



(12) 发明专利

(10) 授权公告号 CN 101438595 B

(45) 授权公告日 2011. 05. 25

(21) 申请号 200780016046. 0

(22) 申请日 2007. 03. 14

(30) 优先权数据

128093/2006 2006. 05. 02 JP

(85) PCT申请进入国家阶段日

2008. 11. 03

(86) PCT申请的申请数据

PCT/JP2007/055784 2007. 03. 14

(87) PCT申请的公布数据

W02007/129508 JA 2007. 11. 15

(73) 专利权人 索尼株式会社

地址 日本东京

(72) 发明人 小仓誉之 重本大乘

(74) 专利代理机构 北京康信知识产权代理有限

责任公司 11240

代理人 余刚 吴孟秋

(51) Int. Cl.

H04N 7/26 (2006. 01)

(56) 对比文件

JP 2002218474 A, 2002. 08. 02, 全文.

JP 2005223533 A, 2005. 08. 18, 全文.

JP 2006054865 A, 2006. 02. 23, 全文.

JP 2005130099 A, 2005. 05. 19, 全文.

JP 2004040261 A, 2004. 02. 05, 全文.

审查员 张春

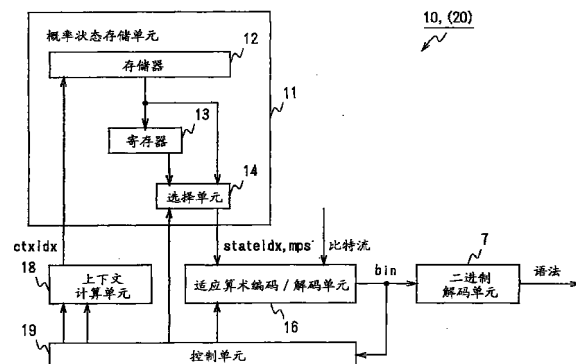
权利要求书 2 页 说明书 15 页 附图 12 页

(54) 发明名称

运动图像处理方法及运动图像处理设备

(57) 摘要

例如, 本发明应用于基于 ITU-T H. 264 的运动图像编码设备和解码设备。使用在访问等待时间端的第二存储器 (13) 中保持的概率状态变量来处理具有高出现频率的语法元素, 并使用在访问等待时间长的第一存储器 (12) 中保持的概率状态变量来处理其它语法元素。



1. 一种运动图像处理方法,用于计算上下文并编码或解码运动图像,其特征在于,包括:

上下文计算处理步骤,计算构成所述运动图像的语法元素的上下文,并顺序地检测能够被所述语法元素采用的概率状态变量;以及

概率状态变量处理步骤,顺序地选择并处理在所述上下文计算处理步骤中获得的所述概率状态变量,并编码或解码所述语法元素,

其中,所述概率状态变量处理步骤包括:

概率状态变量选择步骤,选择在概率状态存储单元中保持的所述概率状态变量,

其中,在所述概率状态存储单元中设置第一存储器和与所述第一存储器相比访问等待时间短的第二存储器,以及

其中,所述概率状态变量选择步骤

当处理具有低出现频率的语法元素时,从所述第一存储器中顺序地选择所述概率状态变量,以及

当处理具有高出现频率的语法元素时,从所述第二存储器中顺序地选择所述概率状态变量。

2. 根据权利要求1所述的运动图像处理方法,其特征在于,具有高出现频率的语法元素是通过以预定块为单位对形

成所述运动图像的图像数据执行正交转换处理所获得的系数数据。

3. 根据权利要求1所述的运动图像处理方法,其特征在于,

所述第二存储器是寄存器。

4. 根据权利要求1所述的运动图像处理方法,其特征在于,

所述第一存储器

保持处理所有语法元素所需的所述概率状态变量,以及

所述运动图像处理方法包括:

第二存储器写入步骤,读取保持在所述第一存储器中的概率状态变量的一部分,并在所述第二存储器中存储所述概率状态变量的读取部分。

5. 根据权利要求4所述的运动图像处理方法,其特征在于,

所述上下文计算处理步骤

在所述概率状态变量处理步骤中正在处理所述概率状态变量的期间内,计算后续语法元素的上下文,并检测所述后续语法元素的概率状态变量,以及

所述第二存储器写入步骤

在所述概率状态变量处理步骤中正在处理所述概率状态变量的期间内,在所述第二存储器中存储在所述上下文计算处理步骤中检测出的所述后续语法元素的概率状态变量。

6. 根据权利要求5所述的运动图像处理方法,其特征在于,

在具有高出现频率的语法元素中能够被检测的概率状态变量中,在所述第二存储器写入步骤中的处理中,提供了预处理步骤,所述预处理步骤预先将对于在所述概率状态变量处理步骤中开始所述后续语法元素的概率状态变量的处理来说太迟的概率状态变量从所述第一存储器存储至所述第二存储器。

7. 一种计算上下文并编码或解码运动图像的运动图像处理设备,

其特征在于,包括:

上下文计算单元,用于计算构成所述运动图像的语法元素的上下文,并顺序地检测能够被所述语法元素采用的概率状态变量;

概率状态存储单元,其中设置了第一存储器和与所述第一存储器相比访问等待时间短的第二存储器,所述概率状态变量被保持在所述第一存储器和所述第二存储器中;以及

概率状态变量处理单元,用于基于所述上下文计算单元的检测结果顺序地选择并处理来自所述概率状态存储单元的概率状态变量,并编码或解码所述语法元素,

其中,所述概率状态变量处理单元

当处理具有低出现频率的语法元素时,从所述第一存储器中顺序地选择所述概率状态变量,以及

当处理具有高出现频率的语法元素时,从所述第二存储器中顺序地选择所述概率状态变量。

运动图像处理方法及运动图像处理设备

技术领域

[0001] 本发明涉及运动图像处理方法、运动图像处理方法的程序、其上记录有运动图像处理方法的程序的记录介质以及运动图像处理设备,例如可应用于基于 ITU(国际电信联盟)-T H. 264 的运动图像的编码设备和解码设备。本发明使用保持在访问等待时间较小的第二存储器中的概率状态变量来处理具有高出现频率的语法元素,并使用保持在访问等待时间较大的第一存储器中的概率状态变量来处理其它语法元素,因此与过去相比避免了整体结构尺寸的增加并提高了处理速度。

背景技术

[0002] 至今,运动图像编码处理已经应用了熵编码处理的技术,从而有效地执行运动图像的数据压缩。此外,在 MPEG(运动图片专家组)2、MPEG4 等中,可变长度编码处理被应用于这种熵编码处理。此外,在 ITU-T H. 264 中,基于上下文的自适应二进制算术编码处理(CABAC)被应用于这种熵编码处理。

[0003] 这里,由于基于上下文的自适应二进制算术编码处理与可变长度编码处理相比可以有效地执行数据压缩,所以当以一定的比特率执行数据传输时,与 MPEG2 和 MPEG4 中的编码处理相比,在 ITU-T H. 264 中可以通过编码处理改善图像质量。但是,基于上下文的自适应二进制算术编码处理具有弱点,即,与可变长度编码处理相比,它涉及到复杂的处理和较大的处理负荷。

[0004] 图 1 是示出基于上下文的自适应二进制算术编码处理的处理程序的流程图。基于上下文的自适应二进制算术编码处理对连续的语法元素的每一个顺序执行该处理。通过执行图 1 中的处理程序,基于上下文的自适应二进制算术编码处理对连续的语法元素执行上下文计算,以顺序检测各个语法元素可以采用的概率状态变量。此外,基于上下文的自适应二进制算术编码处理顺序地选择并处理所检测的概率状态变量,以对语法元素进行编码。

[0005] 即,当基于上下文的自适应二进制算术编码处理开始该处理时,基于上下文的自适应二进制算术编码处理从步骤 SP1 前进至步骤 SP2,并获得将被处理的多值语法元素(syntax element)。此后,通过步骤 SP3 中的二进制处理,基于上下文的自适应二进制算术编码处理根据每个语法元素的类型的规则将在步骤 SP2 中获得的语法元素二进制化。

[0006] 此后,通过步骤 SP4 中的上下文计算处理,基于上下文的自适应二进制算术编码处理获得用于步骤 SP3 中所转换的各个二进制值的位置的上下文索引(ctxIdx)。这里,上下文索引(ctxIdx)是指定概率状态变量的索引。概率状态变量是表示二进制值的每个值的出现频率的变量。通过对应于每个二进制值的值的 MPS(最或然符号)和状态索引(stateIdx)来表示概率状态变量。这里,MPS(最或然符号)是表示所述值是否为具有高发生概率的符号的标记,并且状态索引(stateIdx)是表示 MPS 符号的发生概率表的索引。因此,基于上下文的自适应二进制算术编码处理通过步骤 SP4 中的上下文计算处理根据上下文索引(ctxIdx)获得将被处理的语法元素的每个二进制值可以采用的概率状态变量。

[0007] 随后,在步骤 SP5 的概率预测处理中,基于上下文的自适应二进制算术编码处

理基于在步骤 SP4 中获得的内容索引 (ctxIdx) 选择对应于将被处理的二进制值的值的 MPS(最或然符号) 和状态索引 (stateIdx)。

[0008] 此后,在步骤 SP6 的算术编码处理中,基于上下文的自适应二进制算术编码处理使用在步骤 SP5 中选择的 MPS(最或然符号) 和状态索引 (stateIdx) 来执行算术编码处理。

[0009] 基于上下文的自适应二进制算术编码处理从低位侧的二进制值开始对所有的二进制值重复步骤 SP4 ~ SP5 ~ SP6 ~ SP4 的处理程序,直至完成所有待被编码的二进制值的处理为止,因此,顺序生成比特流。此外,当完成所有待被编码的二进制值的处理时,处理从步骤 SP6 前进至步骤 SP7,并输出生成的比特流。处理前进至步骤 SP8,处理程序结束。

[0010] 此外,如通过从步骤 SP6 至步骤 SP4 的箭头所示,当基于上下文的自适应二进制算术编码处理处理一个语法元素时,基于上下文的自适应二进制算术编码处理基于处理结果更新保持在存储器中的概率状态变量。当对相同的语法元素执行编码处理时,基于上下文的自适应二进制算术编码处理使用经过更新的概率状态变量来执行编码处理。

[0011] 相反,图 2 是示出基于上下文的自适应二进制算术解码处理的流程图。基于上下文的自适应二进制算术解码处理对每个语法元素执行该处理。如在基于上下文的自适应二进制算术编码处理中一样,基于上下文的自适应二进制算术解码处理对连续的语法元素执行上下文计算,以顺序地检测各个语法元素可以采用的概率状态变量。此外,基于上下文的自适应二进制算术解码处理顺序地选择并处理所检测的概率状态变量,从而对语法元素进行解码。

[0012] 即,当基于上下文的自适应二进制算术解码处理开始该处理程序时,基于上下文的自适应二进制算术解码处理从步骤 SP11 前进至步骤 SP12。这里,基于上下文的自适应二进制算术解码处理从待被执行解码处理的比特流中检测指定语法元素类型的语法模式 (syntax mode) 和作为指定待被处理的比特位置的信息的 bin。

[0013] 此外,在此后的步骤 SP13 中,如在基于上下文的自适应二进制算术编码处理中一样,基于上下文的自适应二进制算术解码处理使用在步骤 SP12 中获得的信息执行上下文计算,以获得对应的上下文索引 (ctxIdx)。使用该上下文索引 (ctxIdx),如在基于上下文的自适应二进制算术编码处理中一样,基于上下文的自适应二进制算术解码处理顺序地获得 MPS 和状态索引 (stateIdx)。此外,通过使用所获得的 MPS 和状态索引 (stateIdx) 的算术编码处理,与执行编码时相反,基于上下文的自适应二进制算术解码处理获得原始二进制值。在步骤 SP17 中,基于上下文的自适应二进制算术解码处理从二进制值中获得多值数,以对原始语法元素进行解码。此外,如在基于上下文的自适应二进制算术编码处理中一样,基于上下文的自适应二进制算术解码处理基于处理结果更新概率状态变量。当对相同的语法元素执行解码处理时,基于上下文的自适应二进制算术解码处理使用经过更新的概率状态变量来执行解码处理。

