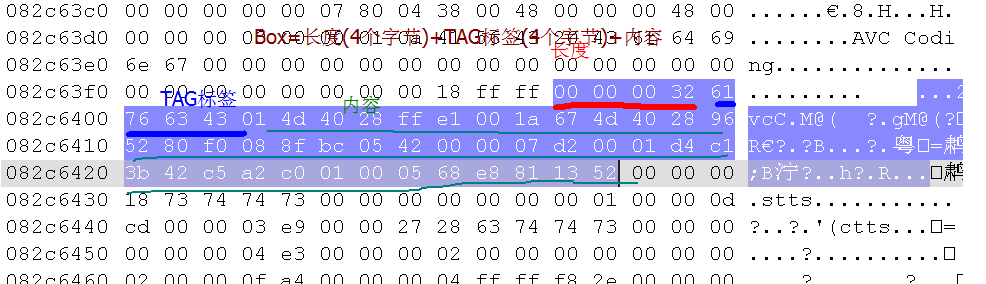
## http://blog.csdn.net/chenchong\_219/article/details/44263691

http://blog.csdn.net/lxmnet123/article/details/10729463

## MP4封装格式结构图 MP4中box存储方式为大端模式(打开文件 直接看就可以)

<http://blog.csdn.net/sunnylgz/article/details/7680203>

Box = 长度(4个字节,包含长度本身)+TAG标签(4个字节)+内容



/\* **1080i 隔行扫描的 获取的时间戳有点特别 尚未知道原因 解码之后 奇偶场会合并成一帧 容器里面会提示先发奇还是偶场 而且会有原始帧率和帧率两个概念**

**1080p 逐行扫描的 时间戳是按照帧率倒数隔开的 比如30fps: 33333 66666 100000**

D/streamer\_jni(14148): Extractor 0

D/streamer\_jni(14148): Extractor 16683

D/streamer\_jni(14148): Extractor 100100

D/streamer\_jni(14148): Extractor 116783

D/streamer\_jni(14148): Extractor 33366

D/streamer\_jni(14148): Extractor 50050

D/streamer\_jni(14148): Extractor 66733

D/streamer\_jni(14148): Extractor 83416

D/streamer\_jni(14148): Extractor 200200

D/streamer\_jni(14148): Extractor 216883

D/streamer\_jni(14148): Extractor 133466

D/streamer\_jni(14148): Extractor 150150

D/streamer\_jni(14148): Extractor 166833

D/streamer\_jni(14148): Extractor 183516

D/streamer\_jni(14148): Extractor 300300

D/streamer\_jni(14148): Extractor 316983

D/streamer\_jni(14148): Extractor 233566

D/streamer\_jni(14148): Extractor 250250

D/streamer\_jni(14148): Extractor 266933

D/streamer\_jni(14148): Extractor 283616

\* \*/

用mediainfo查看mp4经常会看到有两个帧率信息，一个是framerate，一个是original framerate，前者是文件的实际帧率，后者是h264解码信息中的帧率。

计算实际帧率时，需要从mp4的moov box中的mvhd中取得timescale和duration，将两者相除得到文件总时长，

再从stsz box中取得sample总数，即总帧数sample\_count，用总帧数除以总时长即平均帧率，计算公式为：

frame\_rate = sample\_count/(duration/timescale)

h264解码信息中的帧率位于sps中的vui\_parameters内，计算公式为

frame\_rate = sps.vui\_parameters.num\_units\_in\_tick.time\_scale / sps.vui\_parameters.num\_units\_in\_tick

参考链接：

MP4文件格式详解——元数据moov（一）mvhd box：http://blog.csdn.net/pirateleo/article/details/7590056/

mp4文件格式系列26--- Sample Size Atoms - STSZ：http://www.360doc.com/content/13/1119/17/6979751\_330551009.shtml

播放器适配经验总结——IOS：http://blog.csdn.net/luansxx/article/details/7721282

Android libstagefright库中的MPEG4Extractor.cpp 获得 sample数目

<http://blog.csdn.net/u011576407/article/details/51765190>

# [ffmpeg 从mp4上提取H264的nalu](http://blog.csdn.net/gavinr/article/details/7183499) <http://blog.csdn.net/gavinr/article/details/7183499>

