基于 AVS 标准的熵解码器设计*

赵龙辉，陈新华，任怀鲁
(山东科技大学，山东 青岛 266510)

摘 要：阐述了我国拥有自主知识产权的音视频编码技术标准——AVS 标准的熵解码算法，介绍了基于 AVS 标准的熵解码器的设计。根据码流的特点划分硬件模块，采用筒形移位器结构提高解码并行性，应用 Verilog 硬件描述语言、EDA 软件 ModelSim 仿真、QuartusII 软件综合，并通过了 Altera 公司的 Cyclone 系列 FPGA 芯片的下载验证，证明该设计能够实现 AVS 码流的实时解码功能。

关键词：AVS；熵解码；可编程逻辑门阵列；Verilog 硬件描述语言

中图分类号：TN919.8 **文献标识码**：A **文章编号**：1674-7720(2010)20-0053-03

Hardware design of variable length decoder based on AVS

ZHAO Long Hui, CHEN Xin Hua, REN Huai Lu
(Shandong University of Science and Technology, Qingdao 266510, China)

Abstract: This paper discusses the AVS standard which is the audio and video standard that we own independent intellectual property rights and the algorithm of variable length decoder for AVS standard. The design is separated into several parts according to the specialty of the code flow; Barrel shifter and other structures are used to improve decoding parallelism; The design has been implemented with Verilog HDL, Simulated using modelsim software, compilated using quatrusII software and verified by Altera's Cyclone series FPGA. The realized system can decode the AVS video in real-time.

Key words: AVS; variable length decoder; FPGA; Verilog HDL

AVS 标准是《信息技术 先进音视频编码》系列标准的简称，是我国具备自主知识产权的第二代信源编码标准。AVS 标准包括系统、视频、音频、数字版权管理等四个主要技术标准和一致性测试等支撑标准。

目前音视频产业可以选择的信源编码标准有四个：MPEG-2、MPEG-4、AVC(也称 JVT、H.264)和 AVS。从制订者分，前三个标准是由 MPEG 专家组完成的，第四个是我国自主制定的。从发展阶段分，MPEG-2 是第一代信源标准，其余三个为第二代标准。从主要技术指标——编码效率比较，MPEG-4 是 MPEG-2 的 1.4 倍，AVS 和 AVC 相当，都是 MPEG-2 的两倍以上。

AVC 标准中，对预测残差有两种熵编码的方式：基于上下文的自适应变长码(CAVLC)和基于上下文的自适应二进制算数编码(CABAC)；对于非预测残差，采用指数哥伦布码或 CABAC 编码，视编码器的设置而定[1]。

AVS 标准所用的熵编码技术相比于以前有了很多改进，它的语法元素和残差系数是由定长码和指数哥伦布码构成的，其中指数哥伦布码和语法元素之间存在多种映射关系。

1 AVS 标准熵解码算法描述

在整个 AVS 视频的解码过程中，熵解码模块位于系统的最前端，负责从压缩后的码流中解析出宏块头信息以及量化系数，供后续的帧内预测模块和帧间预测模块使用。而熵解码模块又可以大体分为两个部分：解析 K 阶指数哥伦布码部分和解析语法元素部分。

解析 K 阶指数哥伦布码时，首先从比特流的当前位置开始寻找第一个非零比特，并将找到的零比特个数记为 leadingZeroBits，然后根据 leadingZeroBits 计算 CodeNum。用伪代码描述如下：

```
leadingZeroBits=-1;
for (b=0; ! b; leadingZeroBits++)
    b=read_bits(1)
CodeNum=2^(leadingZeroBits+k)-2^k+read_bits(leadingZeroBits+k)[2]
```

由于 AVS 视频中所有的语法元素以及经过变换和量化的残差系数都是以指数哥伦布码的形式映射成二

* 基金项目：复旦大学专用集成电路与系统国家级重点实验室(09KF007)