[0014] 图 3 是示出执行基于上下文的自适应二进制算术解码处理的解码设备的框图。

[0015] 在该解码设备 1 中,在控制单元 3 的控制下,上下文计算单元 2 从待被处理的比特流中活动上下文索引 (ctxIdx)。概率状态存储单元 4 在存储器 5 中存储并保持由这些上下文索引 (ctxIdx) 所指定的概率状态变量。响应于来自适应算术编码 / 解码单元 6 的访问,概率状态存储单元 4 向适应算术编码 / 解码单元 6 通知所保持的概率 状态变量。例如,注

意,在系数数据处理中,在存储器 5 中存储 7 比特 $\times$ 59 种类型的概率状态变量。

[0016] 适应算术编码 / 解码单元 6 通过待被处理的比特流生成二进制化的数据,并通过使用二进制化数据的概率预测,顺序地选择状态索引 (stateIdx) 和 MPS。适应算术编码 / 解码单元 6 通过处理状态索引 (stateIdx) 和 MPS 生成语法信息,并执行算术解码处理。基于适应算术编码 / 解码单元 6 的处理结果,二进制解码单元 7 解码并输出原始语法元素 (syntax)。

[0017] 控制单元 3 是控制整个解码设备 1 的操作的控制单元。控制单元 3 指示上下文计算单元 2 使用通过适应算术编码 / 解码单元 6 生成的二进制化数据来执行上下文计算。此外,控制单元 3 向适应算术编码 / 解码单元 6 通知通过上下文计算单元 2 获得的上下文索引 (ctxIdx),并指示二进制解码单元 7 对适应算术编码 / 解码单元 6 的处理结果执行处理。

[0018] 在开始处理 1NAL (网络提取层) 单元之前,控制单元 3 对保持在概率状态存储单元 4 的存储器 5 中的概率状态变量进行初始化,随后根据适应算术编码 / 解码单元 6 的处理结果更新存储在存储器 5 中的概率状态变量。此外,在系数数据处理中,控制单元 3 基于当处理系数数据的一项时适应算术编码 / 解码单元 6 的执行结果,使用连续的系数数据来设置待被处理的二进制值的数。

[0019] 图 4 是描述对解码设备 1 中的连续语法元素的处理的时序图。基于上下文的自适应二进制算术解码处理被用于在 ITU-T H.264 中定义的切片数据层以下的多个语法元素。因此,解码设备 1 基于宏块 (macroblock) 顺序处理诸如宏块型 (mb 型)、转换型 (transform)、码块模式 (cbp) 和子宏块型 (子 mb 型) 的语法元素,然后在宏块处理中处理剩余层 (图 4(A) 和图 4(C))。这里,剩余层被划分为 4×4 像素剩余块。通过对图像数据执行离散余弦变换处理和量化处理而获得的系数数据 (coeff abs level minus1 (由图 4 中的 level 所表示)) 被分配给每个剩余块 (图 4(B))。解码设备 1 对该剩余层中的连续剩余块的系数数据 (coeff abs level minus1) 顺序执行编码处理 (图 4(C))。

[0020] 对于诸如宏块型 (mb 型)、转换型、码块模式 (cbp) 和子宏块型 (子 mb 型) 的语法元素来说,在一个宏块中仅存在一个。相反,对于剩余块存在 64 项系数数据 (coeff abs level minus1)。因此,例如,当处理所谓的 4:2:0 图像数据时,由于通过 16×16 个像素形成亮度信号宏块且通过 8×8 个像素形成色度信号宏块,所以在这种情况下,一个宏块存在 $6 \times 64 = 384$ 项的系数数据 (coeff abs level minus1)。

[0021] 因此,当解码设备 1 对剩余块的系数数据 (coeff abs level minus1) 执行编码处理时,解码设备 1 首先将记录在存储器 5 中的概率状态变量进行初始化,随后根据适应算术编码 / 解码单元 6 的处理结果顺序更新存储在存储器 5 中的概率状态变量。注意,在开始 1NAL (网络提取层) 单元时执行该初始化处理。此外,解码设备 1 基于当处理一项系数数据 (coeff abs level minus1) 时适应算术编码 / 解码单元 6 的执行结果,使用连续的系数数据 (coeff abs level minus1) 来设置待被处理的二进制值的数。

[0022] 图 5 是描述对剩余块的系数数据 (coeff abs level minus1) 的连续处理的时序图。注意,图 5 对应于使用访问等待时间较长的 SRAM 构成图 3 中的存储器 5 的情况。而且,每个处理的区段对应于作为处理单位的 1 个周期。在首先初始化存储器 5 的内容之后,解码设备 1 使用上下文计算单元 2 来计算第一系数数据的上下文 (coeff abs level minus1(0)) (图 5(A) 和图 5(B))。注意,图 5 中的实例是通过计算这个上下文利用低位二

进制值获得上下文索引 (ctxIdx0) 并利用随后的第二~第四二进制值连续获得一个上下文索引 (ctxIdx1) 的情况。因此,在这两个上下文索引 (ctxIdx0) 和 (ctxIdx1) 中,第二个上下文索引 (ctxIdx1) 用于处理第二~第四二进制值。注意,在下文中,根据需要使用表示从开头开始的顺序的括号中的数字来表示每项系数数据。

[0023] 在解码设备 1 中,通过基于上下文索引 (ctxIdx0) 和 (ctxIdx1) 的第一个上下文索引 (ctxIdx0) 使用适应算术编码/解码单元 6 计算概率 (arith0) 来选择将被记录在存储器 5 中的状态索引 (stateIdx0) 和 MPS(图 5(C) 和图 5(D))。此外,通过计算后续的概率 (arith1),基于上下文索引 (ctxIdx1) 来选择将被记录在存储器 5 中的状态索引 (stateIdx1) 和 MPS。解码设备 1 重复待被处理的二进制值数的状态索引 (stateIdx) 和 MPS 的选择,使用二进制解码单元 7 处理处理结果 bin,并解码语法元素 (syntax0)(图 5(E) 和图 5(F))。此外,如箭头 A 所示,在基于处理结果更新记录在存储器 5 中的概率状态变量之后,解码设备 1 类似地处理后续的概率数据 (coeff abs level minus1)。

[0024] 关于这种基于上下文的适应型二进制解码设备,在日本未审查专利申请公开第 2005-130099 号、日本未审查专利申请公开第 2005-217871 号等中推荐了用于提高处理速度的方案。

[0025] 顺便提及,在这种类型的编码设备和解码设备中,期望提高处理速度。在为了提高速度而研究图 1 所示解码装置 1 的操作之后,意识到存在在连续语法元素 (syntax) 的处理之间产生空闲时间的问题。

[0026] 即,如图 5 所示,当使用访问等待时间较长的 SRAM 构成存储器 5 时,在计算了上下文之后的 2 个周期后获得概率状态变量的状态索引 (stateIdx) 和 MPS。在图 5 中的实例中,在后续的 4 个周期内,执行基于计算结果对概率状态变量的处理,因此对语法元素 (syntax) 进行解码。注意,基于上下文计算结果对概率状态变量的处理是获得概率状态变量的处理、算术解码处理和二进制化处理。在图 5 中的实例中,使用流水线系统来执行这些处理。此外,在解码语法元素 (syntax) 之后,基于处理结果来更新存储器 5 中的记录,并开始对后续系数数据 (coeff abs level minus1) 的处理。因此,当使用访问等待时间较长的 SRAM 构成存储器 5 时,在传统的解码设备 1 中,从在一个语法元素中完成对概率状态变量的处理时到在随后的语法元素中开始对概率状态变量的处理时产生了 3 个周期的空闲时间。

[0027] 在对所有项系数数据的处理中产生了 3 个周期的空闲时间。由于在对一个宏块中的连续系数数据项的处理中产生了空闲时间,所以当从整体上来看时空闲时间非常多。

[0028] 作为解决该问题的一个方法,可以使用将访问等待时间为 0 的寄存器应用于存储器 5 并保持概率状态变量的方法。但是,在这种方法的情况下,尽管上述 3 个周期的空闲时间可以被减小至 1 个周期,但与 SRAM 相比,存储器 5 的面积被增大。存在整体结构的尺寸增加的问题。

发明内容

[0029] 鉴于上述几点,本发明提供了运动图像处理方法、运动图像处理方法的程序、其上记录有运动图像处理方法的程序的记录介质以及运动图像处理设备,其与过去相比可以避免整体结构尺寸的增加并提高了处理速度。

[0030] 为了解决上述问题,本发明应用于计算上下文并编码或解码运动图像的运动图像

处理方法,包括:上下文计算处理步骤,计算构成运动图像的语法元素的上下文,并顺序地检测可以被语法元素采用的概率状态变量;以及概率状态变量处理步骤,顺序地选择并处理在上下文计算处理步骤中获得的概率状态变量,并编码或解码语法元素。概率状态变量处理步骤包括:选择在概率状态存储单元中保持的概率状态变量的概率状态变量选择步骤。在概率状态存储单元中设置第一存储器和与第一存储器相比访问等待时间短的第二存储器。概率状态变量选择步骤当处理具有低出现频率的语法元素时从第一存储器中顺序选择概率状态变量,而当处理具有高出现频率的语法元素时从第二存储器中顺序选择概率状态变量。

[0031] 在本发明的结构中,尽管整体形状的尺寸增加,但使用访问等待时间短的第二存储器来选择概率状态变量的结构可以提高处理速度。相反,尽管整体形状的尺寸减小,但使用访问等待时间长的第一存储器来选择概率状态变量的结构却难以提高处理速度。因此,根据权利要求1的结构,当进行配置使得当具有低出现频率的语法元素待被处理时从第一存储器中顺序选择概率状态变量,而当具有高出现频率的语法元素待被处理时从第二存储器中顺序选择概率状态变量,在使用第一和第二存储器的情况下,该结构可以利用两者的优势效果,因此与过去相比避免了整体结构尺寸的增加并提高了处理速度。

[0032] 此外,本发明应用于计算上下文并编码或解码运动图像的运动图像处理方法的程序,包括:上下文计算处理步骤,计算构成运动图像的语法元素的上下文,并顺序地检测可以被语法元素采用的概率状态变量;以及概率状态变量处理步骤,顺序地选择并处理在上下文计算处理步骤中获得的概率状态变量,并编码或解码语法元素。概率状态变量处理步骤包括选择在概率状态存储单元中保持的概率状态变量的概率状态变量选择步骤。在概率状态存储单元中设置第一存储器和与第一存储器相比访问等待时间短的第二存储器。概率状态变量选择步骤当处理具有低出现频率的语法元素时从第一存储器中顺序选择概率状态变量,而当处理具有高出现频率的语法元素时从第二存储器中顺序选择概率状态变量。

[0033] 在本发明的结构中,尽管整体形状的尺寸增加,但使用访问等待时间短的第二存储器来选择概率状态变量的结构可以提高处理速度。相反,尽管整体形状的尺寸减小,但使用访问等待时间长的第一存储器来选择概率状态变量的结构却难以提高处理速度。因此,根据本发明的结构,当进行配置使得当具有低出现频率的语法元素待被处理时从第一存储器中顺序选择概率状态变量,而当具有高出现频率的语法元素待被处理时从第二存储器中顺序选择概率状态变量,在使用第一和第二存储器的情况下,该结构可以利用两者的优势效果,因此与过去相比避免了整体结构尺寸的增加并提高了处理速度。

[0034] 此外,本发明应用于其上记录有计算上下文并编码或解码运动图像的运动图像处理方法的程序的记录介质。运动图像处理方法的程序包括:上下文计算处理步骤,计算构成运动图像的语法元素的上下文,并顺序地检测可以被语法元素采用的概率状态变量;以及概率状态变量处理步骤,顺序地选择并处理在上下文计算处理步骤中获得的概率状态变量,并编码或解码所述语法元素。概率状态变量处理步骤包括选择在概率状态存储单元中保持的概率状态变量的概率状态变量选择步骤。在概率状态存储单元中设置第一存储器和与第一存储器相比访问等待时间短的第二存储器。概率状态变量选择步骤当处理具有低出现频率的语法元素时从第一存储器中顺序选择概率状态变量,而当处理具有高出现频率的

