

# 一、MP4 格式基本概念

MP4 格式对应标准 MPEG-4 标准 (ISO/IEC14496)

## 二、MP4 封装格式核心概念

1 MP4 封装格式对应标准为 ISO/IEC 14496-12 (信息技术 视听对象编码的第 12 部分: ISO 基本媒体文件格式/Information technology Coding of audio-visual objects Part 12: ISO base media file format)

附一 标准免费下载: Freely Available  
Standards <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>

2 MP4 封装格式是基于 QuickTime 容器格式定义, 媒体描述与媒体数据分开, 目前被广泛应用于封装 h. 264 视频和 ACC 音频, 是高清视频/HDV 的代表。

3 MP4 文件中所有数据都封装在 box 中 (对应 QuickTime 中的 atom), 即 MP4 文件是由若干个 box 组成, 每个 box 有长度和类型, 每个 box 中还可以包含另外的子 box (称 container box)。

一个 MP4 文件首先会有且只有一个 “ftyp” 类型的 box, 作为 MP4 格式的标记并包含关于文件的一些信息; 之后会有且只有一个 “moov” 类型的 box (Movie Box), 它是一种 container box, 子 box 包含了媒体的 metadata 信息; MP4 文件的媒体数据包含在 “mdat” 类型的 box (Media Data Box) 中, 该类型的 box 也是 container box, 可以有多个, 也可以没有 (当媒体数据全部引用其他文件时), 媒体数据的结构由 metadata 进行描述。

4 MP4 中 box 存储方式为大端模式。一般, 标准的 box 开头会有四个字节的 box size。

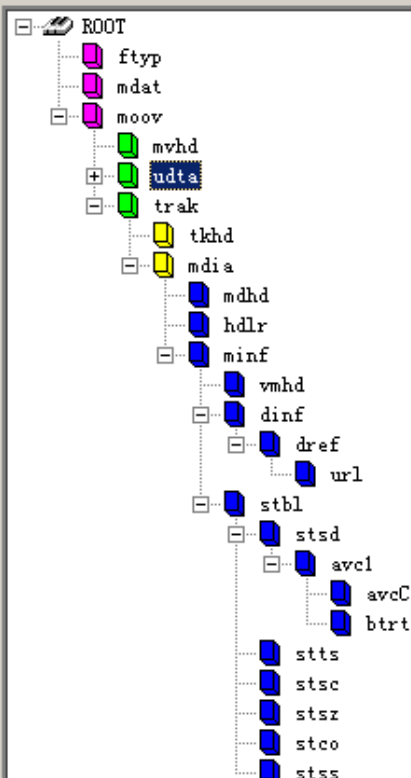
### 5 几个名词

<i>track</i>	表示一些 sample 的集合, 对于媒体数据来说, track 表示一个视频或音频序列。
<i>hint track</i>	特殊的 track, 并不包含媒体数据, 包含的是一些将其他数据 track 打包成流媒体的指示信息。
<i>sample</i>	对于非 hint track 来说, video sample 即为一帧视频, 或一组连续视频帧, audio sample 即为一组连续的压缩音频, 它们统称 sample。 对于 hint track, sample 定义一个或多个流媒体包的格式。
<i>sample table</i>	指明 sample 时序和物理布局的表。
<i>chunk</i>	一个 track 的几个 sample 组成的单元。

## 三、MP4 封装格式结构图

### 1 实例样本

来源于 Android MediaRecoder 视频录制, 平台为华为 T8300 和 TCL968, 用 [mp4info](#) 查看如下:



