到寄存器 R32 的输入端,R32 的输出 x32 连接到寄存器 R33 的输入端,R33 的输出 x33 连接到寄存器 R34

的输入端，R34 的输出为 x34；输入 In4 连接到第四 4 单位延时移位寄存器组的寄存器 R41 的输入端，R41 的输出 x41 连接到寄存器 R42 的输入端，R42 的输出 x42 连接到寄存器 R43 的输入端，R43 的输出 x43 连接到寄存器 R44 的输入端，R44 的输出 x44 连接到 N 单位延时移位寄存器组中由 N-3 个寄存器串联组成的移位寄存器组 DR01 (N 表示编码子块的宽度)，DR01 的输出 x01 连接到寄存器 R02，R02 的输出 x02 连接到寄存器 R03，R03 的输出 x03 连接到寄存器 R04，R04 的输出为 x04。

图 9 给出了图 7 中编码通道预测逻辑电路，包括并行工作的 4 个处理器单元 PE0、PE1、PE2、PE3，每个处理器单元输入当前被预测编码通道的系数及其 8 邻域系数、8 邻域系数中已预测的编码通道值，产生被预测编码通道系数的编码通道预测值，4 个处理器单元在单时钟周期并行产生一系列 4 个系数除最高非零位平面位外的所有幅度位的编码通道预测值。当前被预测编码通道的系数 x12 及其 8 邻域系数 x01、x02、x03、x11、x13、x21、x22、x23 以及系数 x01、x02、x03 已预测的最高非零幅度位的编码通道值 p1m、p2m、p3m 和系数 x13、x23 已预测的编码通道值 cpp12、cpp22 分别连接到 PE0 的输入端，PE0 的输出为系数 x12 的所有幅度位（最高非零位平面位除外）样本的编码通道预测值 cpp11。当前被预测编码通道的系数 x22 及其 8 邻域系数 x11、x12、x13、x21、x23、x31、x32、x33 以及 x12、x13、x23、x33 已预测的编码通道值 cpp11、cpp12、cpp22、cpp32 分别连接到 PE1 的输入端，PE1 的输出为系数 x22 的所有幅度位（最高非零位平面位除外）样本的编码通道预测值 cpp21。当前被预测编码通道的系数 x32 及其 8 邻域系数 x21、x22、x23、x31、x33、x41、x42、x43 以及 x22、x23、x33、x43 已预测的编码通道值 cpp21、cpp22、cpp32、cpp42 分别连接到 PE2 的输入端，PE2 的输出为系数 x32 的所有幅度位（最高非零位平面位除外）样本的编码通道预测值 cpp31。当前被预测编码通道的系数 x42 及其 8 邻域系数 x31、x32、x33、x41、x43 以

及 x_{32} 、 x_{33} 、 x_{43} 已预测的编码通道值 cpp_{31} 、 cpp_{32} 、 cpp_{42} 分别连接到 PE3 的输入端，PE3 的输出为系数 x_{32} 的所有幅度位（最高非零位平面位除外）样本的编码通道预测值 cpp_{41} 。

图 10 给出了图 7 中编码通道状态寄存器单元的内部电路，包括 4 组 2 单位延时寄存器组和一个 $N+2$ 单位延时移位寄存器组；每组 2 单位延时寄存器组由 2 个寄存器串联构成，用来存储当前被编码条带列与其前一行条带列的 1 行 2 个系数所有幅度位（最高非零位平面位除外）的编码通道值； $N+2$ 单位延时移位寄存器组由 $N+2$ 个寄存器串联构成，分别用来存储当前被编码条带列的第 4 系数与其前一行条带列的第 4 系数以及缓存在数据缓冲寄存器单元的 N 单位延时移位寄存器的 $N+2$ 个系数的最高非零幅度位的编码通道值。 cpp_{11} 连接到第一组 2 单位延时寄存器组的寄存器 R1 的输入端，R1 的输出 cpp_{12} 连接到寄存器 R5 的输入端，R5 的输出为 cpp_{13} ； cpp_{21} 连接到第二组 2 单位延时寄存器组的寄存器 R2 的输入端，R2 的输出 cpp_{22} 连接到寄存器 R6 的输入端，R6 的输出为 cpp_{23} ； cpp_{31} 连接到第三组 2 单位延时寄存器组的寄存器 R3 的输入端，R3 的输出 cpp_{32} 连接到寄存器 R7 的输入端，R7 的输出为 cpp_{33} ； cpp_{41} 连接到第四组 2 单位延时寄存器组的寄存器 R4 的输入端，R4 的输出 cpp_{42} 连接到寄存器 R8 的输入端，R8 的输出为 cpp_{43} ； cpp_{4m} 连接到 $N+2$ 单位延时移位寄存器组中由 $N-1$ 个寄存器串联组成的移位寄存器组 PR01 的输入端（ N 表示编码子块的宽度），PR01 的输出 p_{1m} 连接到寄存器 R51 的输入端，R51 的输出 p_{2m} 连接到寄存器 R52，R52 的输出 p_{3m} 连接到寄存器 R53，R53 的输出为 p_{4m} 。

