

# 适用于HEVC标准的二进制算术 编码模块

申请号：[201510298848.1](#)

申请日：2015-06-03

**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](#) [程魏](#) [郭勇](#) [黄磊磊](#) [金怡泽](#) [曾晓洋](#)  
**主分类号** [H04N19/13\(2014.01\)I](#)  
**分类号** [H04N19/13\(2014.01\)I](#) [H04N19/103\(2014.01\)I](#)  
**公开(公告)号** 104918049A  
**公开(公告)日** 2015-09-16  
**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 104918049 A

(43) 申请公布日 2015. 09. 16

(21) 申请号 201510298848. 1

(22) 申请日 2015. 06. 03

(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 程魏 郭勇 黄磊磊

金怡泽 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 19/13(2014. 01)

H04N 19/103(2014. 01)

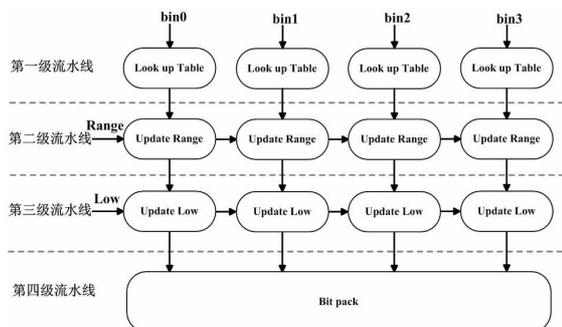
权利要求书1页 说明书3页 附图1页

(54) 发明名称

适用于 HEVC 标准的二进制算术编码模块

(57) 摘要

本发明属于数字高清视频压缩编解码技术领域,具体为一种适用于 HEVC 标准的二进制算术编码模块。本发明设计的二进制算术编码模块分为 4 级流水线,其中,第一级根据待编码字符 bin 的状态索引查表得到 RangeLPS 和 shift 候选值;第二级用于更新 Range 值;第三级用于更新 low 值;最后一级用于将 low 值不在变化的比特移出打包作为输出;本发明采用每个时钟周期处理 4 个编码字符的编码方式,并对二进制算术编码中的 Regular 模式, Bypass 模式, Terminal 模式分开处理,极大的提高了系统的吞吐率。



1. 一种适用于 HEVC 标准的二进制算术编码模块,其特征在于:所述二进制算术编码模块分为 4 级流水线,其中,第一级根据待编码字符 bin 的状态索引查表得到 RangeLPS 和 shift 候选值;第二级用于更新 Range 值;第三级用于更新 low 值;最后一级用于将 low 值不在变化的比特位移出打包作为输出;

二进制算术编码模块每个周期处理 4 个待编码字符 bin;每个 bin 的输入包括其状态索引 PState,当前小概率字符 LPS 以及当前 bin 的编码模式 MODE;

二进制算术编码中有四种编码模式:Regular 模式,Bypass 模式,Multi-bypass 模式和 Terminal 模式,其中 Bypass 模式和 Multi-bypass 模式合为一种模式 Mbyypass,并在每级流水线中将 Mbyypass、Regular 和 Terminal 模式分开处理,以减小关键路径长度。

2. 根据权利要求 1 所述的适用于 HEVC 标准的二进制算术编码模块,其特征在于:每个输入 bin 的计算流程为:

在第一级流水线中,根据编码字符 bin 的状态索引 PState 对表格 Range Table 和 Shift Table 进行查表,分别得到 4 个 RangeLPS 和 Shift 候选值;并对这些候选值,输入的小概率字符 LPS 以及当前 bin 的模式 MODE 用寄存器保存下来,进入第二级流水线;

在第二级流水线中,首先用前一个 bin 的 Range 值作为 4-1 选择器的输入从 4 个 RangeLPS 和 Shift 候选值中选出正确的 RangeLPS 和 Shift 值,然后根据这两个值以及前一个 bin 的 Range 值分别算出 Regular 模式,Mbyypass 模式,Terminal 模式以及 Invalid 模式下的当前 bin 的 Range 值;Invalid 模式表示当前输入 bin 无效,Range 值保持不变;最后根据 MODE 值对 4 种模式下的 Range 值进行 4 选 1 操作;

在第三级流水线中,根据第二级流水线中计算出的 Range 值,Shift 值以及前一个 bin 的 Low 值分别算出 Regular 模式,Mbyypass 模式,Terminal 模式以及 Invalid 模式下的当前 bin 的 Low 值,然后根据 MODE 值对 4 种模式下的 Low 值进行 4 选 1 操作;

