

【H.264/AVC视频编解码技术详解】六、H.264的NAL解析_测试_Workshop of Wenjie.Yin-CSDN博客

 blog.csdn.net/shagoneal/article/details/52166777

《H.264/AVC视频编解码技术详解》视频教程已经在“CSDN学院”上线，视频中详述了H.264的背景、标准协议和实现，并通过一个实战工程的形式对H.264的标准进行解析和实现，欢迎观看！

“纸上得来终觉浅，绝知此事要躬行”，只有自己按照标准文档以代码的形式操作一遍，才能对视频压缩编码标准的思想和方法有足够深刻的理解和体会！

链接地址：[H.264/AVC视频编解码技术详解](#)

本节视频免费

GitHub代码地址：[点击这里](#)

H.264的NAL解析

在有序字节流格式的H.264码流中，我们可以根据前缀起始码0x 00 00 01或0x 00 00 01获取到一个完整的NAL Unit所包含的字节数据。H.264码流中的每一个NAL Unit的作用并不是相同的，而是根据不同的类型起不同的作用。因此将H.264的码流解析为NAL Unit之后，下一步将是对NAL Unit中的数据进行解析。

一、NAL Unit结构

H.264标准中规定的一个NAL Unit的结构如下图：

Zero_bit	nal_ref_idc		nal_unit_type				
0	1	1	0	0	1	1	1

一个NAL Unit都是由一个NAL Header和一个NAL Body组成。对于基本版本的H.264标准（不考虑SVC和MVC扩展），一个NAL Header的长度固定为1，即8bit。这8bit的含义分别为：

- forbidden_zero_bit：每一个NAL Header的第一个bit，规定必须为0；
- nal_ref_idc：第2和3位，主要表示NAL的优先级。当该值为正时，表示当前NAL Unit中包含了SPS、PPS和作为参考帧的Slice等重要数据。

- `nal_unit_type`：表示NAL Unit的类型，包括VCL层和非VCL层的多种数据类型。常见的`nal_unit_type`取值有：7表示SPS，8表示PPS，5表示IDR帧，1表示非IDR帧等。

二、NAL Unit的有效负载数据及其封装

在NAL Header之后，NAL Unit的其余部分，即NAL Body包含了有效负载数据的封装。从NAL Body到实际的语法元素的码流共3层封装：

(1). 第一层：EBSP——扩展字节序列载荷

EBSP全称为Extended Byte String Payload，等同于NAL Body的数据本身。在EBSP中包含了一个特殊的字节0x03，表示防止竞争校验字节：

`emulation_prevention_three_byte`：设置该值的目的是为了防止NAL Body内部出现于NAL Unit起始码0x 00 00 01或0x 00 00 00 01冲突。

当内部的连续4字节数据出现了下列情况时：

- 0x 00 00 00
- 0x 00 00 01
- 0x 00 00 02
- 0x 00 00 03

在两个0字节之后会插入值为3的一个字节，形成下列情况：

- 0x 00 00 03 00
- 0x 00 00 03 01
- 0x 00 00 03 02
- 0x 00 00 03 03

在进行解析时需要将附加的03字节去掉，得到RBSP数据。

(2). 第二层：RBSP——原始字节序列载荷

RBSP全称为Raw Byte Sequence Payload，相当于NAL Body去掉`emulation_prevention_three_byte`之后的数据，是对原始的语法元素码流进一步处理后产生的数据。相比于原始的语法元素码流，RBSP在末尾添加了`rbsp_trailing_bits()`部分，其主要目的是字节对齐。每个`rbsp_trailing_bits()`包括一个1bit和若干个0bit，0bit的个数不定，以实现字节的对齐。

(3). 第三层：SODB——数据字节流

SODB全称为String Of Data Bits，表示H.264的语法元素编码完成后的实际的原始二进制码流。SODB通常不能保证字节对其。