图 11 给出了本发明图 7 中邻域重要性贡献生成电路，包括并行工作的处理器单元 PE5、PE6、PE7、PE8，每个处理器单元输入当前被编码系数及其 8 邻域系数、以及它们的编码通道值，产生被编码系数的 8 邻域重要性贡献，4 个处理器单元在单时钟周期内并行地生成当前被编码条

带列 4 个系数的所有位样本的上下文形成时需要的邻域重要性贡献。当前被编码系数 x_{13} 、其 8 邻域系数 x_{02} 、 x_{03} 、 x_{04} 、 x_{12} 、 x_{14} 、 x_{22} 、 x_{23} 、 x_{24} 和 x_{02} 、 x_{03} 、 x_{04} 的最高非零幅度位的编码通道值 p_{2m} 、 p_{3m} 、 p_{4m} ，以及 x_{12} 、 x_{14} 、 x_{22} 、 x_{23} 、 x_{24} 的编码通道值 cpp_{11} 、 cpp_{12} 、 cpp_{13} 、 cpp_{21} 、 cpp_{22} 、 cpp_{23} 连接到处理器 PE5 的输入端，PE5 输出当前被编码系数 x_{13} 在水平方向邻域的幅度重要性贡献 σ_{0h} 、在垂直方向邻域的幅度重要性贡献 σ_{0v} 、在对角方向邻域的幅度重要性贡献 σ_{0d} 、在水平方向邻域符号的重要性贡献 σ_{0h}^x 以及在垂直方向邻域符号的重要性贡献 σ_{0v}^x 。当前被编码系数 x_{23} 及其 8 邻域系数 x_{12} 、 x_{13} 、 x_{14} 、 x_{22} 、 x_{24} 、 x_{32} 、 x_{33} 、 x_{34} 以及它们的编码通道值 cpp_{11} 、 cpp_{12} 、 cpp_{13} 、 cpp_{21} 、 cpp_{22} 、 cpp_{23} 、 cpp_{31} 、 cpp_{32} 、 cpp_{33} 连接到处理器 PE6 的输入端，PE6 输出当前被编码系数 x_{23} 在水平方向邻域的幅度重要性贡献 σ_{1h} 、在垂直方向邻域的幅度重要性贡献 σ_{1v} 、在对角方向邻域的幅度重要性贡献 σ_{1d} 、在水平方向邻域符号的重要性贡献 σ_{1h}^x 以及在垂直方向邻域符号的重要性贡献 σ_{1v}^x 。当前被编码系数 x_{33} 、其 8 邻域系数 x_{22} 、 x_{23} 、 x_{24} 、 x_{32} 、 x_{34} 、 x_{42} 、 x_{43} 、 x_{44} 以及它们的编码通道值 cpp_{21} 、 cpp_{22} 、 cpp_{23} 、 cpp_{31} 、 cpp_{32} 、 cpp_{33} 、 cpp_{41} 、 cpp_{42} 、 cpp_{43} 连接到处理器 PE7 的输入端，PE7 输出当前被编码系数 x_{33} 在水平方向邻域的幅度重要性贡献 σ_{2h} 、在垂直方向邻域的幅度重要性贡献 σ_{2v} 、在对角方向邻域的幅度重要性贡献 σ_{2d} 、在水平方向邻域符号的重要性贡献 σ_{2h}^x 以及在垂直方向邻域符号的重要性贡献 σ_{2v}^x 。当前被编码系数 x_{43} 、其 5 邻域系数 x_{32} 、 x_{33} 、 x_{34} 、 x_{42} 、 x_{44} 以及它们的编码通道值 cpp_{31} 、 cpp_{32} 、 cpp_{33} 、 cpp_{41} 、 cpp_{42} 、 cpp_{43} 连接到处理器 PE8 的输入端，PE8 输出当前被编码系数 x_{43} 在水平方向邻域的幅度重要性贡献 σ_{3h} 、在垂直方向邻域的幅度重要性贡献 σ_{3v} 、在对角方向邻域的幅度重要性贡献 σ_{3d} 、在水平方向邻域符号的重要性贡献 σ_{3h}^x 以及在垂直方向邻域符号的重要性贡献 σ_{3v}^x 。