在第四级流水线中,根据第三级流水线中得到 Low 值对输出码流进行打包输出。

## 适用于 HEVC标准的二进制算术编码模块

### 技术领域

[0001] 本发明属于数字高清视频压缩编解码技术领域,针对 HEVC 视频编解码标准,具体涉及一种适用于 HEVC 视频编码标准的、二进制算术编码模块。

### 背景技术

[0002] 作为下一代视频编解码标准,HEVC (High Efficiency Video Coding) 是于 2013 年由国际电信组织 (ITU) 和运动图像专家组 (MPEG) 联合成立的组织 JCTVC 所提出。其目标是,与上一代标准 H. 264/AVC 相比,在相同的视觉效果的前提下,比特率减少 50%。

[0003] 上一代视频编解码标准 H. 264/AVC 采用了两种编码方式:基于上下文的可变长编码 CAVLC 和基于上下文的二进制算术编码 CABAC。而在 HEVC 标准的软件模型 HM 前期版本中也是采用了这两种编码方式,只是在编码过程,编码语法元素上做了修改。但是大量测试发现,在相同的编码内容上,CABAC 比 CAVLC 节省了 8%~14% 码流,因此在 HM6.0 以后 HEVC 只保留了 CABAC 一种编码方式而不再支持 CAVLC 编码。

[0004] 在 HEVC 中采用的二进制算术编码模块是基于传统的算术编码改进而来。该编码方式有以下特点:

(1) 无损编码方式,适用于图像压缩处理;

(2) 被编码信息数据的符号序列表示成实数 0 和 1 之间的一个小数,无论信息有多长,其输出仅仅是一个数,而且是一个介于 0 和 1 之间的二进制小数;

(3) 在编码过程中需要不断计算被编码字符的概率。

[0005] 算术编码过程可总结如下:

(1) 编码器在开始时将“当前间隔” $[L, H]$  设置为  $[0, 1]$ ;

(2) 对每一个编码字符编码器在当前间隔进行如下递归处理:

(a) 将“当前间隔”按照各个编码字符的概率进行分割;

(b) 选中当前待编码字符编码器的分割区间为“当前间隔”,并回到(a)继续编码;

(3) 最后输出的“当前间隔”的下边界就是给定编码序列的算术编码。

[0006] 在硬件当中,整数计算比浮点数计算简单方便,而且在计算编码字符的概率当中除法也需要消耗较大的资源,因此 HEVC 中采用的二进制算术编码对传统的算术编码进行了以下改进:

(1) 编码器在开始时将“当前间隔” $[0, 1]$  改为  $[0, 510]$ ;

(2) 对当前的待编码字符概率计算由传统的除法运算改为查表计算,并设置了有 64 个状态的状态表,通过当前待编码字符在状态表中的索引查表计算概率;

(3) HEVC 建立 399 个上下文概率模型,不同的语法元素,同一个语法元素二进制序列的不同位对应不同的上下文概率模型,通过概率模型查表可得到相应的状态表索引。

### 发明内容

[0007] 本发明的目的在于设计一种工作频率高、吞吐率高的适用于 HEVC 标准的二进制

算术编码模块。

[0008] 本发明设计的二进制算术编码模块,如图 1 所示,分为 4 级流水线,其中,第一级根据待编码字符 bin 的状态索引查表,得到 4 个 RangeLPS 候选值以及相应的 4 个移位 shift 候选值;第二级用于更新 Range 值;第三级用于更新 low 值;最后一级用于将 low 值不在变化的比特位移出,打包作为输出。每一级流水线具体计算如下:

第一级计算 RangeLPS。为了折中由乘法带来的计算复杂度,CABAC 将所有的 RangeLPS 可能取值保存到一个 64x4 的表格内,而每个待编码字符 bin 的 RangeLPS 值需要根据其状态索引 PState 和当前 Range 值通过查找表格得到。因为第 2、第 3、第 4 个带编码字符 bin 的 Range 值在第一级流水线还不能计算出,所以不能确定具体的 RangeLPS,为了不影响后续流水线的操作,这里只通过 PState 变量将 4 个 RangeLPS 全部取出,在第二级流水线 Range 值确定时再确定具体的 RangeLPS 的值。

[0009] 第二级计算划分区间 Range, Range 的计算公式由当前待编码字符 bin 和 MPS 来决定,当 bin 不等于 MPS 时则取概率区间 RangeLPS。在编码过程中当 Low 的值过小时,会对其进行修正操作,保证二进制计算过程正确进行,为了保证 Range 和 Low 值能一一对应,对 Range 也要进行同样的计算操作,以保证计算过程在合理的区间范围内。如果将修正过程放到最后,就不能实现算术编码的四级流水,所以该设计中将修正过程前置,在更新 Range 的过程中完成 Range 修正重置计算,同时修正过程中用到的移位 shift 值由第一级查表得到。