语法元素时从第二存储器中顺序选择概率状态变量。

[0035] 在本发明的结构中, 尽管整体形状的尺寸增加, 但使用访问等待时间短的第二存储器来选择概率状态变量的结构可以提高处理速度。相反, 尽管整体形状的尺寸减小, 但使用访问等待时间长的第一存储器来选择概率状态变量的结构却难以提高处理速度。因此, 根据本发明的结构, 当进行配置使得当具有低出现频率的语法元素待被处理时从第一存储器中顺序选择概率状态变量, 而当具有高出现频率的语法元素待被处理时从第二存储器中顺序选择概率状态变量, 在使用第一和第二存储器的情况下, 该结构可以利用两者的优势效果, 因此与过去相比避免了整体结构尺寸的增加并提高了处理速度。

[0036] 此外, 本发明应用于计算上下文并编码或解码运动图像的运动图像处理装置, 包括: 上下文计算单元, 用于计算构成运动图像的语法元素的上下文, 并顺序地检测可以被语法元素采用的概率状态变量; 概率状态存储单元, 其中, 设置了第一存储器和与所述第一存储器相比访问等待时间短的第二存储器, 概率状态变量被保持在第一存储器和第二存储器中; 以及概率状态变量处理单元, 用于基于上下文计算单元的检测结果顺序地选择并处理来自概率状态存储单元的概率状态变量, 并编码或解码语法元素。概率状态变量处理单元当处理具有低出现频率的语法元素时从第一存储器中顺序选择概率状态变量, 而当处理具有高出现频率的语法元素时从第二存储器中顺序选择概率状态变量。

[0037] 在本发明的结构中, 尽管整体形状的尺寸增加, 但使用访问等待时间短的第二存储器来选择概率状态变量的结构可以提高处理速度。相反, 尽管整体形状的尺寸减小, 但使用访问等待时间长的第一存储器来选择概率状态变量的结构却难以提高处理速度。因此, 根据本发明的结构, 当进行配置使得当具有低出现频率的语法元素待被处理时从第一存储器中顺序选择概率状态变量, 而当具有高出现频率的语法元素待被处理时从第二存储器中顺序选择概率状态变量, 在使用第一和第二存储器的情况下, 该结构可以利用两者的优势效果, 因此与过去相比避免了整体结构尺寸的增加并提高了处理速度。

[0038] 根据本发明, 与过去相比可以避免整体结构尺寸的增加并提高了处理速度。

附图说明

[0039] 图 1 是示出基于上下文的自适应二进制算术编码处理的处理程序的流程图。

[0040] 图 2 是示出基于上下文的自适应二进制算术解码处理的处理程序的流程图。

[0041] 图 3 是示出基于传统的基于上下文的自适应二进制算术处理的解码设备的框图。

[0042] 图 4 是用于描述在图 3 的解码设备中对连续语法元素的处理的时序图。

[0043] 图 5 是用于描述在图 3 的解码设备中对剩余块的系数数据的连续处理的时序图。

[0044] 图 6 是示出本发明的实施例 1 的解码设备的框图。

[0045] 图 7 是用于描述图 6 中的解码设备的操作的时序图。

[0046] 图 8 是示出图 6 的解码设备中的上下文索引转换的示意图。

[0047] 图 9 是示出图 8 中的转换的一个实例的示意图。

[0048] 图 10 是示出图 6 的解码设备中的寄存器结构的示意图。

[0049] 图 11 是用于描述本发明实施例 2 的解码设备的示意图。

[0050] 图 12 是用于描述本发明实施例 3 的解码设备的框图。

具体实施方式

[0051] 下文,根据需要参照附图详细描述本发明的实施例。

[0052] (1) 实施例 1 的结构

[0053] 与图 3 相比,图 6 是示出本发明实施例 1 的解码设备的框图。在解码设备 10 中,与图 3 所述解码设备 1 相同的结构被给出对应的参考标号,并忽略了其重复描述。解码设备 10 根据 ITU-T H. 264 的规则顺序地对来自比特流的语法元素 (syntax) 进行解码,并通过对语法元素 (syntax) 执行逆量化处理和逆正交变换处理来对运动图像的图像数据进行解码。注意,可以使用硬件构成图 6 所示的每个单元,或者可以使用算术处理装置的功能块来构成每个单元。注意,当使用算术处理装置的功能块构成每个元件时,可以通过预先安装来提供用于算术处理装置的程序。可选地,可以通过将其记录在诸如光盘、磁盘、存储卡等的记录介质上来提供程序。此外,可以通过诸如互联网等的网络下载来提供程序。

[0054] 在解码设备 10 中,在概率状态存储单元 11 中设置访问等待时间长的第一存储器和与第一存储器相比访问等待时间较短的第二存储器。这里,在该实施例中,由例如与第二存储器相比功耗低且形状小的 SRAM 的存储器 12 来构成第一存储器。相反,通过寄存器 13 来构成第二存储器。注意,例如,通过触发器构成寄存器 13。

[0055] 概率状态存储单元 11 在存储器 12 中存储用于所有语法元素的概率状态变量。此外,在存储器 12 中所存储的概率状态变量中,对应于具有高出现频率的语法元素并被频繁使用的概率状态变量被加载并保持在寄存器 13 中。当解码具有高出现频率的语法元素时,解码设备 10 使用保持在寄存器 13 中的概率状态变量来处理具有高出现频率的语法元素。相反,解码设备 10 使用记录在存储器 12 中的数据来处理除具有高出现频率的语法元素之外的语法元素。具体地,在该实施例中,对频繁使用的概率状态变量分配用于系数数据 (coeff abs level minus1) 的概率状态变量。

[0056] 在概率状态存储单元 11 中,在控制单元 19 的控制下,选择单元 14 切换操作,并且在存储器 12 和寄存器 13 之间切换适应算术编码 / 解码单元 16 的访问目标。此外,在控制单元 19 的控制下,基于解码结果来更新记录在存储器 12 中的概率状态变量。此外,在控制单元 19 的控制下,当将保持在存储器 12 中的概率状态变量加载到寄存器 13 并基于解码结果更新记录在存储器 12 中的概率状态变量时,更新存储在寄存器 13 中的对应概率状态变量,以对应于存储器 12 中的记录。

[0057] 在控制单元 19 的控制下,上下文计算单元 18 从待被处理的比特流中获得上下文索引 (ctxIdx)。在系数数据项 (coeff abs levelminus1(1-N)) 的连续解码处理中,上下文计算单元 18 在正在处理紧前的语法元素中计算的上下文索引的概率状态变量的周期内对后续的语法元素执行上下文计算。

[0058] 适应计算编码 / 解码单元 16 将比特流二进制化,以对应于上下文计算单元 18 中的上下文计算,并向控制单元 19 通知经过二进制化的比特流。此外,基于由上下文计算单元 18 获得的上下文索引 (ctxIdx),适应算术编码 / 解码单元 16 访问存储器 12 和寄存器 13,顺序地选择 MPS (最或然符号) 和状态索引 (stateIdx),并执行算术解码处理。

[0059] 控制单元 19 是控制整个解码设备 10 操作的控制单元。除了对系数数据 (coeff abs level minus1) 之外,控制单元 19 如上面参照图 4 所描述的解码设备 1 的控制单元 3 一样来控制整体操作。即,在这种情况下,控制单元 19 指示上下文计算单元 18 对通过适应

算术编码 / 解码单元 16 生成的二进制数据执行上下文计算。而且,控制单元 19 指示适应算术编码 / 解码单元 16 和二进制解码单元 7 对通过上下文计算单元 18 获得的上下文索引 (ctxIdx) 执行处理。此外,基于适应算术编码 / 解码单元 16 的处理结果,控制单元 19 更新保持在概率状态存储单元 11 中的概率状态变量。

[0060] 相反,当对系数数据 (coeff abs level minus1) 执行解码处理时,控制单元 19 将在存储器 12 中所保持的概率状态变量加载到寄存器 13,并控制每个单元的操作,以使用加载在寄存器 13 中的概率状态变量来执行状态索引 (stateIdx) 和 MPS 的选择。此外,控制单元 19 控制上下文计算单元 18、适应算术编码 / 解码单元 16 等的操作,从而在正在处理一个语法元素的概率状态变量的周期内计算后续语法元素中的索引,并将对应的概率状态变量存储在寄存器 13 中。

[0061] 相对于图 5,图 7 是示出在控制单元 19 的控制下对系数数据 (coeff abs level Minus1) 的处理的时序图。注意,如在图 4 中的情况一样,图 7 对应于通过第一系数数据 (coeff abs level minus1(0)) 的低位二进制值和第二~第四二进制值分别获得上下文索引 (ctxIdx0) 和上下文索引 (ctxIdx1) 并通过后续系数数据 (coeff abs level minus1(1)) 的两个连续二进制值分别获得上下文索引 (ctxIdx2) 和 (ctxIdx3) 的情况。

[0062] 当控制单元 19 开始对系数数据 (coeff abs level minus1(0)) 的处理时,控制单元 19 控制上下文计算单元 18 和适应算术编码 / 解码单元 16 在连续周期内顺序获得上下文索引 (ctxIdx0) 和 (ctxIdx1) (图 7(A) 和图 7(B))。控制单元 19 还控制概率状态存储单元 11 在后续的两个周期内将对应于通过上下文计算单元 18 获得的上下文索引 (ctxIdx0) 和 (ctxIdx1) 的概率状态变量从存储器 12 加载至寄存器 13 (图 7(C))。控制单元 19 控制适应算术编码 / 解码单元 16 使用加载到寄存器 13 中的概率状态变量来执行选择状态索引 (stateIdx) 和 MPS 的概率状态选择处理、算术解码处理以及二进制化处理 (图 7(D) ~ 图 7(F))。

[0063] 这里,由于寄存器 13 的访问等待时间为 0,所以概率状态选择处理、算术解码处理和二进制化处理在将在处理中使用的概率状态变量存储在寄存器 13 中的周期内开始对第一二进制值的处理。

[0064] 控制单元 19 控制上下文计算单元 18,以在通过适应算术编码 / 解码单元 16 执行概率状态选择处理、算术解码处理和二进制化处理的周期内计算后续系数数据 (coeff abs level minus1(1)) 的上下文。在该实施例中,对后续系数数据 (coeff abs level minus1(1)) 的上下文计算处理从完成对紧前的系数数据 (coeff abs level minus1(0)) 的上下文计算后紧接的周期开始。

[0065] 此外,当控制单元 19 完成对后续系数数据 (coeff abs level minus1(1)) 的上下文计算时,控制单元 19 在后续周期中将在这些上下文计算中获得的上下文索引 (ctxIdx2) 和 (ctxIdx3) 的概率状态变量从存储器 12 加载到寄存器 13,在完成对紧前的系数数据 (coeff abs level minus1(0)) 的处理之前,在寄存器 13 中预先保持这些概率状态变量,使得可以处理后续的系数数据 (coeff abs level minus1(1))。

[0066] 基于适应算术编码 / 解码单元 16 的处理结果,控制单元 19 顺序更新保持在寄存器 13 和存储器 12 中的概率状态变量,并当完成对紧前的系数数据 (coeff abs level minus1(0)) 的处理时,指示适应算术编码 / 解码单元 16 处理后续的系数数据 (coeff abs

level minus1(1))。

[0067] 这里,在解码设备 10 中,由于预先在寄存器 13 中存储了处理后续系数数据 (coeff abs level minus1(1)) 所需的上下文索引 (ctxIdx2) 和 (ctxIdx3) 的概率状态变量,所以适应算术编码 / 解码单元 16 可以在完成紧前的系数数据 (coeff abs level minus1(0)) 的处理时开始在后续周期中处理后续的系数数据 (coeff abs level minus1(1)),而不会引发空闲时间。因此,在解码装置 10 中,与过去相比可以以更高的速度解码图像数据。

