

适用于HEVC标准的熵编码上下文概率模型建模模块设计方法

申请号：[201510298837.3](#)

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 江亲伟 陆彦珩 曾晓洋](#)
主分类号 [H04N19/13\(2014.01\)I](#)
分类号 [H04N19/13\(2014.01\)I](#) [H04N19/70\(2014.01\)I](#)
公开(公告)号 [104918048A](#)
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 104918048 A

(43) 申请公布日 2015. 09. 16

(21) 申请号 201510298837. 3

(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 江亲伟

陆彦珩 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 19/70(2014. 01)

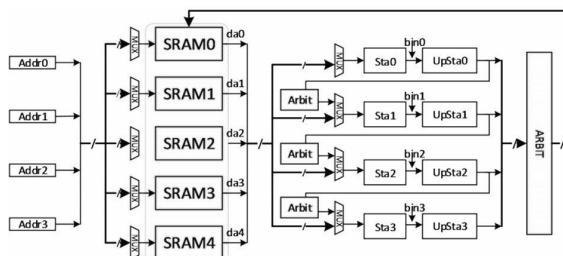
权利要求书2页 说明书4页 附图1页

(54) 发明名称

适用于 HEVC 标准的熵编码上下文概率模型建模模块设计方法

(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 标准的熵编码上下文概率模型建模模块的设计方法。HEVC 使用的基于上下文的二进制算术编码,并独特的设计了 399 个上下文概率模型。在实际编码中,不同的语法元素,相同的语法元素的二进制序列的不同位都可能使用不同的上下文概率模型。本发明使用 HEVC 标准的熵编码建立的 399 个上下文概率模型中的 280 个上下文概率模型,能够正确解码;这些上下文概率模型支持 I 帧与 P 帧;采用 5 块 SRAM 和 2 个寄存器存储相关上下文概率模型,并为 SRAM 设计仲裁结构,能够准确判断概率模型的地址和待写入的数据,并顺利读取数据。



1. 一种适用于 HEVC 标准的熵编码上下文概率模型建模模块的设计方法,其特征在于具体步骤为:

(1) 使用 HEVC 标准的熵编码建立的 399 个上下文概率模型中的 280 个上下文概率模型,能够正确解码;这些上下文概率模型支持 I 帧与 P 帧;

(2) 使用 5 块双端口 SRAM 以及 2 个寄存器存储 280 个上下文概率模型,每块 SRAM 的深度都为 32,宽度为 7;280 个上下文概率模型在 SRAM 中的分布见表 1 所示,其中,横向表头代表 5 块 SRAM,竖向表头代表各 SRAM 的地址,表中斜杠“/”前面的数字表示 I 帧中存储的索引,斜杠“/”后面的数字表示 P 帧中存储的索引;对于单独的 I 帧或 P 帧,160 个上下文概率模型已经足够,另外 160 个上下文概率模型在 SRAM 中的存放经过独特的设计以保证每个时钟周期都能对四组输入进行上下文建模;

(3) 为每块 SRAM 的读写地址设置仲裁器,以避免 SRAM 读写冲突,仲裁器对每个输入中的地址信息进行仲裁,判断不同的输入之间是否发生地址冲突,如果没有冲突现象,将直接根据当前地址去存储器中读取数据;

(4) 以流水线方式连续的从 SRAM 或者寄存器中读写正确的数据,其中,对连续两个周期内的输入也设计仲裁器,进行仲裁,即从第一个周期中读取的数据更新后的值与第二个周期读取的数据进行选择;

(5) 每个时钟周期支持对 4 组输入进行上下文概率模型建模;这 4 组输入中可能存在如下情况:2 或更多的输入需要从同一块 SRAM 的同一个地址读取数据,这样将导致这些输入读取数据都是相同的而发生错误;实际上第二个输入读取的数据是第一个读取的数据更新后的值,第三个输入取的数据是第二个读取的数据更新后的值,第四个输入取的数据是第三个读取的数据更新后的值;因此,对同一周期从同一块 SRAM 的同一个地址读取数据的多个输入进行仲裁,以保证读取数据的正确;