mp4info mp4tool

audio: (null)  
 smprate: 0  
 channel: -1  
 bitrate: 0

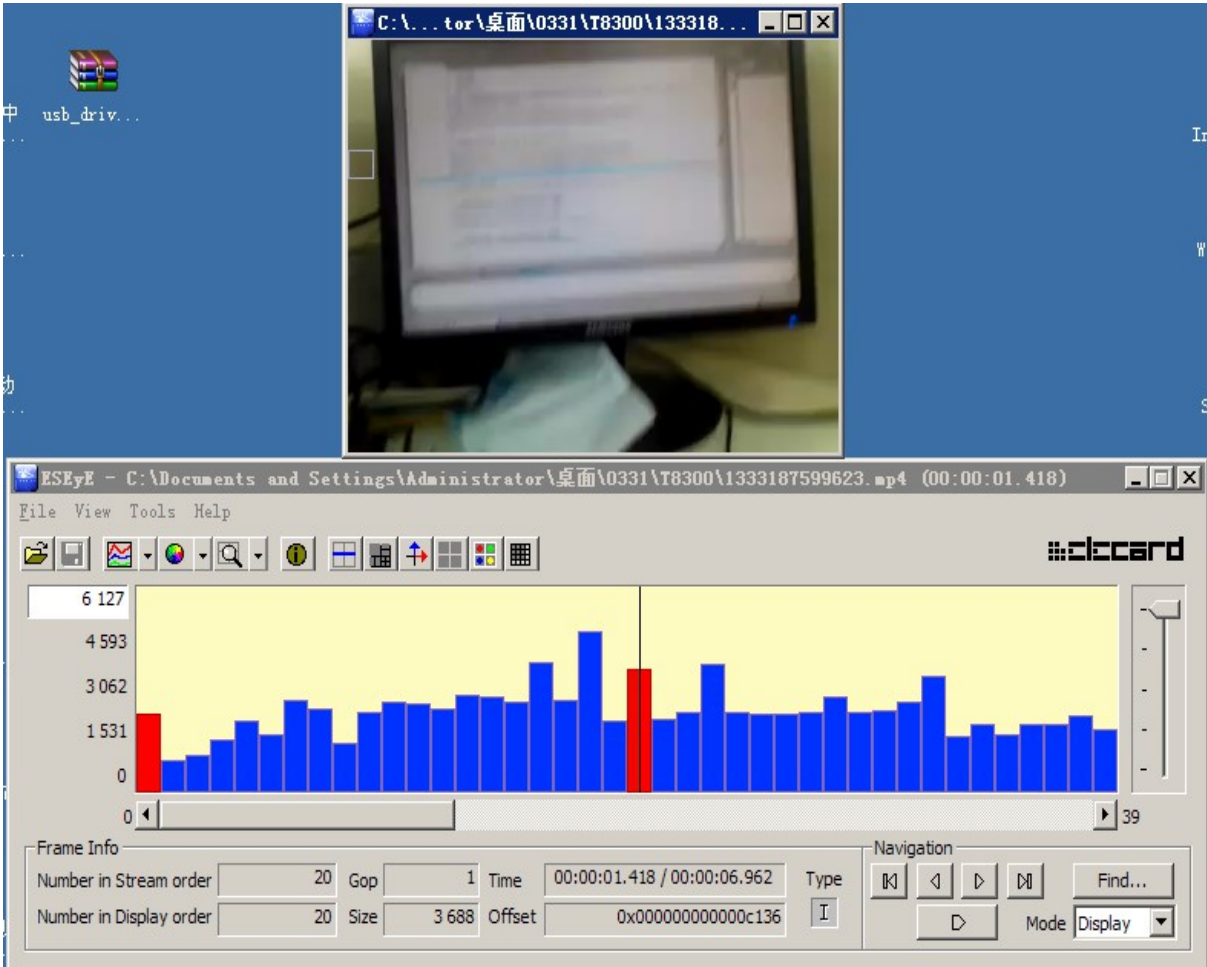
total\_time: 6963--[7000]  
 duration: 6963  
 timescale: 1000  
 maxaudiosize: 0  
 audiosamplenun: 0  
 maxvideosize: 6127  
 videosamplenun: 127

video: avc1  
 width: 320  
 height: 240  
 bitrate: 305223  
 frames: 127  
 fps: 18.239265

```

00 00 00 FD 75 64 74 61 00 00 00 12 63 70 72 74 ....udta....cp
00 00 00 00 00 00 FE FF 00 00 00 00 12 61 75 .....au
74 68 00 00 00 00 00 00 FE FF 00 00 00 00 00 12 th.....
74 69 74 6C 00 00 00 00 00 00 FE FF 00 00 00 00 titl.....
00 12 64 73 63 70 00 00 00 00 00 00 FE FF 00 00 ..dscp.....
00 00 00 12 70 65 72 66 00 00 00 00 00 00 FE FF ....perf.....
00 00 00 00 00 12 67 6E 72 65 00 00 00 00 00 00 .....gnre.....
FE FF 00 00 00 00 00 1A 72 74 6E 67 00 00 00 00 .....rtng....
00 00 00 00 00 00 00 00 00 00 FE FF 00 00 00 00 .....
00 18 63 6C 73 66 00 00 00 00 00 00 00 00 00 00 ..clsf.....
00 00 FE FF 00 00 00 00 00 0F 6B 79 77 64 00 00 .....kywd..
00 00 55 C4 00 00 00 00 27 6C 6F 63 69 00 00 00 .....loci...
00 00 00 FE FF 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 FE FF 00 00 00 00 FE FF 00 00 00 00 13 .....
61 6C 62 6D 00 00 00 00 00 00 FE FF 00 00 01 00 albm.....
00 00 0E 79 72 .....yr
  