[0068] 注意,当控制单元 19 指示适应算术编码 / 解码单元 16 开始处理后续的系数数据 (coeff abs level minus1(1)) 时,控制单元 19 同时指示上下文计算单元 18 对再一个系数数据 (coeffabs level minus1(2)) 执行上下文计算。

[0069] 顺便提及,当预先执行对后续系数数据 (coeffabs level minus1(N)) 的上下文计算并通过有效利用正在处理存储在寄存器 13 中的概率状态变量的周期来将概率状态变量保持在寄存器 13 中时,会发生在适应算术编码 / 解码单元 16 开始处理后续系数数据 (coeffabs level minus1(N)) 的处理时没有完成将概率状态变量存储在寄存器 13 中的情况。即,发生概率状态变量在寄存器 13 中的存储对于每个语法元素的解码处理来说太迟的情况。

[0070] 更具体地,图 8 是预测在顺序处理存在于剩余块中的系数数据 (coefflevel abs minus1) 项的情况下所需的上下文索引和概率状态变量的时序图。注意,在图 8 的实例中,假设系数数据 (coeff levelabs minus1) 的每项的每个二进制值的值为 0 或 1。

[0071] 这里,在剩余块的第一系数数据 (coeff level abs minus1(0)) 中,唯一确定上下文索引 (ctxIdx),且 ctxIdx = 1。在解码设备 10 中,当系数数据 (coeff level abs minus1(0)) 的低位二进制值 (level0bin0) 具有 0 值时,解码设备 10 前进至处理后续的系数数据 (coefflevel abs minus1(1)),并处理后续系数数据 (coefflevel abs minus1(1)) 的低位二进制值 (level1bin0)。可选地,与上述情况相反,当系数数据 (coeff level abs minus1(0)) 的低位二进制值 (level0bin0) 具有 1 值时,解码设备 10 处理后续的高位二进制值 (level0bin1)。这里,在图 8 的实例中,假设这些二进制值 (level1bin0) 和 (level0bin1) 的上下文索引 (ctxIdx) 分别为 ctxIdx = 2 和 ctxIdx = 5。

[0072] 此外,在图 8 的实例中,当选择第一系数数据 (coeff level abs minus1(0)) 后的二进制值 (level0bin1) 时,随后将处理后续系数数据 (coeff level abs minus1(1)) 的低位二进制值 (level1bin0)。此外,当系数数据 (coeff level abs minus1(0)) 的低位二进制值 (level0bin0) 具有 0 值且后续系数数据 (coeff level abs minus1(1)) 的低位二进制值 (level1bin0) 具有 0 值和 1 值时,随后即将处理后续的系数数据 (coeff level abs minus1(2)) 的低位二进制值 (level2bin0) 和相同系数数据 (coeff level abs minus1(2)) 的后续二进制值 (level2bin1)。注意,图 8 中最右边的标志示出了图 8 中的最终转换目的地中直到对应转换目的地的系数数据项。此外,尽管与图 8 不对应,但图 9 是示出在系数数据的连续项中存在具有值 1 的各个二进制值的预定数的序列的情况下的转换示意图。

[0073] 因此,当即将处理图 8 中的实例时,在对第一系数数据 (coefflevel abs minus1(0)) 的上下文计算中需要获得上下文索引 ctxIdx = 1 和 ctxIdx = 5,并且在对后续系数数据 (coeff level abs minus1(1)) 预先的上下文计算中计算上下文索引 ctxIdx = 0、ctxIdx = 2、ctxIdx = 5 和 ctxIdx = 6。

[0074] 这里,图 8 中水平方向上的每个转换对应于图 7 中的 1 个周期。因此,通过图 6 中的结构,当预先执行上下文计算并在寄存器 13 中存储概率状态变量时,对于第一系数数据 (coeff level abs minus1(0)),在寄存器 13 中存储上下文索引 ctxIdx = 1 和 ctxIdx = 5 的概率状态变量后,处理开始。在这种情况下,可在寄存器 13 中存储概率状态变量,而根本不会迟于概率状态变量的处理的开始。

[0075] 此外,在处理后续系数数据 (coeff level abs minus1(1)) 的过程中,需要上下文索引 ctxIdx = 0、ctxIdx = 2、ctxIdx = 5 和 ctxIdx = 6 的概率状态变量。这里,优先于在低位二进制值的处理被切换至后续高位二进制值的处理的情况下 bin = 1 侧的上下文索引 ctxIdx = 0 和 ctxIdx = 6,在寄存器 13 中存储在低位二进制值的处理被切换至下一系数数据的处理的情况下 bin = 0 侧的上下文索引 ctxIdx = 2 和 ctxIdx = 5。此外,当在寄存器 13 中存储概率状态变量时,从对应于低位侧的二进制值的上下文索引开始,在完成第一系数数据 (coeff level abs minus1(0)) 的低位二进制值 (level0bin0) 的时刻,可以在寄存器 13 中保持在 bin = 0 的情况下处理后续系数数据 (coeff level abs minus1(1)) 的低位二进制值 (level1bin0) 所需的上下文索引 ctxIdx = 2 的概率状态变量。此外,在完成低位二进制值 (level1bin0) 的处理的时刻,可以在寄存器 13 中保持在 bin = 1 的情况下处理后续低位二进制值 (level1bin1) 所需的上下文索引 ctxIdx = 5 的概率状态变量。因此,在这种情况下,还可以在寄存器 13 中存储概率状态变量,而根本不会太迟于后续系数数据 (coeff level abs minus1(1)) 的处理。

[0076] 然而,当这种情况是第一系数数据 (coeff level abs minus1(0)) 的低位二进制值 (level0bin0) 为 bin = 1 的情况时,在完成后续高位二进制值 (level0bin1) 的处理时,没有在寄存器 13 中预备处理后续系数数据 (coeff level abs minus1(1)) 的低位二进制值 (level1bin0) 所需的上下文索引 ctxIdx = 0 的概率状态变量。概率状态变量在寄存器 13 中的存储对于语法元素的解码处理来说变得太迟。

[0077] 因此,如图 10(A) 所示,在解码设备 10 中,通过预定数目的概率状态变量临时寄存器 13A 和预定数目的概率状态变量寄存器 13B 来构成寄存器 13。这里,如上面参照图 7 所描述的,概率状态变量临时寄存器 13A 是与上下文计算、算术解码处理等并行地从存储器 12 加载并存储概率状态变量的寄存器。概率状态变量寄存器 13B 是在开始剩余层的处理之前预先存储其在寄存器 13 中的存储会太迟于语法元素的解码处理的概率状态变量的寄存器。

[0078] 这里,用于系数数据的语法元素的上下文索引 (ctxIdx) 根据类型,即 AC 分量、DC 分量、亮度信号 (Y) 分量和色度信号 (Cr, Cb) 分量而划分为 6 个类别。在每个类别中,具有 9 ~ 10 种类型的上下文索引 (ctxIdx)。此外,由于在每个剩余块中唯一地选择类别,所以在一个类别中使用 9 种类型或 10 种类型的上下文索引 (ctxIdx)。因此,为各个类别提供概率状态变量临时寄存器 13A。在每个类别中,最多提供 10 个概率状态变量临时寄存器 13A,从而能够保持对应于 9 种类型或 10 种类型的上下文索引的概率状态变量。

[0079] 注意,在 1 个周期中可存储在寄存器 13 中的概率状态变量的数目根据从存储器 12 传送至寄存器 13 的数据传送能力而改变,并根据这个数目的改变,太迟而不能被存储的概率状态变量数发生改变。此外,太迟而不能被存储的概率状态变量数根据通过格式确定的系数数据中的上下文索引 (ctxIdx) 转换而改变。因此,可以根据从存储器 12 传送至寄存器

13 的数据传送能力并根据通过格式确定的系数数据中的上下文索引 (ctxIdx) 转换来可变地设置概率状态变量临时寄存器的数目和概率状态变量寄存器的数目。注意, 可以省略概率状态变量寄存器 13B 的结构, 这是由于在一个语法元素中, 考虑到从存储器 12 向寄存器 13 传送的数据传送能力对于可通过上下文计算获得的上下文索引的最大数来说是足够的, 即, 更具体地, 例如, 考虑到通过上下文计算获得的所有上下文索引的概率状态变量可以在 1 个周期内被传送至寄存器 13。

[0080] 与寄存器 13 的结构相关, 如图 10 (B2) 所示, 在开始处理剩余块的初始化处理中, 控制单元 19 预先在概率状态变量寄存器 13B 中载入可能太迟而不能存储在寄存器 13 中的概率状态变量。注意, 可通过系数数据 (coeff level abs minus1) 中的上下文索引转换的规则来获得太迟而不能被存储在寄存器 13 中的概率状态变量。

[0081] 此外, 当完成初步加载时, 每个剩余块的处理一个接一个地开始。虽然通过上下文计算获得的概率状态变量正在存储在概率状态变量临时寄存器 13A 中, 但每个语法元素被解码。注意, 在这种情况下, 控制概率状态存储单元 11 的操作, 以省略已经存储在概率状态变量临时寄存器 13A 中的上下文索引的概率状态变量从存储器 12 向概率状态变量临时寄存器 13A 的存储。注意, 可以省略已经存储在概率状态变量寄存器 13B 中的上下文索引的概率状态变量在概率状态变量临时寄存器 13A 中的存储。

[0082] 注意, 图 10 (B1) 是示出当开始每个剩余块的处理时所有的对应概率状态变量从存储器 12 加载至寄存器 13 的情况的时序图。如图 10 (B1) 所示, 通过预先将所有对应的概率状态变量加载在寄存器 13 中, 可以避免概率状态变量在寄存器 13 中的加载对于每个语法元素的解码处理来说太迟的情况。

[0083] 然而, 在这种情况下, 需要在开始处理每个剩余块时将所有随应的概率状态变量从存储器 12 加载至寄存器 13。相反, 在解码设备 10 中, 在处理语法元素的同时, 所需概率状态变量被载入寄存器 13。仅需要将对于加载处理来说太迟的概率状态变量预先存储在寄存器 13 中。这使得与图 10 (B1) 所示的情况相比, 处理时间显著减少, 当从整体上看时, 可以有效地处理图像数据。

[0084] (2) 实施例 1 的操作

[0085] 在上述结构中, 通过适应算术编码 / 解码单元 16 顺序地检测顺序输入的比特流 (图 6 和图 7) 中的语法元素的类型。此外, 在基于适应算术编码 / 解码单元 16 的检测结果的单元 19 的控制下, 通过上下文计算单元 18 顺序计算每个语法元素的上下文, 并获得二进制化的语法元素的各个比特可以采用的上下文索引 (ctxIdx)。在比特流中, 基于上下文索引 (ctxIdx) 的概率状态变量通过适应算术编码 / 解码单元 16 来顺序选择状态索引 (stateIdx) 和 MPS, 并通过处理选择结果来解码原始语法元素。

[0086] 在解码设备 10 中, 根据由适应算术编码 / 解码单元 16 检测的语法元素的类型, 当具有低出现频率的语法元素被解码时, 适应算术编码 / 解码单元 16 访问设置在概率状态存储单元 11 中并作为访问等待时间较长的第一存储器的存储器 12。检测通过上下文计算单元 18 获得的上下文索引 (ctxIdx) 的概率状态变量。此外, 使用所检测的概率状态变量来执行对原始语法元素的解码处理。由于通过 SRAM 构成存储器 12, 所以对于解码这种具有低出现频率的语法元素的解码设备 10 的结构来说, 整体形状可以被小型化。