进制码流的，因此在解析出 K 阶指数哥伦布码的 CodeNum 后，下一步就是要还原出各种语法元素和残差系数。

在 AVS 标准中规定了四种映射方式：ue(v)、se(v)、me(v)和 ce(v)。其中 ue(v)、se(v)和 me(v)所描述的语法元素采用 0 阶的指数哥伦布码，ce(v)用来描述残差系数，可以采用 0 阶、1 阶、2 阶或者 3 阶指数哥伦布码。它们的解析过程如下：

ue(v)：无符号直接映射，语法元素的值等于 CodeNum；

se(v)：有符号映射，映射关系为：当 CodeNum=k 时，语法元素值为 $(-1)^{k+1}\times Ceil(k\div 2)$

me(v)：分为 MbCBP 和 MbCBP422 两种模式，分别根据 CodeNum 的值，查找相对应的表来得到语法元素的值。

ce(v)：ce(v)描述的语法元素可以采用 0 阶、1 阶、2 阶或 3 阶指数哥伦布码进行解析，还有 19 个相关的码表，对于阶数的确定规则以及码表的切换规则，在 AVS 标准中都有详细的说明。解析时，首先语法元素 trans_coefficient 等于 CodeNum，如果 trans_coefficient 小于 59，可以根据 trans_coefficient 的值查找相关的码表得到残差系数；如果 trans_coefficient 大于等于 59，解析下一个 ce(v)语法元素，得到一个新的 CodeNum，escape_level_diff 等于 CodeNum，然后根据 trans_coefficient 和 escape_level_diff 求得残差系数[3]。

2 熵解码器的硬件设计

由于熵解码器位于整个解码结构的最前端，所有后续模块中需要用到的数据都是熵解码模块从原始码流中解析出来的，因此熵解码器性能的优劣直接影响到整个 AVS 视频解码器的性能。

从前面的介绍中可以了解到，经过指数哥伦布编码后形成的码流中，每个码的码长不固定，而且前后具有很大的相关性，这样在解码时就必须逐位读取数据、解析完一个码字后才能解析下一个码字。这种串行解码的方式严重限制了熵解码器的性能，所以需要找到一种能够并行解码的方式。这里的并行解码并不是指同时对好几个码进行解码，而是针对一个变长码的多个位来说的。具体来说就是一次读入 N 位数据，通过比较操作，得到码长，使得解码可以在确定的时间长度内进行，而不是随着码长的不同而变化。显然，这将提高硬件的复杂度，但是换来了解码速度的提高[4]。

整个熵解码器可以分为四个模块：数据准备模块、解指数哥伦布码模块、语法元素解析模块和解码控制模块。硬件结构如图 1 所示。

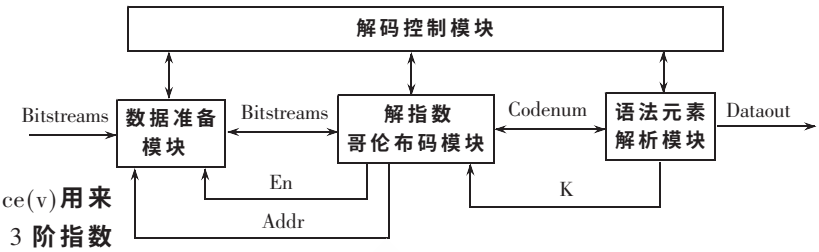

图 1 熵解码器硬件结构图

下面通过分析码流数据的解析过程来逐个说明各个模块的功能和实现。

2.1 数据准备模块

这个模块的功能是为后边的解指数哥伦布码模块准备好未解码数据，要求解指数哥伦布码模块解完一个码，数据准备模块就要在码流中将解完的码移走，并准备好下一个要解的码字送给下一模块。硬件结构如图 2 所示。

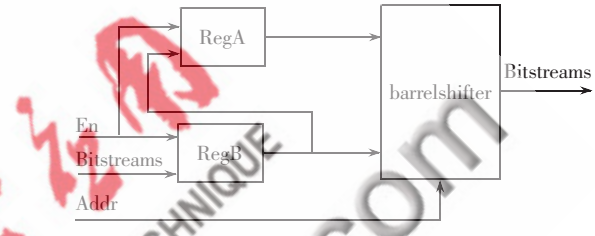

图 2 数据准备模块结构图

RegA 和 RegB 是两个 32 位的寄存器，其中 RegB 直接从未解码码流中读取数据，RegA 则接收 RegB 中的数据，两个寄存器共同组成 barrelshifter 的输入。

barrelshifter 是一个 64 位输入、32 位输出的暂存器，每次输出都将刚解完的一个码字移出，以保证输出的 32 位数据中最低位为下一个要解码的码字的开始。移位的位数由输入的地址 Addr 来决定，即输出 64 位输入的[Addr，Addr+31]位。

当 En 信号有效时，更新数据，将 RegB 中的数据存到 RegA 中，再从码流中读取新的 32 位数据，存到 RegB 中。其中 Addr 和 En 信号由解指数哥伦布码模块给出。