上下文形成逻辑电路单元的内部电路如图 12，包括 4 组单字编码模块、1 个游程编码原语模块、输出端口以及控制电路，每组单字编码模块由零编码原语模块、幅度细化编码原语模块和符号编码原语模块组成；用来根据上面介绍的上下文形成方法，在每个时钟周期并行产生 4 个当前被编码系数 x_{13} 、 x_{23} 、 x_{33} 、 x_{43} 的在每个幅度位平面的所有可能的 15 个上下文环境和判决数据对 CXD01、CXD02、CXD03、CXD11、CXD12、CXD13、CXD21、CXD22、CXD23、CXD31、CXD32、CXD33、CXD4、CXD5、CXD6，通过输出端口从 15 个上下文环境和判决数据对中选择输出 1 到 10 个有效的上下文环境和判决数据对 CXD。在第一组单字编码模块，被编码系数 x_{13} 的幅度值以及 x_{13} 的邻域在水平方向、垂直方向和对角线方向的重要性贡献 σ_{0h} 、 σ_{0v} 和 σ_{0d} 输入到零编码原语模块 ZC0，ZC0 输出对系数 x_{13} 执行零编码产生的上下文环境判决数据对 CXD01 到输出端口。被编码系数 x_{13} 的幅度值以及 x_{13} 的邻域在水平方向、垂直方向和对角线方向的重要性贡献 σ_{0h} 、 σ_{0v} 和 σ_{0d} 输入到幅度细化编码原语模块 MRC0，MRC0 输出对系数 x_{13} 执行幅度细化编码产生的上下文环境和判决数据对 CXD03 到输出端口。被编码系数 x_{13} 的符号位以及 x_{13} 的邻域符号的重要性贡献 σ_{0h}^x 和 σ_{0v}^x 输入到符号编码原语模块 SC0，SC0 输出对系数 x_{13} 执行符号编码产生的上下文环境和判决数据对 CXD02 到输出端口。在第二组单字编码模块，被编码系数 x_{23} 的幅度值以及 x_{23} 的邻域的重要性贡献 σ_{1h} 、 σ_{1v} 和 σ_{1d} 输入到零编码原语模块 ZC1，ZC1 输出对系数 x_{23} 执行零编码产生的上下文环境和判决数据对 CXD11 到输出端口。被编码系数 x_{23} 的幅度值以及 x_{23} 的邻域的重要性贡献 σ_{1h} 、 σ_{1v} 和 σ_{1d} 输入到幅度细化编码原语模块 MRC1，MRC1 输出对系数 x_{23} 执行幅度细化编码产生的上下文环境和判决数据对 CXD13 到输出端口。被编码系数 x_{23} 的符号位以及 x_{23} 的邻域符号的重要性贡献 σ_{1h}^x 和 σ_{1v}^x 输入到符号编码原语模块 SC1，SC1 输出对系数 x_{23} 执行符号编码产生的上下文环境和

判决数据对 CXD12 到输出端口。在第三组单字编码模块,被编码系数 x_{33} 的幅度值以及 x_{33} 的邻域的重要性贡献 σ_{2h} 、 σ_{2v} 和 σ_{2d} 输入到零编码原语模块 ZC2, ZC2 输出对系数 x_{33} 执行零编码产生的上下文环境和判决数据对 CXD21 到输出端口。被编码系数 x_{33} 的幅度值以及 x_{33} 的邻域的重要性贡献 σ_{2h} 、 σ_{2v} 和 σ_{2d} 输入到幅度细化编码原语模块 MRC2, MRC2 输出对系数 x_{33} 执行幅度细化编码产生的上下文环境和判决数据对 CXD23 到输出端口。被编码系数 x_{33} 的符号位以及 x_{33} 的邻域符号的重要性贡献 σ_{2h}^x 和 σ_{2v}^x 输入到符号编码原语模块 SC2, SC2 输出对系数 x_{33} 执行符号编码产生的上下文环境判决数据对 CXD22 到输出端口。在第四组单字编码模块,被编码系数 x_{43} 的幅度值以及 x_{43} 的邻域的重要性贡献 σ_{3h} 、 σ_{3v} 和 σ_{3d} 输入到零编码原语模块 ZC3, ZC3 输出对系数 x_{43} 执行零编码产生的上下文环境和判决数据对 CXD31 到输出端口。被编码系数 x_{43} 的幅度值以及 x_{43} 的邻域的重要性贡献 σ_{3h} 、 σ_{3v} 和 σ_{3d} 输入到幅度细化编码原语模块 MRC3, MRC3 输出对系数 x_{43} 执行幅度细化编码产生的上下文环境和判决数据对 CXD33 到输出端口。被编码系数 x_{43} 的符号位以及 x_{43} 的邻域符号的重要性贡献 σ_{3h}^x 和 σ_{3v}^x 输入到符号编码原语模块 SC3, SC3 输出对系数 x_{43} 执行符号编码产生的上下文环境和判决数据对 CXD32 到输出端口。游程编码原语模块 RLC 的输入为被编码系数 x_{13} 、 x_{23} 、 x_{33} 、 x_{43} 的幅度值和游程编码标志信号 rlc , 输出当前条带列执行游程编码生成的上下文环境和判决数据对 CXD4、CXD5 和 CXD6 到输出端口。控制电路的输入为当前被编码系数 x_{13} 、 x_{23} 、 x_{33} 、 x_{43} , 被编码系数的编码通道值 cpp_{12} 、 cpp_{22} 、 cpp_{32} 、 cpp_{42} 以及每个被编码系数的邻域重要性贡献 σ_{ih} ($i=0,1,2,3$)、 σ_{iv} ($i=0,1,2,3$)、 σ_{id} ($i=0,1,2,3$), 控制电路判定对应条带列是否进行游程编码并产生游程编码标志信号 rlc , 同时产生相应的一组输出控制信号 $ctrl$ 到输出端口, 控制输出端口选择输出有效的上下文环境和判决数据对 CXD。