[0010] 第三级计算划分区间下限 Low 值,同 Range 的计算过程类似,Low 值的计算也是由 bin 和 MPS 来决定,并且在计算 Low 值的过程中还要完成修正的操作过程,经过逐步的移位操作,当计算达到一定程度时,Low 值的高位数字不再变化,这时可以将它的高位输出到最终的 bit 流中去。

[0011] 第四级将 Low 的更新过程中移位出来的 bit 进行打包,形成最终的码流。

[0012] 二进制算术编码模块每个周期处理 4 个待编码字符 bin;每个 bin 的输入包括其状态索引 PState,当前小概率字符 LPS 以及当前 bin 的模式 MODE,如图 2 所示。

[0013] 二进制算术编码中有四种编码模式:regular 模式,bypass 模式,multi-bypass 模式和 terminal 模式,本发明将 bypass 模式,multi-bypass 模式合为一种模式 mbyypass,并在每级流水线中将 mbyypass,regular 和 terminal 模式分开处理,减小关键路径长度。

## 附图说明

[0014] 图 1:4 级编码流水结构。

[0015] 图 2:单个 bin 的算术编码流水结构。

## 具体实施方式

[0016] 下面结合附图,对本发明做进一步的描述。

[0017] 首先结合附图 1,介绍 4 个输入之间的数据依赖性。

[0018] 在二进制算术编码模块设计中,每个时钟周期处理 4 个待编码字符 bin,如图 1 所示。每个 bin 的输入包括其状态索引 PState,当前小概率字符 LPS 以及当前 bin 的模式 MODE,如图 2 所示。

[0019] 在第一级流水线中,根据输入 PState 值会对表格 Range Table 和 Shift Table 进行查表,分别得到 4 个 RangeLPS 和 Shift 候选值。在本级中所用到数据只依赖于输入,各个 bin 之间无数据依赖性,因此 4 个 bin 的第一级是完全并行的。

[0020] 在第二级流水线中,4 个 bin 都要更新 Range,并且要用到前一个 bin 的更新后的 Range 值,因此 4 个 bin 之间存在严重数据依赖性:只有前一个 bin 的 Range 计算完成之后才能更新当前 bin 的 Range 值。

[0021] 第三级同第二级流水线相同,4 个 bin 都要更新 Low,并且要用到前一个 bin 的更新后的 Low 值,因此 4 个 bin 之间存在严重数据依赖性:只有前一个 bin 的 Low 计算完成之后才能更新当前 bin 的 Low 值。

[0022] 4 个输入 bin 的第四级流水线是合在一起计算完成 bit pack 的,没有数据依赖性。

[0023] 然后结合附图 2,介绍每个输入 bin 的计算过程。

[0024] 在第一级流水线中,根据编码字符 bin 的状态索引 PState 对表格 Range Table 和 Shift Table 进行查表,分别得到 4 个 RangeLPS 和 Shift 候选值;并对这些候选值,输入的小概率字符 LPS 以及当前 bin 的模式 MODE 用寄存器保存下来,进入第二级流水线。

[0025] 在第二级流水线中,首先用前一个 bin 的 Range 值作为 4-1 选择器的输入从 4 个 RangeLPS 和 Shift 候选值中选出正确的 RangeLPS 和 Shift 值,然后根据这两个值以及前一个 bin 的 Range 值分别算出 Regular 模式, Mbyypass 模式, Terminal 模式以及 Invalid 模式下的当前 bin 的 Range 值,图中 Invalid 模式表示当前输入 bin 无效, Range 值保持不变;最后根据 MODE 值对 4 种模式下的 Range 值进行 4 选 1 操作。

[0026] 在第三级流水线中,根据第二级流水线中计算出的 Range 值, Shift 值以及前一个 bin 的 Low 值分别算出 Regular 模式, Mbyypass 模式, Terminal 模式以及 Invalid 模式下的当前 bin 的 Low 值,然后根据 MODE 值对 4 种模式下的 Low 值进行 4 选 1 操作。