[0087] 但是, 当 SRAM 存储器 12 用于处理所有语法元素时, 如上面关于具有图 3 中传统结

构的解码设备 1 所描述的,难以以高速执行解码。

[0088] 因此,在解码设备 10 中,当具有高出现频率的语法元素被解码时,将通过上下文计算单元 18 获得的上下文索引 (ctxIdx) 的概率状态变量从存储器 12 加载至作为与第一存储器相比其访问等待时间较长的第二存储器的寄存器 13 并保持在寄存器 13 中。使用存储在寄存器 13 中的概率状态变量,对具有高出现频率的语法元素执行解码处理。因此,尽管其形状变大,但解码设备 10 可以以高速对具有高出现频率的语法元素执行处理。

[0089] 这里,在这种类型的解码设备中,诸如宏块型和子宏块型的语法元素作为具有低出现频率的语法元素存在。当处理一个宏块时,这些语法元素的每一个仅出现一次。

[0090] 相反,DC 系数数据和 AC 系数数据作为具有高出现频率的语法元素存在。在它们中,在 4:2:0 图像数据的情况下,当处理一个宏块时,AC 系数数据连续出现 15 次,意味着 $15 \times 4 \times 6 = 360$ 次。此外,当处理一个宏块时,DC 系数出现 6 次。因此,如在解码设备 10 中一样,当从整体上看时,当寄存器 13 仅被用于具有高出现频率的语法元素时,通过在使用寄存器 13 和存储器 13 的情况下利用两者的优势效果,与过去相比可以显著提高处理速度,而不会显著增大结构尺寸。

[0091] 另外,在解码设备 10 中,不仅具有高出现频率的语法元素的概率状态变量被保持在寄存器 13 中,而且通过上下文计算单元 18 获得的上下文索引 (ctxIdx) 的概率状态变量被选择性地从存储器 12 加载至寄存器 13,并使用这些加载的概率状态变量来执行解码处理。因此,寄存器 13 仅需要被配置为能够仅存储处理系数数据所需的所有上下文索引 (ctxIdx) 的一部分概率状态变量。具体地,处理系数数据所需的所有概率状态变量的数目为 59 种类型。因此,因为寄存器 13 仅需要被配置为能够仅存储处理系数数据所需的所 有上下文索引 (ctxIdx) 的一部分概率状态变量,所以与过去相比可以避免整体结构尺寸的增加并提高了处理速度。

[0092] 此外,在解码设备 10 中,概率状态变量被选择性地从存储器 12 加载至寄存器 13 并保持在寄存器 13 中。在正在处理保持在寄存器 13 中的概率状态变量的周期内,更具体地,在适应算术编码 / 解码单元 16 对系数数据 (coeff abs level Minus1(0)) 的一项执行概率状态选择处理、算术解码处理和二进制化处理的周期内,对后续的系数数据 (coeff abs level Minus1(1)) 执行上下文计算,并在寄存器 13 中存储对应的概率状态变量。此外,根据解码处理的结果更新存储在寄存器 13 中的概率状态变量。因此,在解码设备 10 中,在开始对后续的系数数据 (coeff abs level minus1(1-N)) 的处理时,概率状态变量已经被保持在寄存器 13 中。可以在完成对紧前的系数数据的处理后立刻开始对后续的系数数据 (coeff abs level minus1(1-N)) 的处理。这更加提高了处理速度。具体地,当处理高分辨率图像数据时,与过去相比可以显著提高处理速度。

[0093] 即,例如,在高于或等于在 ITU-T H.264 中标准化的等级 4 的高分辨率 (1920×1088) 的情况下,根据该实施例,与图 5 中的上述传统结构相比,在一个宏块中可以减少 693 个周期的空闲时间。根据试验结果,即使当与传统结构相比操作速度减小了约 170 [MHz] 时,也可以有效地执行解码。

[0094] 然而,当预先对后续的系数数据 (coeff abs level minus1(N)) 执行上下文计算并通过有效地利用处理存储在寄存器 13 中的概率状态变量的周期将概率状态变量保持在寄存器 13 中时,根据系数数据中的上下文索引 (ctxIdx) 转换,会发生概率状态变量在寄存

器 13 中的存储对于每个语法元素的解码处理来说太迟的情况（图 8 和图 9）。

[0095] 因此，在该实施例中，通过概率状态变量临时寄存器 13A 和概率状态变量寄存器 13B 构成寄存器 13。对于可能太迟而不能存储在寄存器 13 中的概率状态变量来说，在开始处理剩余块时，概率状态变量被预先从存储器 12 加载至概率状态变量寄存器 13B 并保持在概率状态变量寄存器 13B 中。此外，在处理每个语法元素的同时，根据需要从存储器 12 加载其它概率状态变量并保持在概率状态变量临时寄存器 13A 中。

[0096] 结果，在解码设备 10 中，通过有效地利用处理存储在寄存器 13 中的概率状态变量的周期，将处理后续系数数据 (coeff abs levelminus1(N)) 所需的概率状态变量预先保持在寄存器 13 中，从而提高了处理速度。可以有效地避免由于概率状态变量在寄存器 13 中的存储太迟并且解码设备 10 等待概率状态变量在寄存器 13 中的存储完成的事实所引起的空闲时间的出现。因此，因为有效地避免了空闲时间的出现，所以可以提高处理速度。

[0097] (3) 实施例 1 的优势效果

[0098] 根据上述结构，使用保持在访问等待时间短的第二存储器中的概率状态变量来解码并处理具有高出现频率的语法元素，并使用保持在访问等待时间长的第一存储器中的概率状态变量来解码其它语法元素，因此与过去相比避免了整体结构尺寸的增加并提高了处理速度。

[0099] 此外，通过将系数数据应用于使用了保持在第二存储器中的概率状态变量的具有高出现频率的语法元素，与过去相比可以进一步避免整体结构尺寸的增加，并且可以提高处理速度。

[0100] 此外，由于访问等待时间短的第二存储器是寄存器，所以可以以 0 的访问等待时间执行连续处理，可以进一步提高处理速度。

[0101] 此外，通过根据上下文计算结果将保持在第一存储器中的概率状态变量选择性地加载到第二存储器并使用所加载的概率状态变量，第二存储器的容量可以被抑制到所要求的最小尺寸，并且可以避免整体形状尺寸的增加。

[0102] 此外，通过在处理保持在第二存储器中的概率状态变量的周期内对后续的语法元素执行上下文计算并在寄存器 13 中存储概率状态变量，可以更进一步地提高处理速度。

[0103] 此外，通过在处理保持在第二存储器中的概率状态变量的周期内对后续的语法元素执行上下文计算并在寄存器 13 中存储概率状态变量，提高了处理速度。在处理开始时，太迟而不能存储在第二存储器中的概率状态变量被加载并保持在第二存储器中。这有效地避免了处理等待概率状态变量在寄存器 13 中的存储完成的空闲时间的出现，并且可提高处理速度。

[0104] (4) 实施例 2

[0105] 与图 8 相对，图 11 是用于描述本发明实施例 2 的解码设备的时序图。注意，在该实施例中，除关于图 11 所示时序图的结构之外，以与实施例 1 相同的方式来构成解码设备。因此，在下面的描述中，图 6 中的结构被用于给出描述。该解码设备 20 同时并行地处理每个语法元素的两个连续的二进制值。

[0106] 因此，在解码设备 20 中，概率状态存储单元 11、上下文计算单元 18、适应算术编码 / 解码单元 16、二进制化单元 7 和控制单元 19 在 1 个周期内执行通过图 8 所描述的两个连续周期内的处理，从而能够同时并行地处理连续的两个二进制值。因此，概率状态存储单元

11 在寄存器 13 中存储 1 个周期内的两个上下文索引的概率状态变量。

[0107] 此外,适应算术编码/解码单元 16 根据同时并行处理的两个二进制值的处理结果通过切换图 11 中的转换目的地来切换寄存器 13 的访问目的地。此外,当低位侧的二进制值被处理且高位侧的后续二进制值变得不需要时,即,更具体地,当低位侧的二进制值为 $\text{bin} = 0$ 时,终止对高位侧的后续二进制值的处理。因此,在图 11 的实例中,当同时并行地处理第一二进制值 level0bin0 和后续的二进制值 level0bin1 时,在第一二进制值 level0bin0 为 $\text{bin} = 0$ 的情况下,终止对通过后续的二进制值 level0bin1 所选择的概率状态变量的处理。

[0108] 另外,当同时并行地处理后续系数数据的第一二进制值 level1bin0 和随后的二进制值 level1bin1 时,在第一二进制值 level1bin0 也为 $\text{bin} = 0$ 的情况下,终止对通过后续的二进制值 level1bin1 所选择的概率状态变量的处理。

[0109] 这里,在图 11 的实例中执行多个处理的情况下,当二进制值 level0bin0 为 $\text{bin} = 0$ 时,在第二周期的处理中,需要通过适应算术编码/解码单元 16 来处理上下文索引 $\text{ctxIdx} = 2$ 和 $\text{ctxIdx} = 5$ 的概率状态变量。而且,当二进制值 level0bin0 和 level0bin1 为 $\text{bin} = 1$ 时,在第二周期的处理中,需要通过适应算术编码/解码单元 16 来处理上下文索引 $\text{ctxIdx} = 0$ 和 $\text{ctxIdx} = 6$ 的概率状态变量。

[0110] 因此,在第二周期开始时,需要在寄存器 13 中预备上下文索引 $\text{ctxIdx} = 0$ 、 $\text{ctxIdx} = 2$ 、 $\text{ctxIdx} = 5$ 和 $\text{ctxIdx} = 6$ 的概率状态变量。因此,如通过图 11 中的矩形所围绕并表示的一样,例如,上下文索引 $\text{ctxIdx} = 0$ 的概率状态变量在寄存器 13 中的存储对于处理来说太迟。

[0111] 因此,即使在图 11 中的实例中,如在实施例 1 中所描述的一样,在寄存器 13 中设置概率状态变量临时寄存器 13A 和概率状态变量寄存器 13B。在概率状态变量寄存器 13B 中预先存储对于处理来说太迟的概率状态变量。

[0112] 根据该实施例,即使当同时并行地处理每个语法元素的两个连续的二进制值时,也可以实现与实施例 1 相同的优势效果。

[0113] (5) 实施例 3

[0114] 图 12 是示出本发明实施例 3 的编码设备的框图。该编码设备 30 可以根据 ITU-T H. 264 的格式对运动图像的图像数据顺序执行正交变换处理、量化处理等,以生成语法元素,处理语法元素,并输出比特流。在该编码处理中,编码设备 30 使用上面在实施例 1 和 2 中描述的概率状态存储单元 11 和上下文计算单元 18,根据基于上下文的自适应二进制算术编码处理对语法元素顺序执行编码处理,并输出比特流。

[0115] 即,相对于图 6 中的上述二进制解码单元 7,二进制化单元 33 将待被编码和处理的语法元素顺序进行二进制化。如图 6 中的上述适应算术编码/解码单元 16 中的一样,适应算术编码单元 32 顺序地处理存储在概率状态存储单元 11 的寄存器 13 和存储器 12 中的概率状态变量,顺序地编码和处理由二进制化单元 33 所生成的二进制化数据,并输出比特流。

[0116] 当本发明应用该实施例中的编码设备时,可以实现与实施例 1 和 2 相同的优势效果。

[0117] (6) 其它实施例

[0118] 在上述实施例中,已经描述了以 ITU-T H. 264 的格式执行基于上下文的自适应二进制算术编码和基于上下文的自适应二进制算术解码处理的情况。但是,本发明不限于此。本发明可以广泛地应用于以除这种格式之外的各种格式来执行基于上下文的自适应二进制算术编码和基于上下文的自适应二进制算术解码处理的情况。

[0119] 工业应用

[0120] 本发明涉及运动图像处理方法、运动图像处理方法的程序、其上记录有运动图像处理方法的程序的记录介质以及运动图像处理设备,例如,可以应用于基于 ITU-T H. 264 的运动图像的编码设备和解码设备。