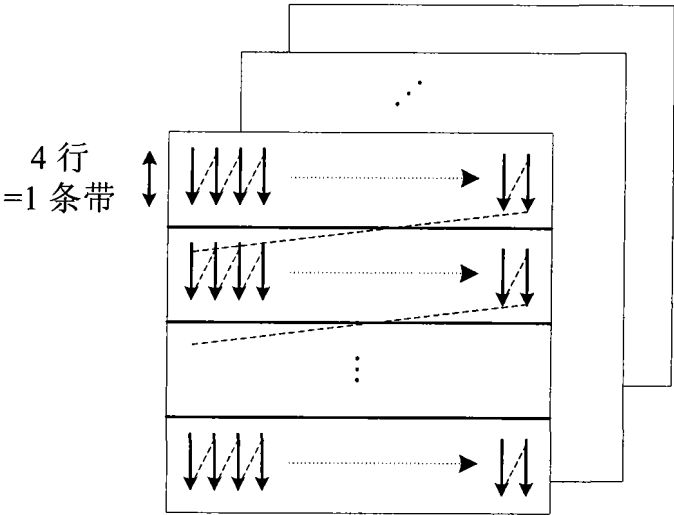


图 1

| | | |
|----|----|----|
| D0 | V0 | D1 |
| H0 | X | H1 |
| D2 | V1 | D3 |