表 1:280 个上下文概率模型在 SRAM 中的分布

地址	SRAM0	SRAM1	SRAM2	SRAM3	SRAM4
0	111 / 153	138 / 138	138 / 94	79 / 79	136 / 121
1	110 / 125	141 / 111	94 / 149	153 / 124	125 / 183
2	125 / 110	154 / 154	139 / 139	138 / 107	153 / 153
3	125 / 125	110 / 110	140 / 140	154 / 154	125 / 183
4	111 / 110	140 / 195	63 / 108	139 / 139	153 / 153
5	111 / 95	127 / 111	140 / 140	124 / 194	125 / 183
6	108 / 108	143 / 111	63 / 108	153 / 79	153 / 153
7	110 / 125	79 / 94	134 / 61	109 / 78	111 / 154
8	125 / 110	123 / 123	139 / 153	127 / 111	110 / 153
9	125 / 125	110 / 110	153 / 167	154 / 154	124 / 123
10	111 / 110	140 / 195	141 / 140	124 / 194	136 / 151
11	111 / 95	127 / 111	125 / 154	153 / 79	136 / 151
12	108 / 108	143 / 111	141 / 140	109 / 78	138 / 107
13	91 / 121	79 / 94	125 / 154	127 / 111	153 / 167
14	182 / 123	123 / 123	141 / 140	154 / 154	136 / 91
15	154 / 154	171 / 140	125 / 154	141 / 154	167 / 22
16	107 / 166	152 / 107	125 / 139	140 / 170	152 / 107
17	179 / 136	110 / 139	94 / 123	138 / 167	152 / 167
18	107 / 166	108 / 63	124 / 153	139 / 182	154 / 154
19	179 / 136	139 / 183	111 / 140	192 / 136	153 / 153

20	107 / 166	139 / 183	111 / 140	152 / 137	200 / 185
21	179 / 136	92 / 296	137 / 196	182 / 167	154 / 154
22	154 / 154	152 / 152	138 / 167	197 / 182	154 / 154
23	140 / 154	74 / 134	149 / 149	154 / 154	154 / 154
24	140 / 154	107 / 121	122 / 136	154 / 154	154 / 154
25	153 / 182	179 / 194	166 / 166	139 / 107	154 / 154
26	139 / 153	227 / 167	122 / 139	184 / 154	154 / 154
27	140 / 169	154 / 154	154 / 154	185 / 185	154 / 154
28	140 / 154	154 / 154	157 / 126	154 / 154	154 / 154
29	154 / 154	149 / 134	139 / 154	141 / 139	154 / 154
30	184 / 197	63 / 154	201 / 201	182 / 123	154 / 198
31	154 / 168	154 / 140	154 / 154	111 / 155	154 / 154

适用于 HEVC 标准的熵编码上下文概率模型建模模块设计 方法

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 标准的熵编码上下文概率模型建模模块的设计方法。

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时,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] 本发明的目的在于提出一种工作频率高、吞吐率高的适用于 HEVC 标准的熵编码上下文概率模型建模模块的设计方法。

[0008] 本发明提出熵编码上下文概率模型建模模块的设计方法,具体步骤为:

(1)HEVC 标准的熵编码建立了 399 个上下文概率模型,本发明使用了其中的 280 个上下文概率模型,使用这些上下文概率模型进行编码能够正确解码。在 HEVC 中有三种类型帧: I 帧,P 帧,B 帧,这三种帧一共使用 399 个上下文概率模型,而本发明支持的 I 帧与 P 帧,使用了其中的 280 个上下文概率模型。

[0009] (2)使用 5 块双端口 SRAM 以及 2 个寄存器存储 280 个上下文概率模型,每块 SRAM 的深度都为 32,宽度为 7。280 个上下文概率模型在 SRAM 中的分布见表 1 所示,其中,横向表头代表 5 块 SRAM,竖向表头代表各 SRAM 的地址,表中斜杠“/”前面的数字表示 I 帧中存储的索引,斜杠“/”后面的数字表示 P 帧中存储的索引。本发明支持的 I 帧与 P 帧是独立的,不支持 I 帧中帧间预测,也不支持 P 帧中帧内预测,因此对于单独的 I 帧或 P 帧 160 个上下文概率模型已经足够,并不需要建立 280 个上下文概率模型。另外 160 个上下文概率模型在 SRAM 中的存放经过独特的设计以保证每个时钟周期都能对四组输入进行上下文建模。

[0010] (3)为避免 SRAM 读写冲突,为每块 SRAM 的读写地址设置了仲裁器,仲裁器对每个输入中的地址信息进行仲裁,判断不同的输入之间是否发生地址冲突(同一时钟周期出现一片 SRAM 有两个不同读地址信号)。如果没有冲突现象,将直接根据当前地址去存储器中读取数据。