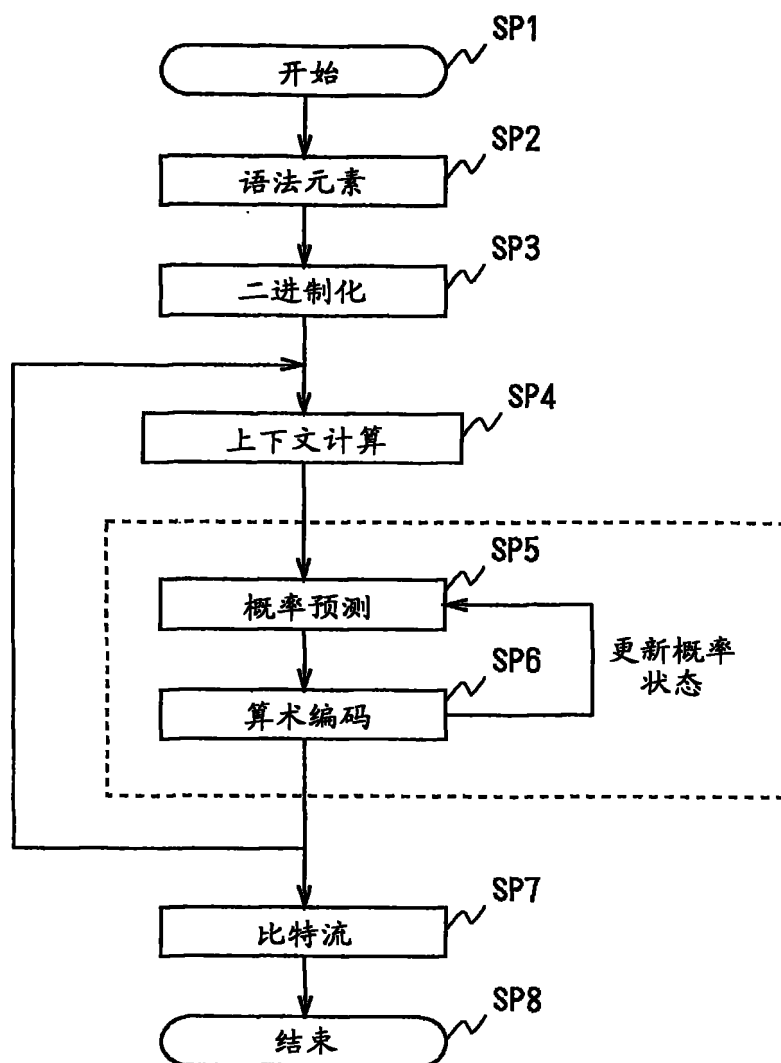


图 1

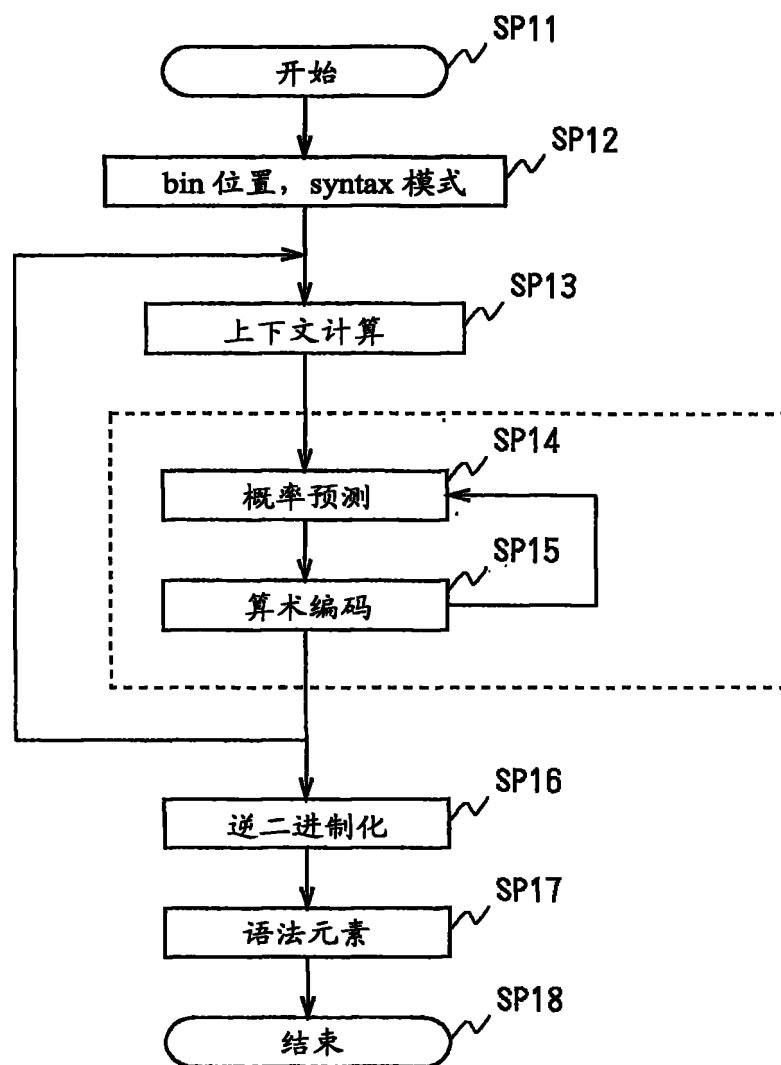


图 2

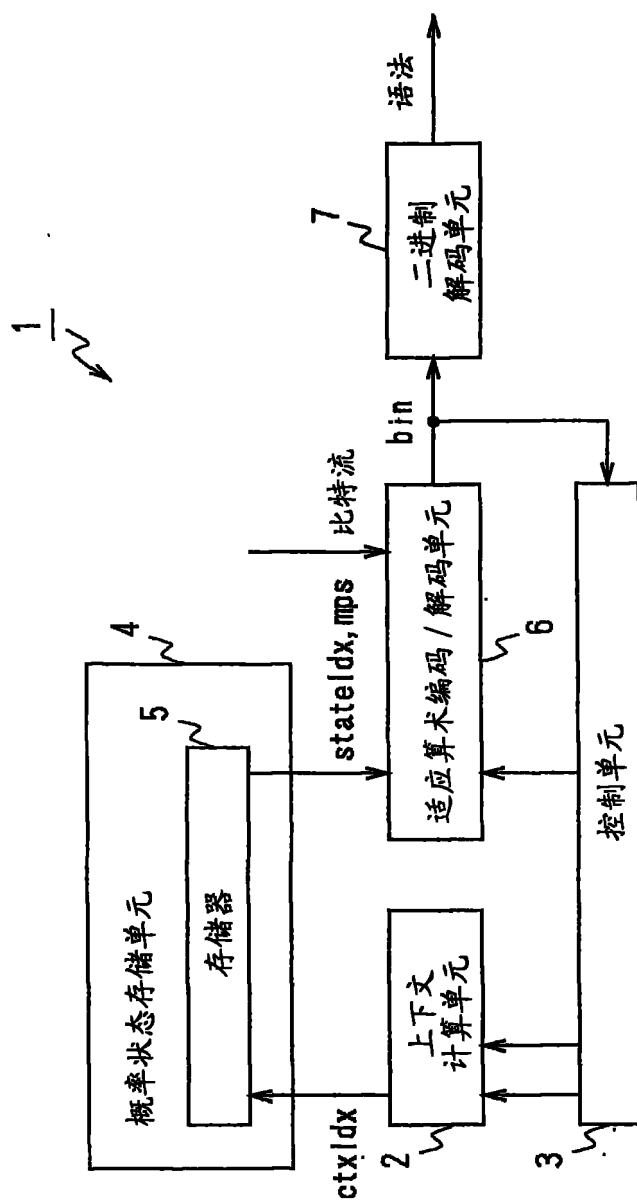