图 2

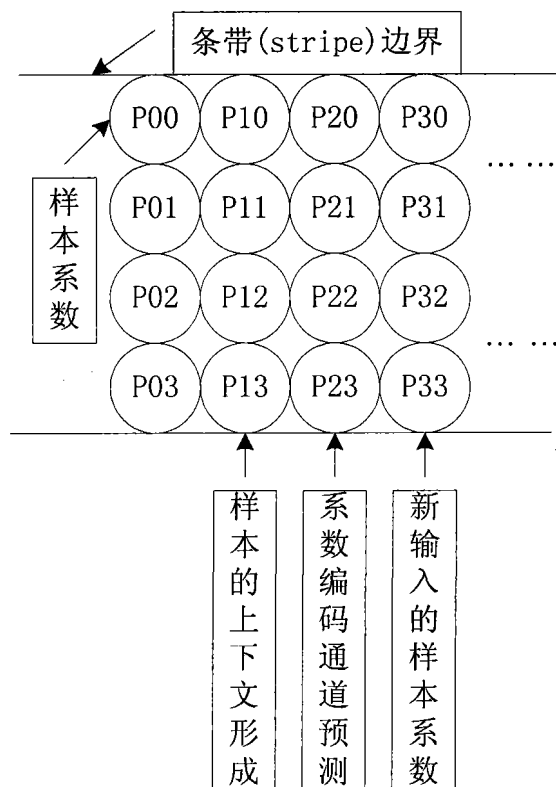


图 3

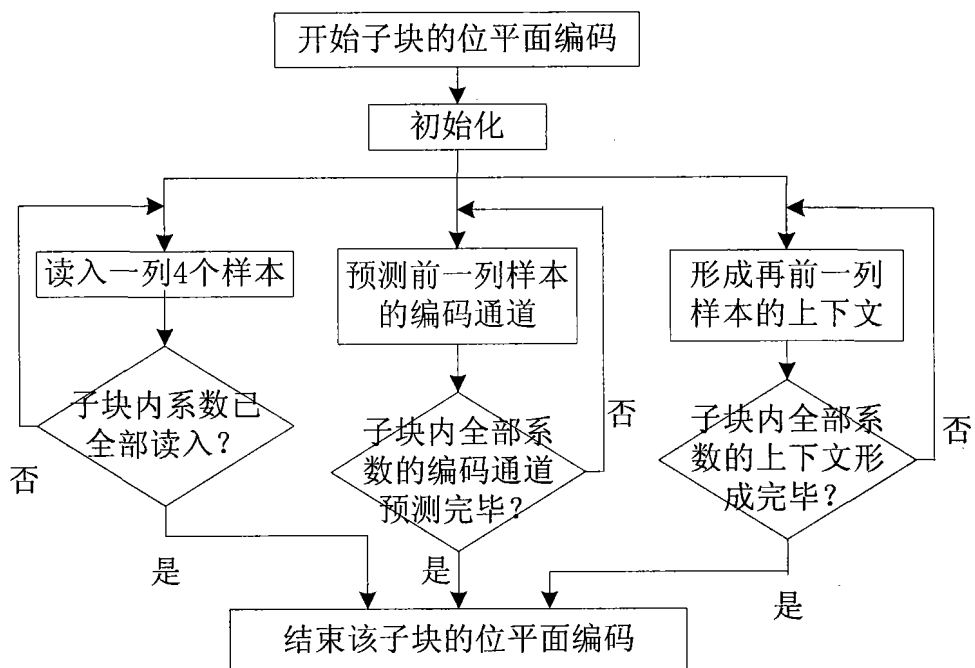


图 4