[0027] 在第四级流水线中,根据第三级流水线中得到 Low 值对输出码流进行打包输出。

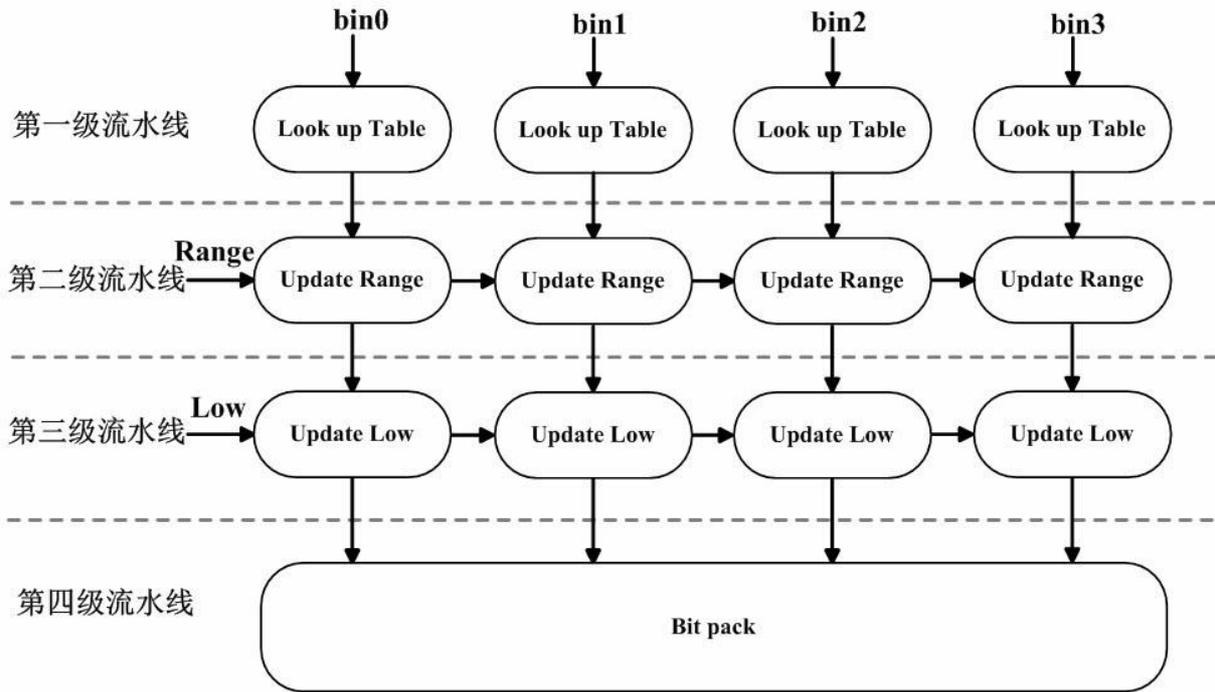


图 1

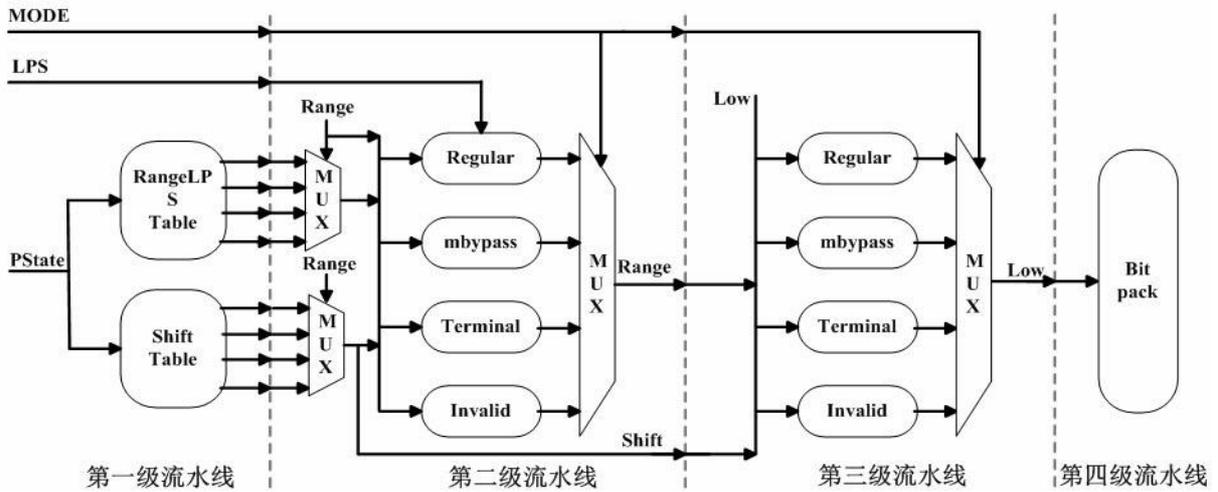


图 2