图 3

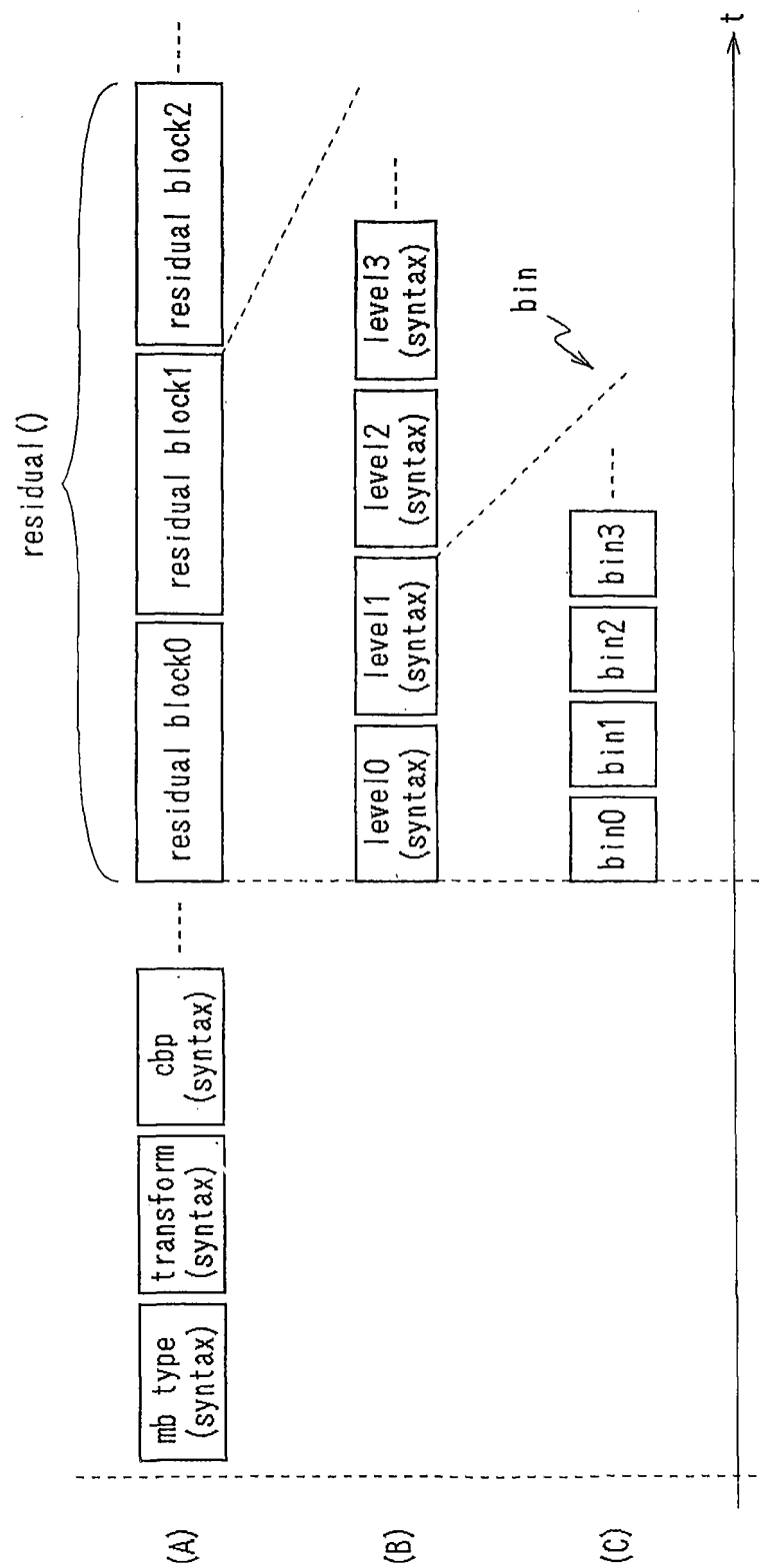


图 4

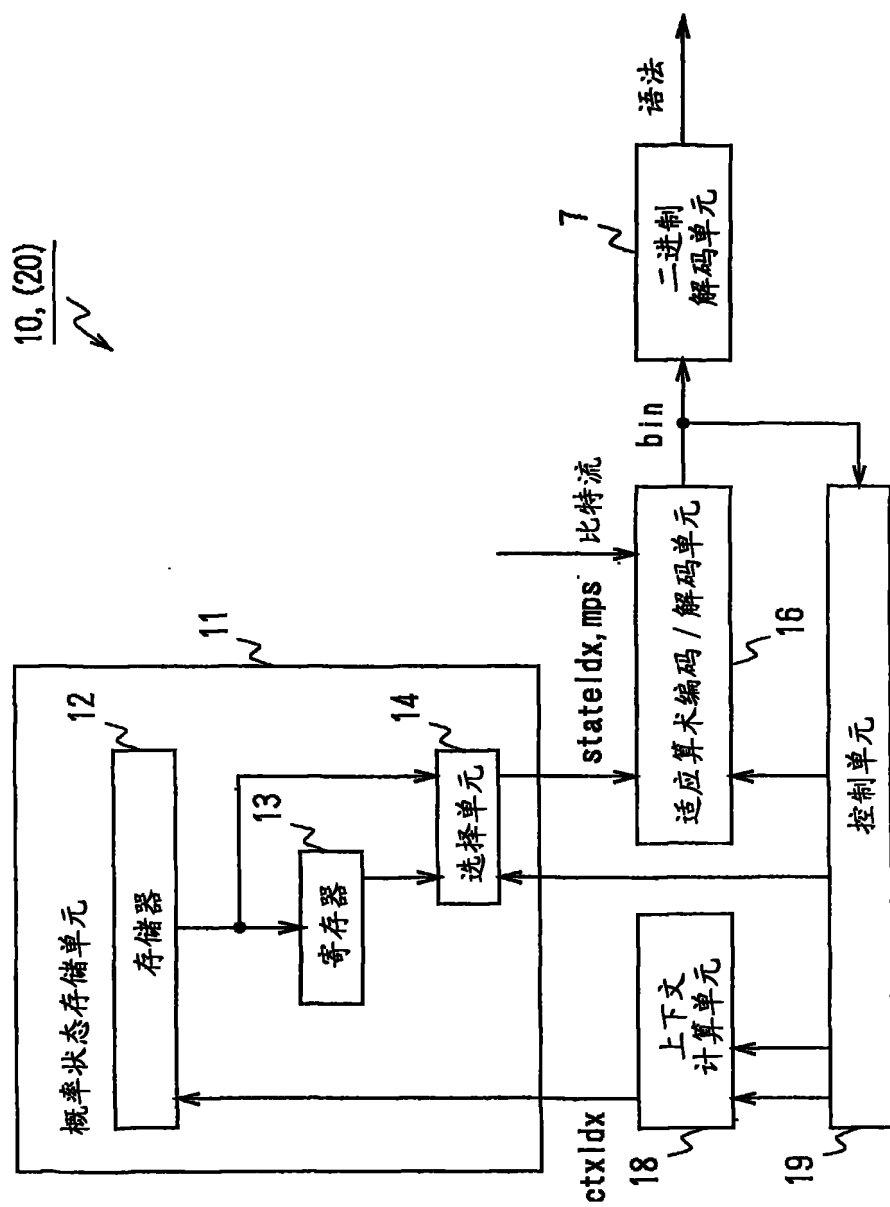
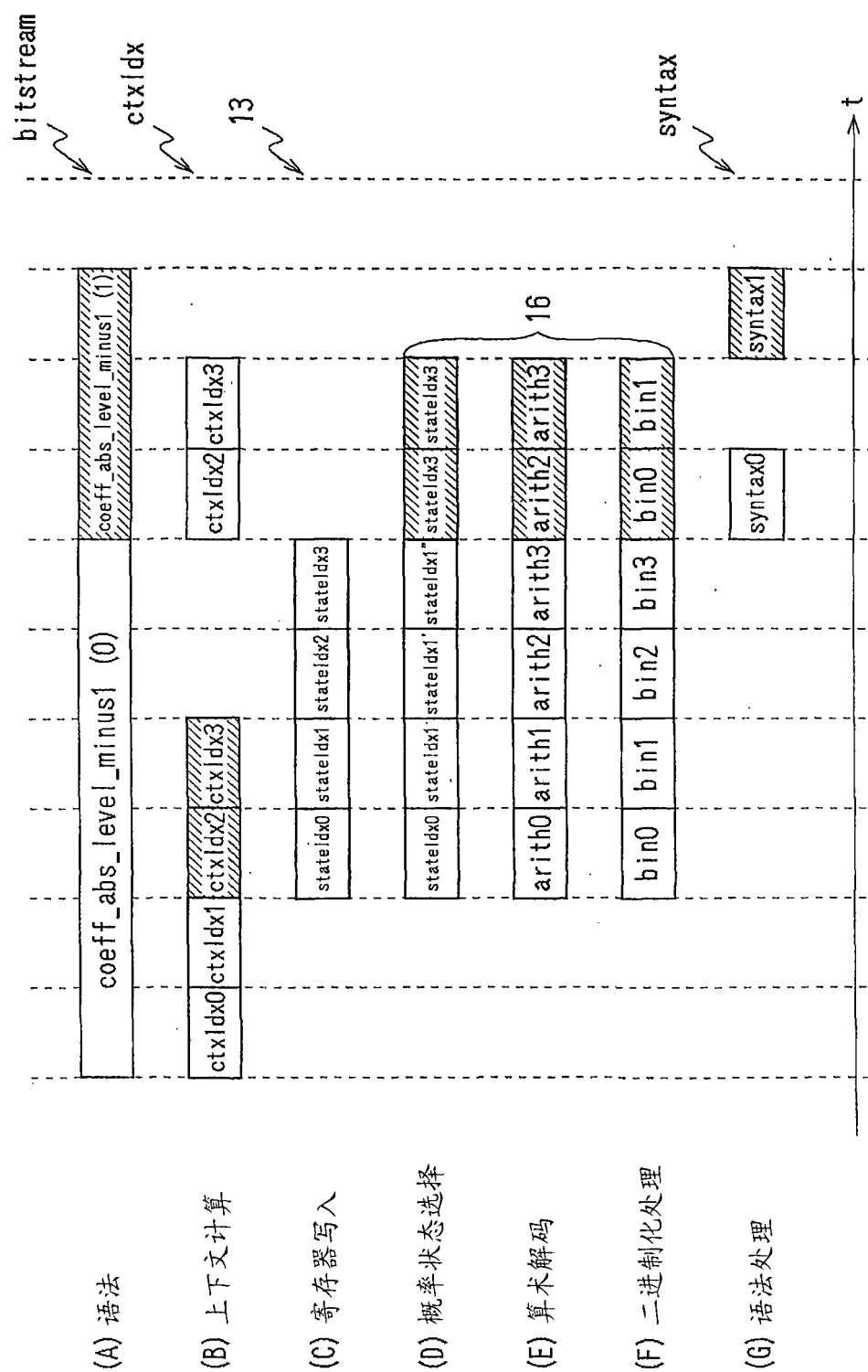