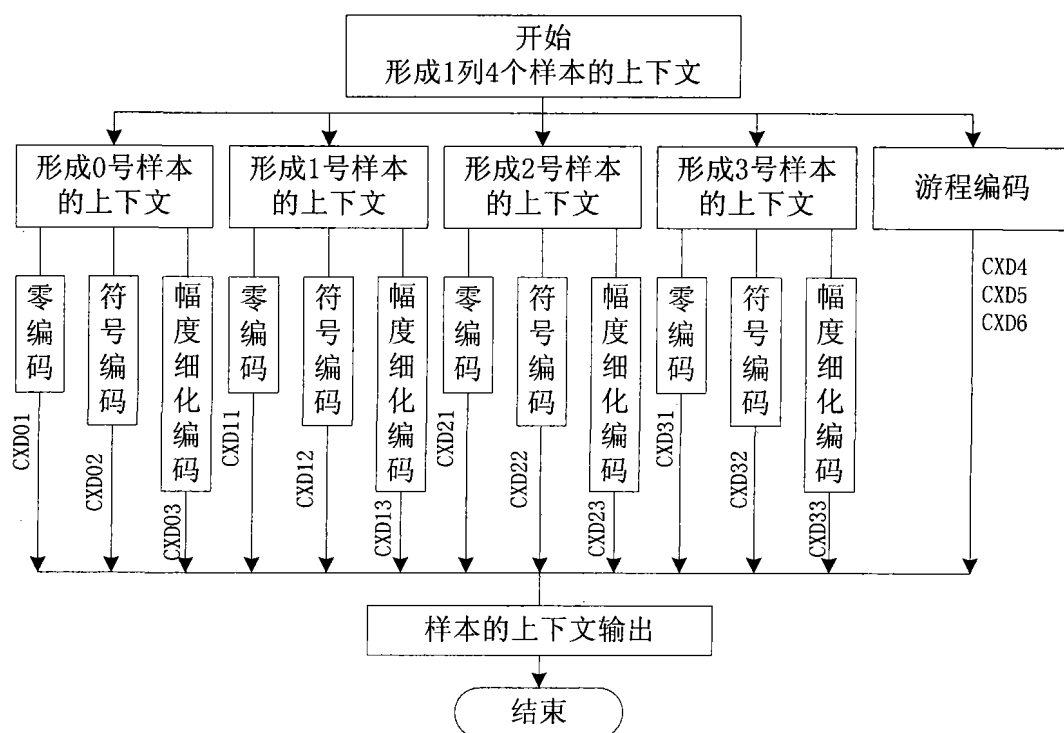


图 5

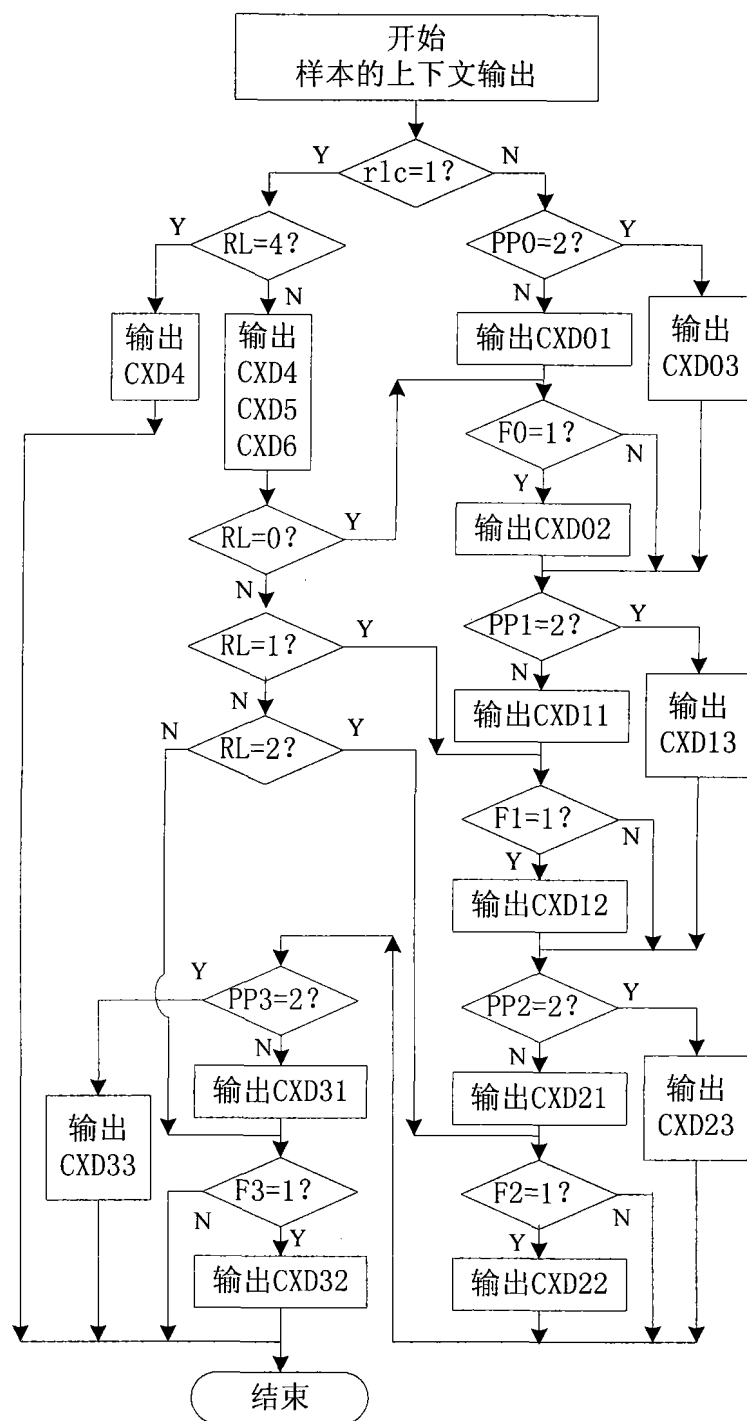


图 6

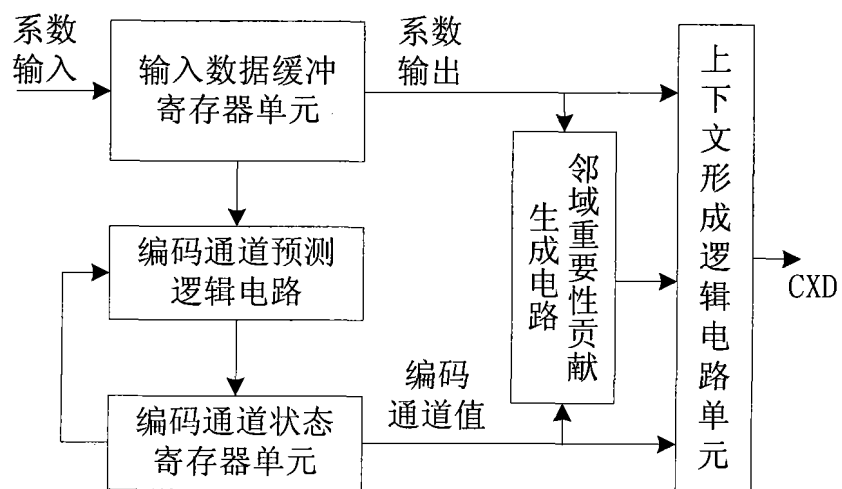