2.2 解指数哥伦布码模块

这个模块的功能是接收上一模块准备好的未解码数据，计算出从最低位开始的一个指数哥伦布码的 CodeNum，并根据这个码的码长算出筒形移位器的移位地址。模块结构如图 3 所示。

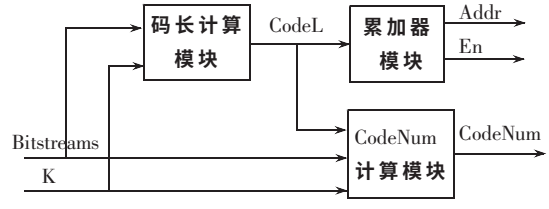

图 3 解指数哥伦布码模块结构图

由于上一个模块准备好的未解码码流是一个以新的指数哥伦布码的开始为最低位的 32 位码流，因此要计算出这个码的 CodeNum，首先要算出这个码的码长，然后在这 32 位数据中截取出这个码字，才可以计算它

的 CodeNum。根据指数哥伦布码的结构，码长与这个码的前导零个数 M 和阶数 K 有密切关系，可以总结出一个公式：$CodeL=2\times M+K+1$。码长计算模块就是根据这个原理来计算码长的，首先通过一组比较器检测从最低位开始有几个零，然后根据上述公式得到码长。对于阶数 K，解 ue(v)、me(v)和 se(v)语法元素时，阶数 K 为 0。解 ce(v)时，当前解码码字的阶数在解码上一个码字时可以得出。

累加器模块是一个模为 32 的加法器，将码长计算模块计算出来的码长 $CodeL$ 累加到上一次的移位地址上，得到一个新的移位地址，筒形移位器根据这个新的地址，将本次解码的码字移出，准备好下一个未解码码流。En 为进位输出，当累加器的 En 为 1 时，说明已经解完了 32 位数据，这时需要对 RegA 和 RegB 进行数据更新。

计算码字的 CodeNum 时，首先根据码长计算模块算出来的码字长度，从上一模块准备好的未解码码流中把本次要解码的码字截取出来，抛弃无用的位，然后再根据公式 $CodeNum=2^{leadingZeroBits+k}-2k+read\_bits(leadingZeroBits+k)$ 计算这个码字的 CodeNum。显然这个公式太过复杂，不适合直接硬件化，但是经过分析公式以及指数哥伦布码的结构，可以改造一下公式。因为指数哥伦布码的结构为前面 M 个前导 0，中间一个 1，后面 $M+K$ 位数据组成，而 read_bits(leadingZeroBits+k)即为后面 $M+K$ 位数据的码字值，2leadingZeroBits+k 正好为中间那个 1 的码字值，所以一个码字的 CodeNum 就等于这个码字的码字值减去 2k，这样对 CodeNum 的计算就大大简化了。

2.3 语法元素解析模块

顾名思义，这个模块就是最终解析出语法元素含义的模块，它接收上一模块计算出来的 CodeNum 值，然后主要通过查表的方式解析出这个码字所代表的语法元素含义，结构如图 4 所示。

这里主要介绍下最复杂的 ce(v)部分的解析过程。

逃逸码判断模块：在解析 ce(v)语法元素时，首先要判断是否为逃逸码，即 CodeNum<59 时，按照正常 ce(v)语法元素解析规则解析；CodeNum≥59 时，下一码字为逃逸码，则按照逃逸码解析规则解析。