图 6



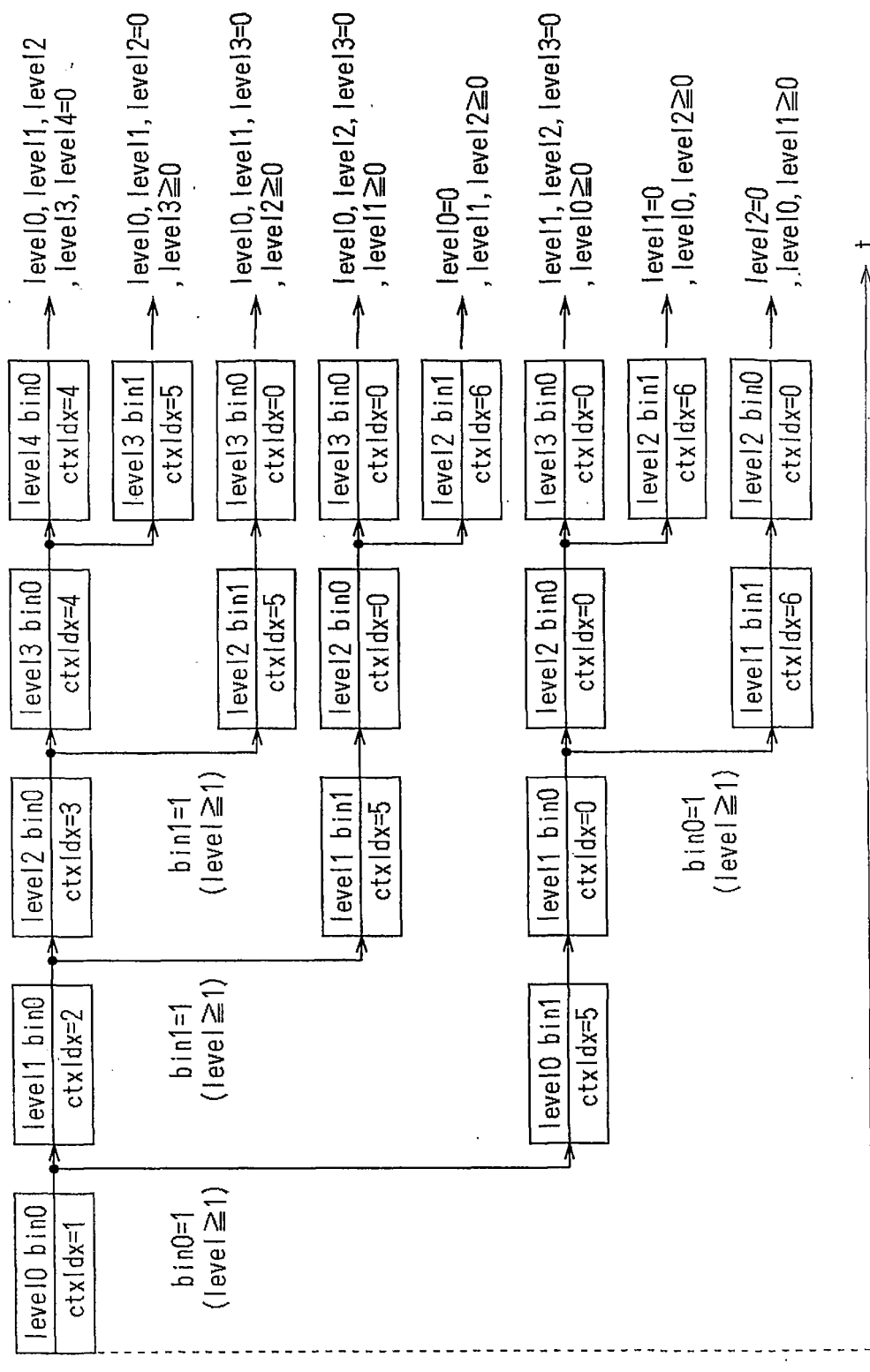


图 8

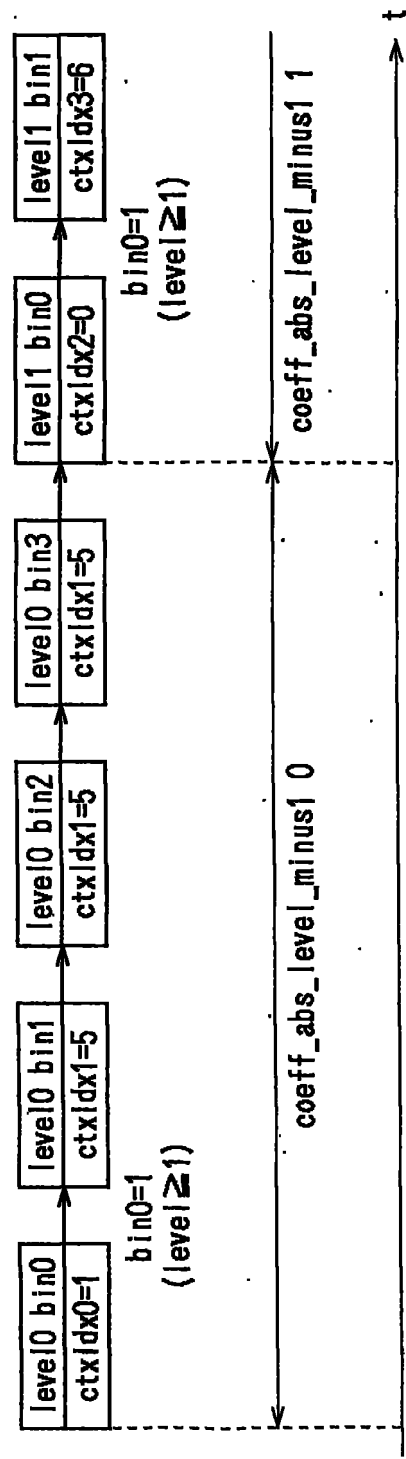


图 9

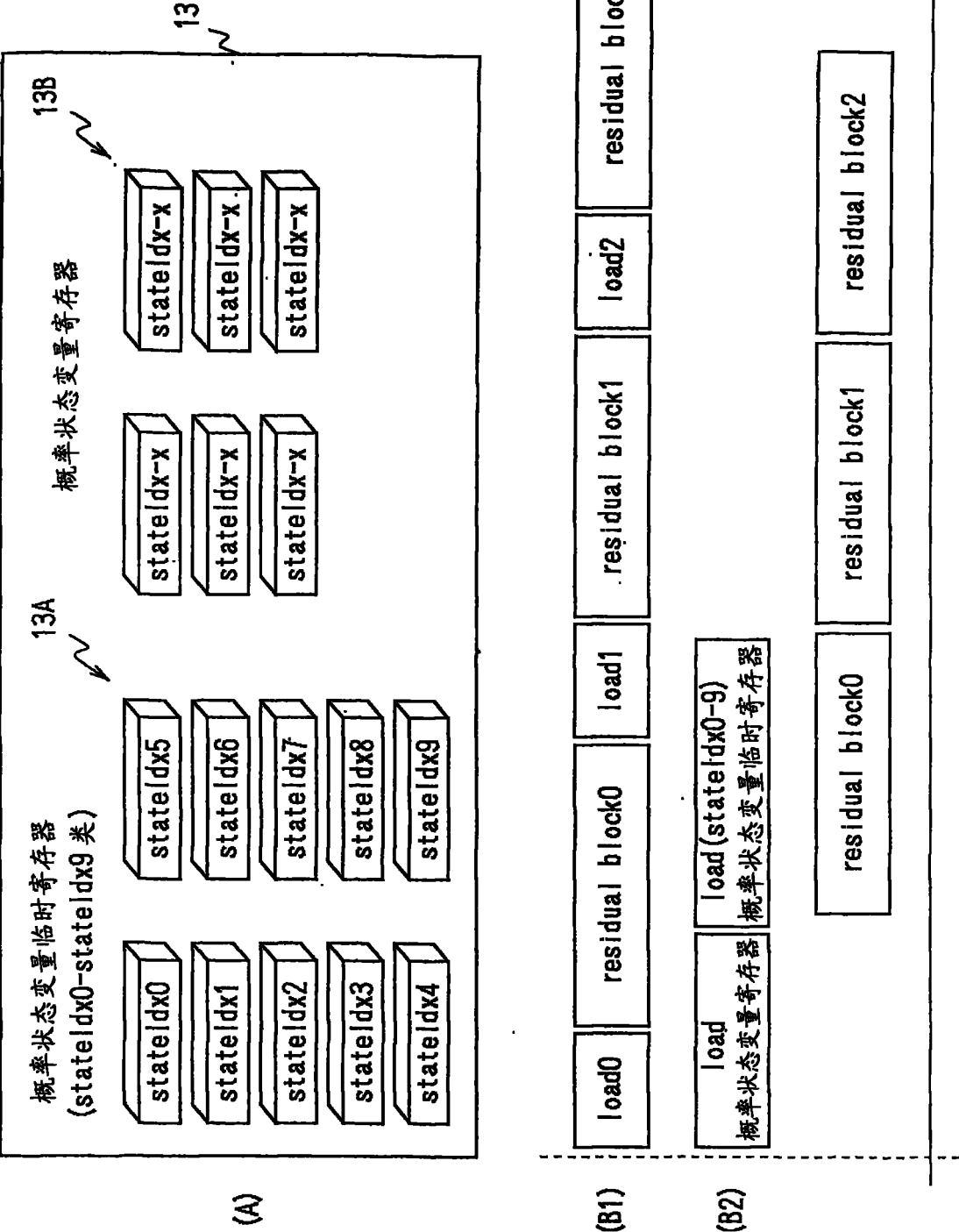


图 10

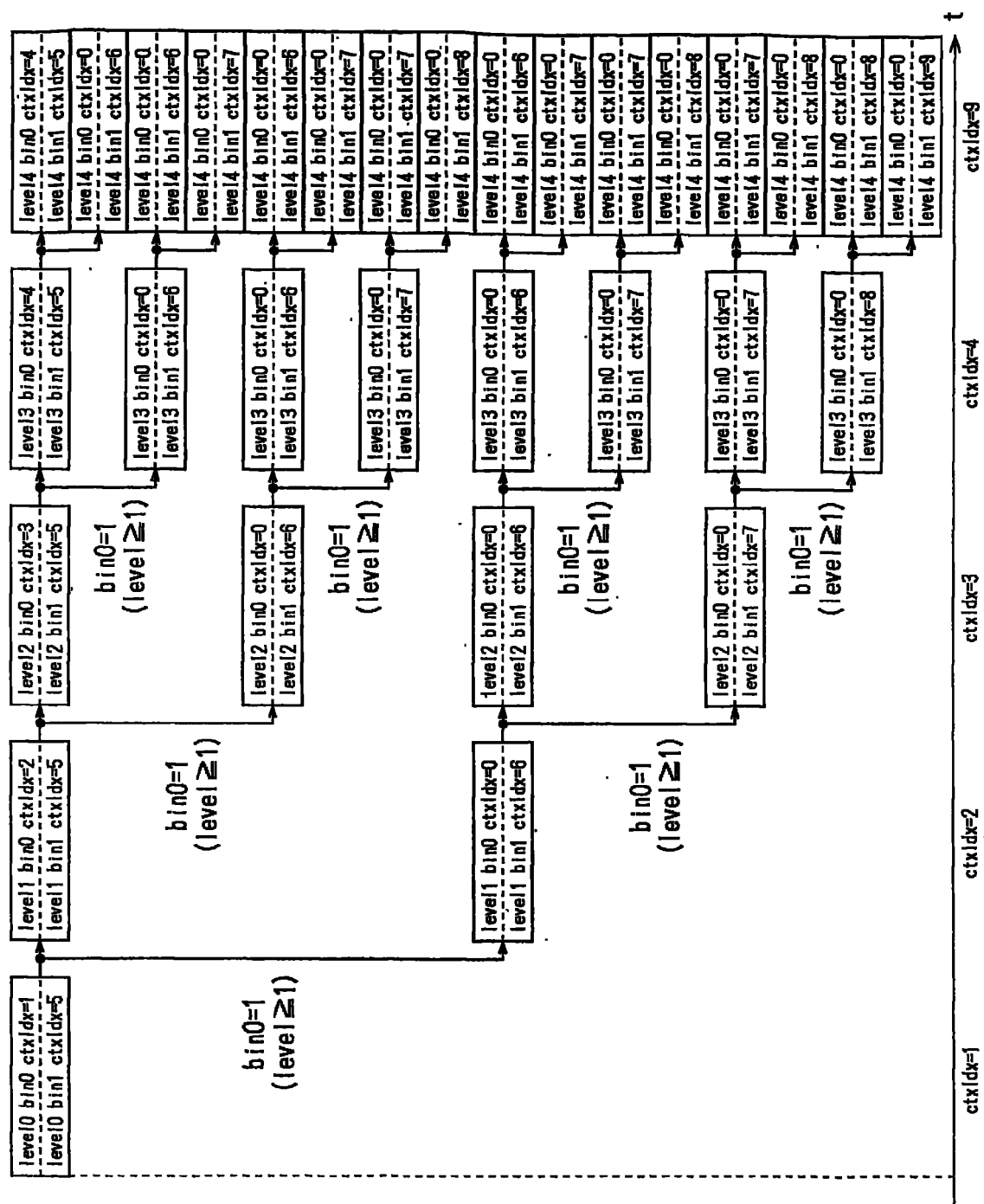


图 11

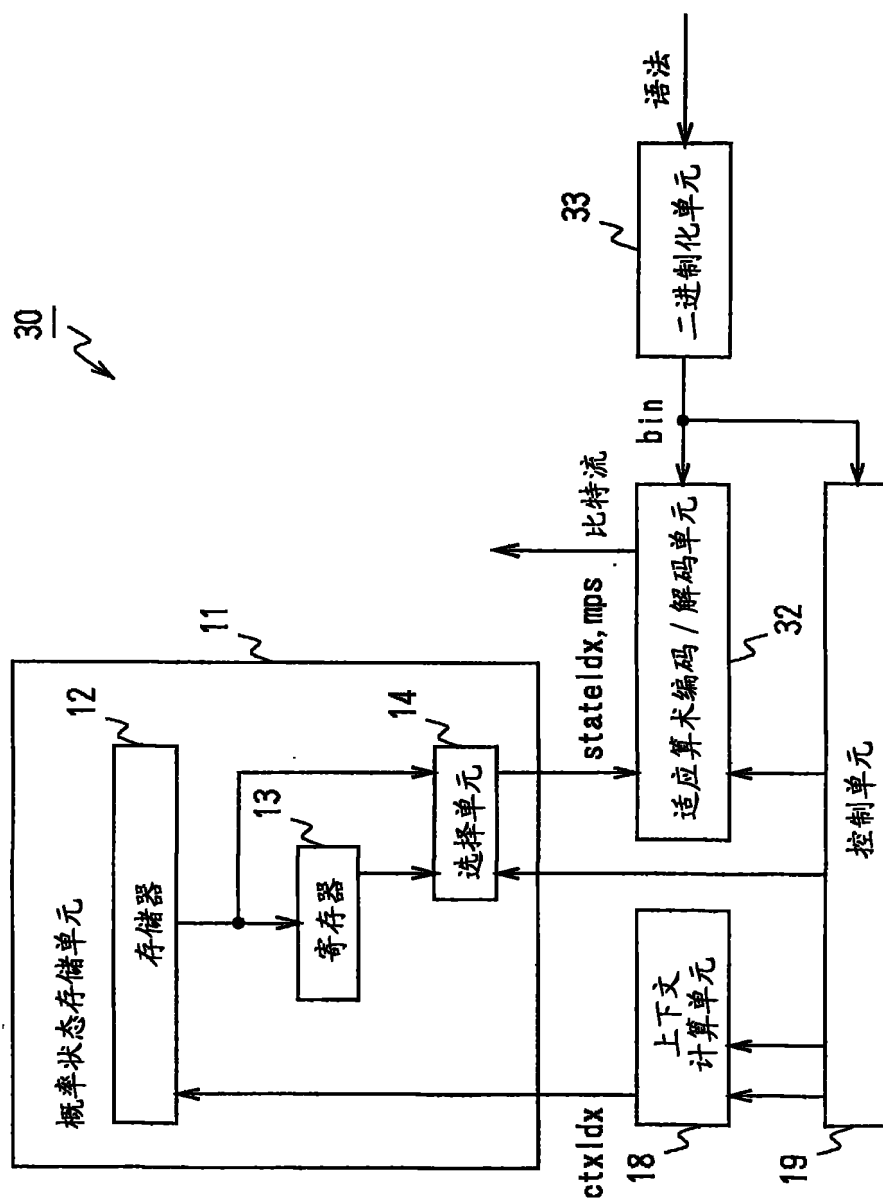


图 12

符号说明 1、10、20 解码设备 2、18 上下文计算单元 3、19 控制单元 4、11 概率状态存储单元 5、12 存储器 6、16 适应算术编码 / 解码单元 7 二进制解码单元 13 寄存器 13A 概率状态变量临时寄存器 13B 概率状态变量寄存器 14 选择单元 30 编码设备 32 适应算术编码单元 33 二进制制化单元