图 7

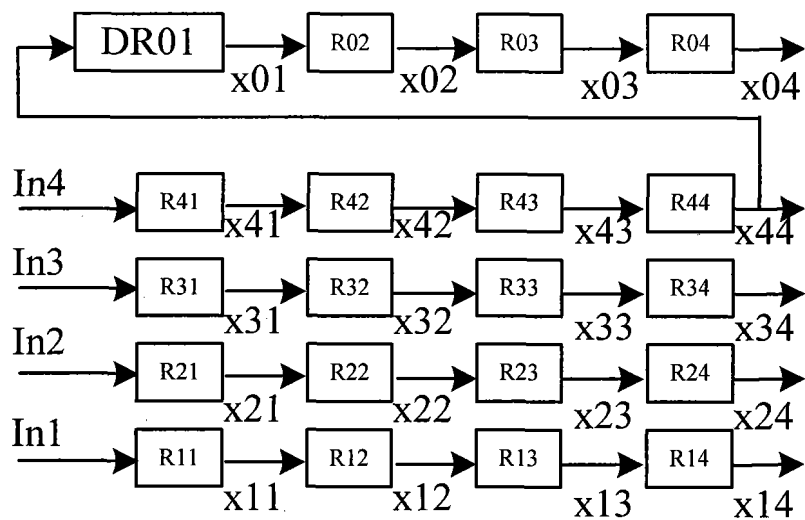


图 8

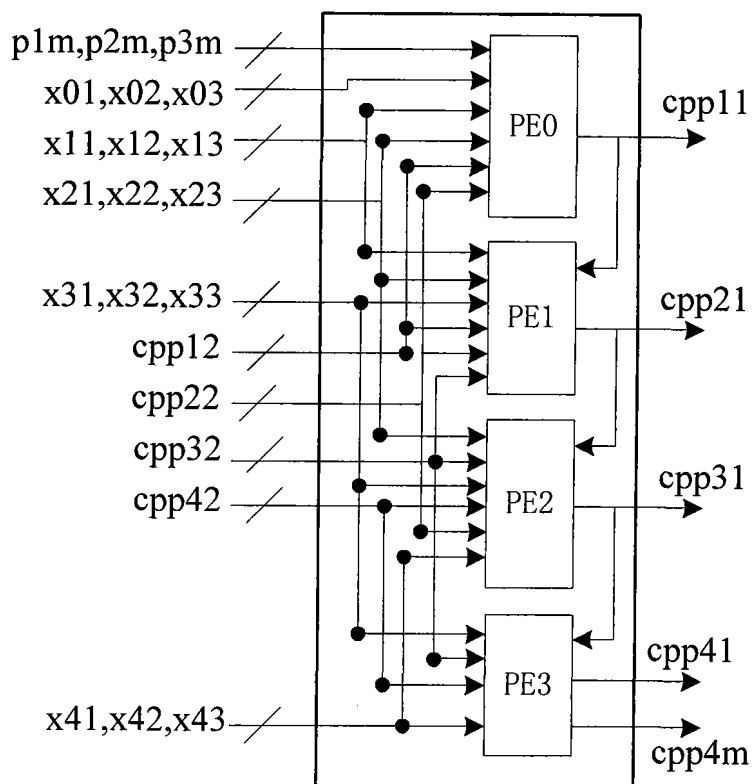


图 9

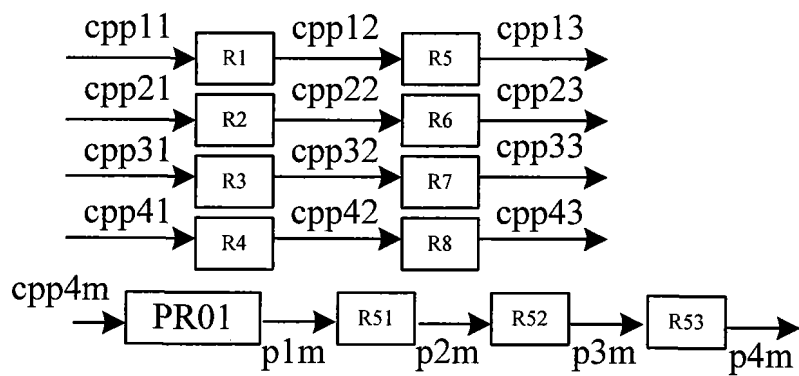