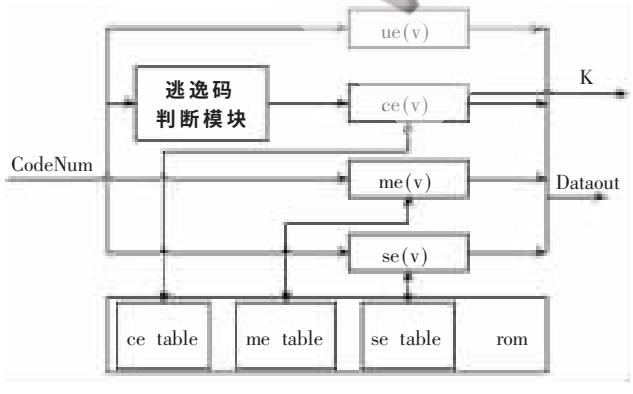

图 4 语法元素解析模块结构图

ce(v)语法元素解析模块：正常 ce(v)语法元素的解析过程为：以 CodeNum 的值为索引，查找当前码表，得到(run、level)，然后根据标准中的码表切换规则进行码表切换，并得到下一码字的阶数 K。解析逃逸码的解析过程为：令 transcoefficient 等于 CodeNum，得到 run=(transcoefficient-59)/2。以当前码表为索引，查表得到对应的 MaxRun，如果 run>MaxRun，则 RefAbslevel=1，否则以 run 为索引查当前码表，得到 RefAbslevel。解析下一个语法元素，得到一个新的 Codenum，escape_level_diff 等于 CodeNum。如果 transcoefficient 为奇数，则 level 等于-(RefAbslevel+escape_level_diff)；如果 transcoefficient 为偶数，则 level 等于(RefAbslevel+escape_level_diff)。最后再按照标准中的码表切换规则进行码表切换，等待解析下一个语法元素。

3 测试与验证

将本文提出的熵解码器结构设计用 Verilog HDL 实现，用 ModelSim 在综合前进行仿真，仿真模型如图 5 所示。AVS 标准参考软件产生测试码流数据，将测试码流数据输入硬件描述语言仿真模型，然后将输出数据与标准输出对比，实现仿真功能，经过测试，硬件熵解码后的数据与软件熵解码后的数据完全一致。

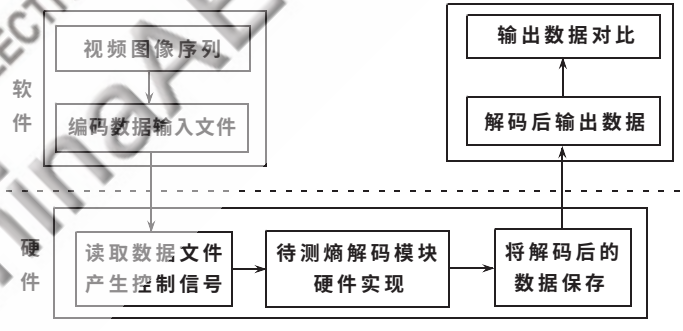

图 5 仿真模型

整个电路使用 Altera 的 QuartusII 9.0 软件进行综合，使用的 LE 总数为 1 906 个。变字长解码器模块是 AVS 解码器的一部分，综合后与其他部分集成，通过了 FPGA 验证，采用 CycloneII 系列的 EP2C35 型 FPGA，其最高频率能达到 94 MHz，可以实现 AVS 高清晰度视频的实时解码。图 6 为 ce(v)解码模块的功能仿真结果图。

本文提出了一种基于 AVS 标准的熵解码器的硬件结构，从总体设计到各个模块的设计都使用了流水线、并行处理、可重用设计等硬件设计的思想和方法；采用 Verilog 硬件描述语言实现，并通过 FPGA 进行了验证，达到了标准清晰度实时解码的要求。

参考文献

[1] 王忠平.AVS 和 H.264 双模解码器 SoC 混成架构的设计与研究[D].上海：上海交通大学，2008.

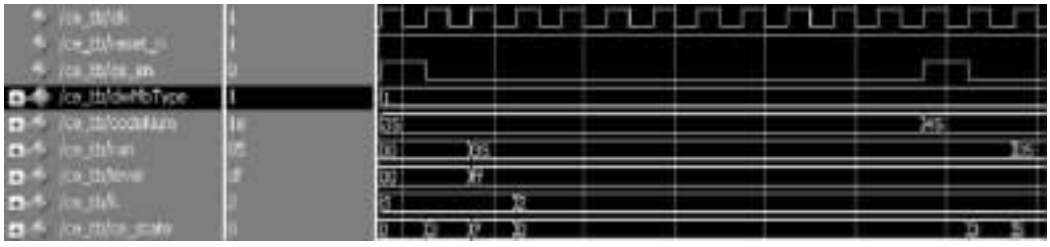

图 6 ce(v)解码模块功能仿真结果

[2] AVS 联合工作组.信息技术:先进音视频编码第 2 部分:视频. 2006.

[3] 徐龙,邓磊.AVS 熵解码器的 VLSI 设计[J].计算机研究与发展,2009.

[4] 张楚.AVS 和 H.264 双标准可变长解码器设计[D].西安:西北工业大学,2007.

(收稿日期:2010-08-10)

作者简介:

赵龙辉,男,1986 年生,硕士研究生,主要研究方向:嵌入式系统及应用。

陈新华,女,1950 年生,教授,硕士生导师,主要研究方向:嵌入式系统的软硬件设计,集成电路及其系统设计,智能信息家电,微型计算机控制与应用。

任怀鲁,男,1986 年生,硕士研究生,主要研究方向:专用集成电路设计。