H264码流

00 00 00 01是Start code

forbidden\_zero\_bit 是禁止位，应该是第一位即f(1)=0，1为语法有错误

nal\_ref\_idc是参考级别，代表被其它帧参考情况，u(2)= 11 = 3最大（0为无参考，详见规范)

nal\_unit\_type是该帧的类型，为剩下的5位，u(5)= 0 0111 = 7

profile\_idc的u(8)

66 Baseline

77 Main

88 Extended

100 High (FRExt)

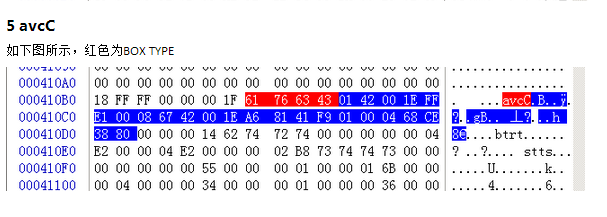
110 High 10 (FRExt)

122 High 4:2:2 (FRExt)

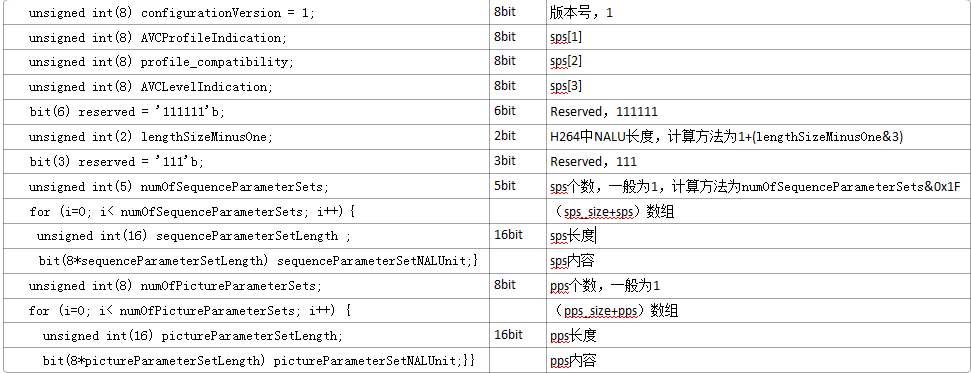
144 High 4:4:4 (FRExt)

100是High (FRExt)

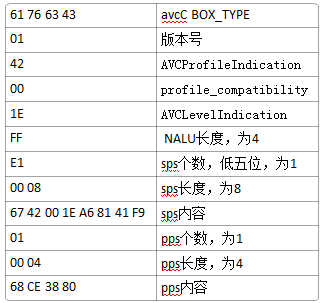
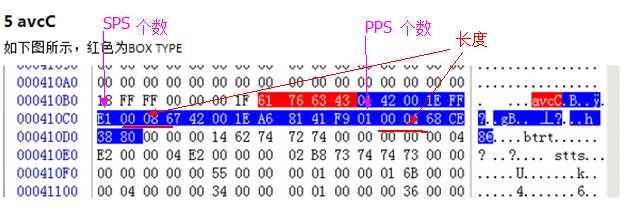
level\_idc则是0D是13，seq\_parameter\_set\_id的ue(v)，则指到AC了，这是哥伦布编码



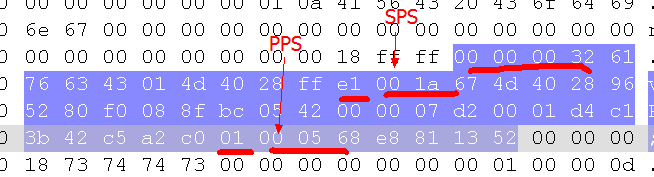
关于AVCDecoderConfigurationRecord结构定义为



4  【实例分析】 数据如上avcC图所示，现在对数据进行详细分析

（NALU长度=0xFF&3+1 = 4个字节）

所以，提取的SPS和PPS分别为67 42 00 1E A6 81 41 F9和68 CE 38 80

 注意mp4info没有显示完，都少了8个字节==>