图 10

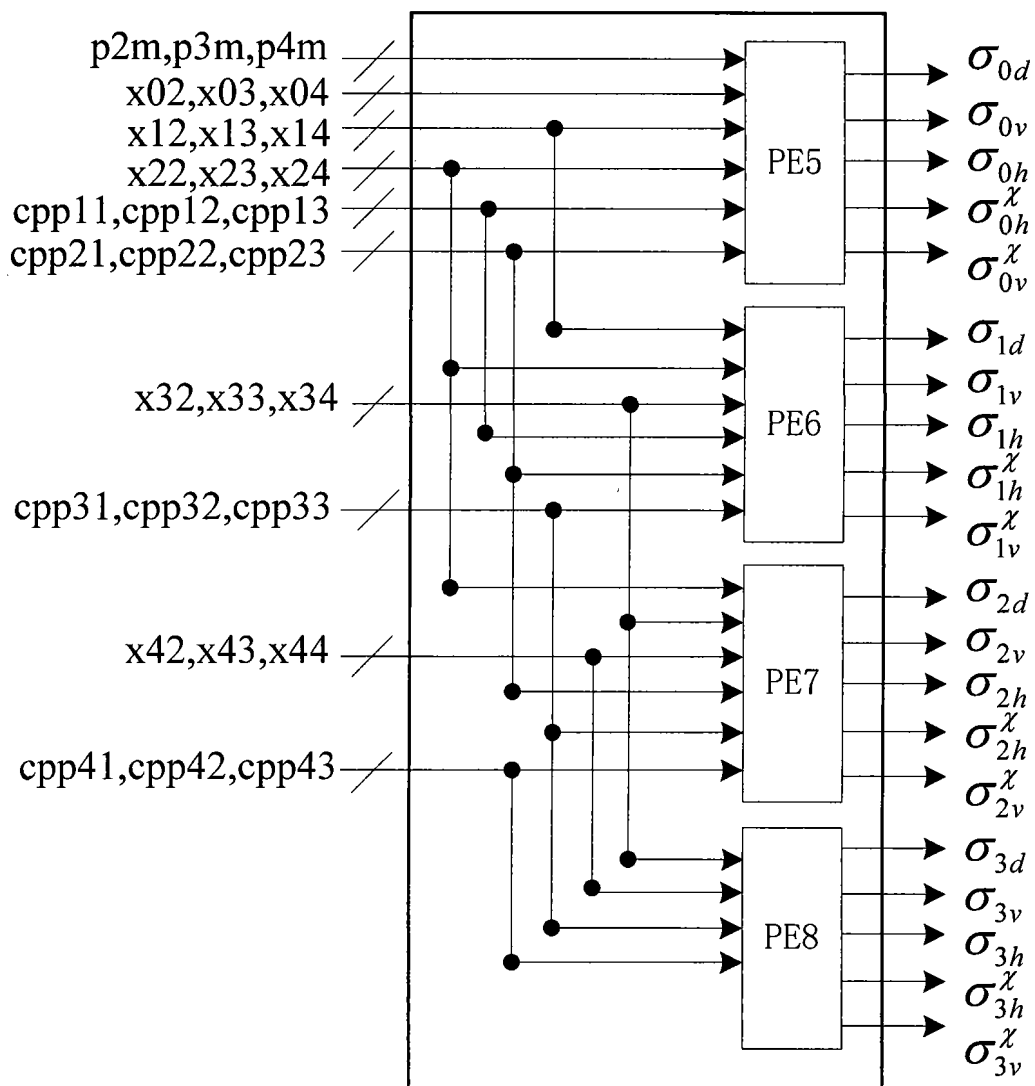


图 11

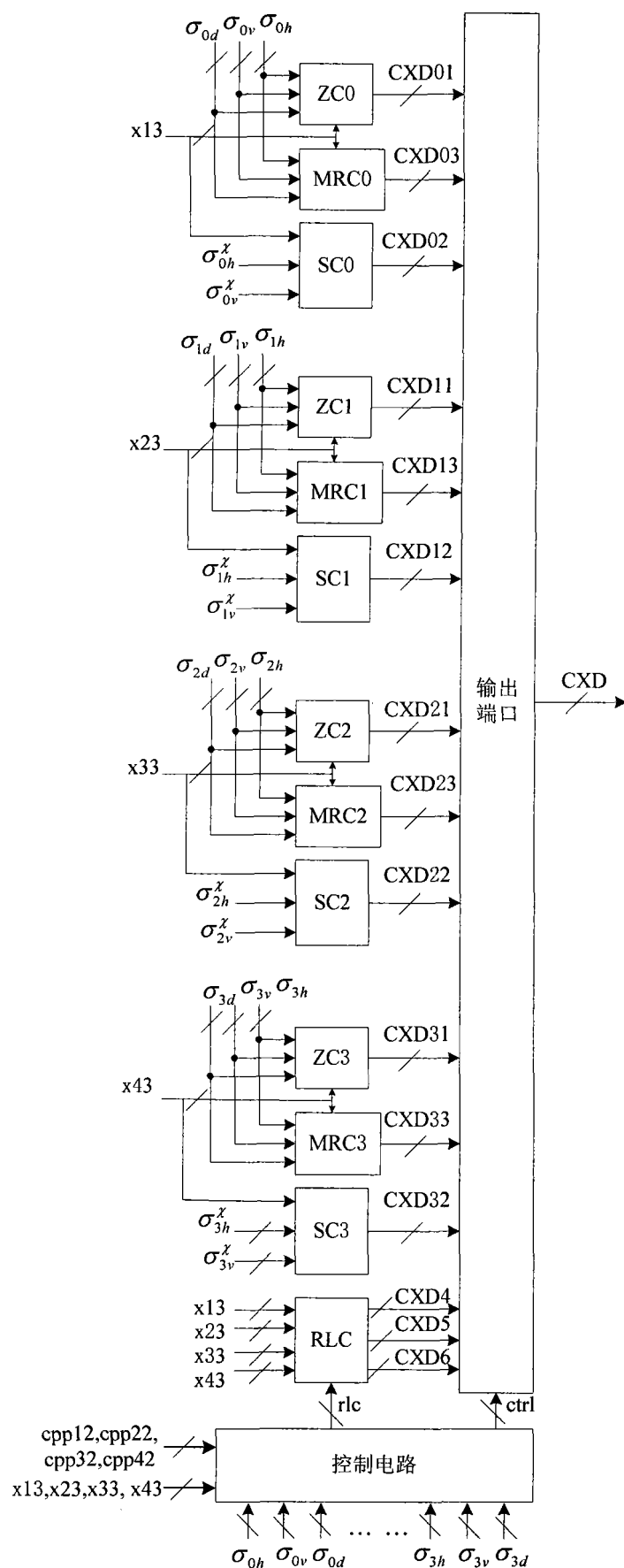


图 12