[0011] (4)以流水线方式连续的从 SRAM 或者寄存器中读写正确的数据。以流水线进行上下文概率模型建模时有这样一种情况:相邻的两个周期内需要从同一块 SRAM 的同一个地址读取数据,这样第二个周期从 SRAM 中读取数据就是错误的,应该读取的是第一个周期读取的数据更新后的值。为避免这种情况发生,本发明对连续两个周期内的输入也进行了仲裁,即从第一个周期中读取的数据更新后的值与第二个周期读取的数据进行选择。

[0012] (5)每个时钟周期支持对 4 组输入进行上下文概率模型建模。这 4 组输入中可能存在这样一种情况:2 或更多的输入需要从同一块 SRAM 的同一个地址读取数据,这样将导致这些输入读取数据都是相同的而发生错误。实际上第二个输入读取的数据应该是第一个读取的数据更新后的值,第三个输入取的数据应该是第二个读取的数据更新后的值,第四个输入取的数据应该是第三个读取的数据更新后的值。为保证读取数据的正确,本发明对同一周期从同一块 SRAM 的同一个地址读取数据的多个输入进行仲裁。

附图说明

[0013] 图 1:上下文建模过程仲裁存储结构。

具体实施方式

[0014] 下面结合附图和附表,对本发明做进一步的描述。

[0015] 附表 1 中横向表头代表 5 块 SRAM,竖向表头代表各 SRAM 的地址,表中斜杠“/”前面的数字表示 I 帧中存储的索引,斜杠“/”后面的数字表示 P 帧中存储的索引。SRAM0~SRAM4 中分别存储了 32 个上下文模型。对于语法元素 sig_coff_flag 的第 33 位和第 34 位存储

在两个单独的寄存器中,其上下文模型索引为 152/107 与 136/121,斜杠“/”前后数字分别表示 I 帧与 P 帧中存储的索引。

[0016] 附图 1 是上下文建模过程仲裁存储结构。每个时钟有四组输入,其地址分别为 Add0, Add1, Add2 和 Add3。首先是一个多路选择器,对地址进行仲裁,将从同一块 SRAM 读取数据的输入分到同一组。然后每组输入从相应的 SRAM 中读取数据,然后此数据与前一个周期从此 SRAM 中读取到的数据更新后的值进行仲裁,避免相邻的两个周期内从同一块 SRAM 的同一个地址读取数据情况下读取到错误的数据的现象发生。之后根据第一个输入读取到数据进行查表,对第一个读取到数据进行更新。然后对第二个输入进行仲裁,如果第一个输入和第二个输入从同一块 SRAM 的同一个地址读取数据,那么第二个输入则选择第一个输入读取到的数据更新后的值进行查表对数据进行更新,否则则选择从 SRAM 中读取到数据进行查表对数据更新。同理,对第三个输入进行仲裁,如果第三个输入和第二个输入从同一块 SRAM 的同一个地址读取数据,那么第三个输入则选择第二个输入读取到的数据更新后的值进行查表对数据进行更新,如果第三个输入和第一个输入从同一块 SRAM 的同一个地址读取数据,那么第三个输入则选择第一个输入读取到的数据更新后的值进行查表对数据进行更新,否则则选择从 SRAM 中读取到数据进行查表对数据更新。最后对第四个输入进行仲裁,如果第四个输入和第三个输入从同一块 SRAM 的同一个地址读取数据,那么第四个输入则选择第三个输入读取到的数据更新后的值进行查表对数据进行更新,如果第四个输入和第二个输入从同一块 SRAM 的同一个地址读取数据,那么第四个输入则选择第二个输入读取到的数据更新后的值进行查表对数据进行更新,如果第四个输入和第一个输入从同一块 SRAM 的同一个地址读取数据,那么第四个输入则选择第一个输入读取到的数据更新后的值进行查表对数据进行更新,否则则选择从 SRAM 中读取到数据进行查表对数据更新。

[0017] 表 1 :280 个上下文概率模型在 SRAM 中的分布

地址	SRAM0	SRAM1	SRAM2	SRAM3	SRAM4
0	111 / 153	138 / 138	138 / 94	79 / 79	136 / 121
1	110 / 125	141 / 111	94 / 149	153 / 124	125 / 183
2	125 / 110	154 / 154	139 / 139	138 / 107	153 / 153
3	125 / 125	110 / 110	140 / 140	154 / 154	125 / 183
4	111 / 110	140 / 195	63 / 108	139 / 139	153 / 153
5	111 / 95	127 / 111	140 / 140	124 / 194	125 / 183
6	108 / 108	143 / 111	63 / 108	153 / 79	153 / 153
7	110 / 125	79 / 94	134 / 61	109 / 78	111 / 154
8	125 / 110	123 / 123	139 / 153	127 / 111	110 / 153
9	125 / 125	110 / 110	153 / 167	154 / 154	124 / 123
10	111 / 110	140 / 195	141 / 140	124 / 194	136 / 151
11	111 / 95	127 / 111	125 / 154	153 / 79	136 / 151
12	108 / 108	143 / 111	141 / 140	109 / 78	138 / 107
13	91 / 121	79 / 94	125 / 154	127 / 111	153 / 167
14	182 / 123	123 / 123	141 / 140	154 / 154	136 / 91
15	154 / 154	171 / 140	125 / 154	141 / 154	167 / 22
16	107 / 166	152 / 107	125 / 139	140 / 170	152 / 107
17	179 / 136	110 / 139	94 / 123	138 / 167	152 / 167
18	107 / 166	108 / 63	124 / 153	139 / 182	154 / 154
19	179 / 136	139 / 183	111 / 140	192 / 136	153 / 153
20	107 / 166	139 / 183	111 / 140	152 / 137	200 / 185
21	179 / 136	92 / 296	137 / 196	182 / 167	154 / 154
22	154 / 154	152 / 152	138 / 167	197 / 182	154 / 154
23	140 / 154	74 / 134	149 / 149	154 / 154	154 / 154
24	140 / 154	107 / 121	122 / 136	154 / 154	154 / 154
25	153 / 182	179 / 194	166 / 166	139 / 107	154 / 154
26	139 / 153	227 / 167	122 / 139	184 / 154	154 / 154
27	140 / 169	154 / 154	154 / 154	185 / 185	154 / 154
28	140 / 154	154 / 154	157 / 126	154 / 154	154 / 154
29	154 / 154	149 / 134	139 / 154	141 / 139	154 / 154
30	184 / 197	63 / 154	201 / 201	182 / 123	154 / 198
31	154 / 168	154 / 140	154 / 154	111 / 155	154 / 154

。

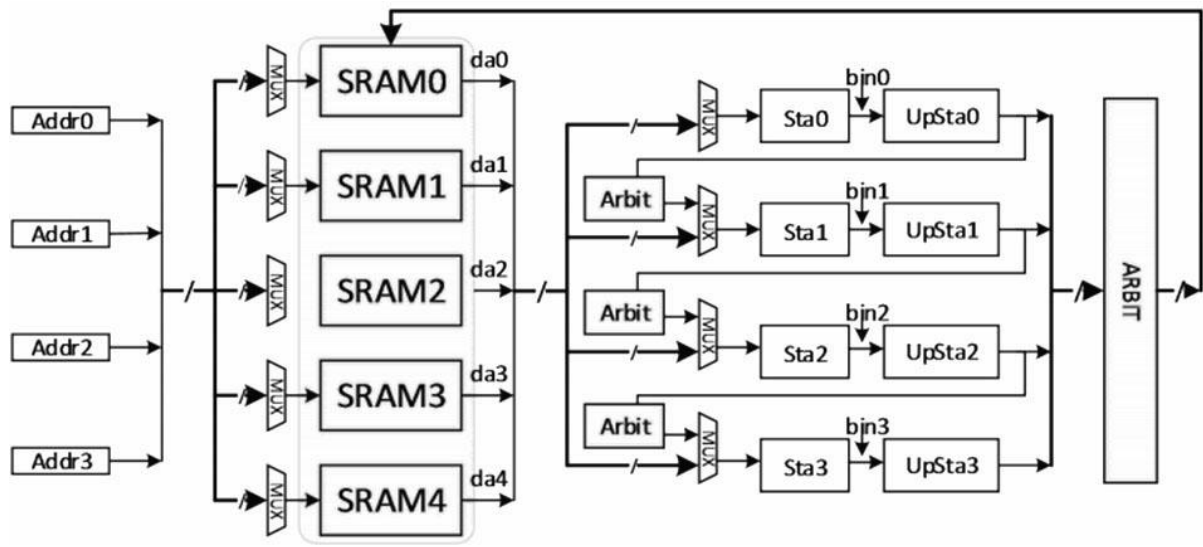


图 1