```

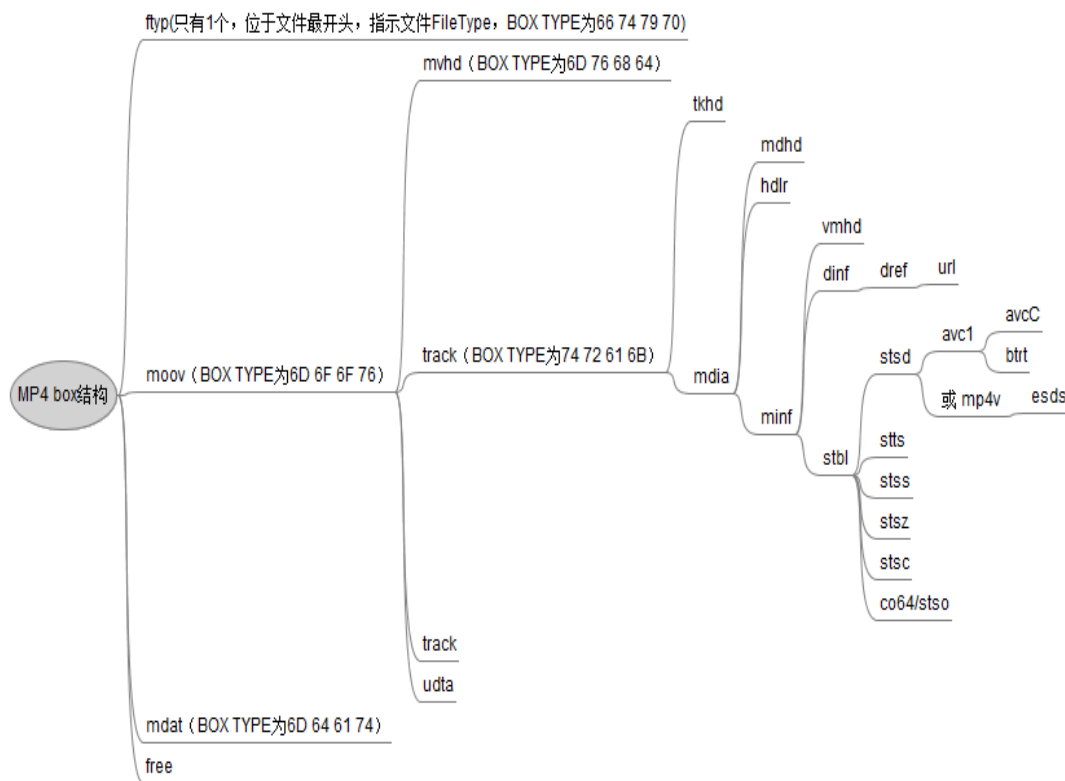
用 EsEYE 查看如下：



用 winhex 分析如下：

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00040DC0	1A	E8	10	A8	2F	51	70	9B	0A	D9	E0	20	B9	65	FA	C3	.. /Qp . ? . 質
00040DD0	E4	D1	93	37	E7	AE	5D	D7	76	35	E0	00	00	08	6B	6D	搜?. 纒] v5?. ... k
00040DE0	6F	6F	76	00	00	00	6C	6D	76	68	64	00	00	00	00	CB	ov...l mvhd.... È.
00040DF0	9C	84	90	CB	9C	84	90	00	00	03	E8	00	00	1B	33	00	漢愼漢?...?...3.
00040E00	01	00	00	01	00	00	00	00	00	00	00	00	00	00	00	00	.....
00040E10	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00040E20	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	40	.....@
00040E30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00040E40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	02	00	.....
00040E50	00	00	FD	75	64	74	61	00	00	00	12	63	70	72	74	00	.. 靚dta.... cppt.
00040E60	00	00	00	00	00	FE	FF	00	00	00	00	12	61	75	74	00	..... aut
00040E70	68	00	00	00	00	00	00	FE	FF	00	00	00	00	12	74	00	h..... t.
00040E80	69	74	6C	00	00	00	00	00	00	FE	FF	00	00	00	00	00	itl.....
00040E90	12	64	73	63	70	00	00	00	00	00	00	FE	FF	00	00	00	. dscp.....
00040EA0	00	00	12	70	65	72	66	00	00	00	00	00	FE	FF	00	00	...perf.....
00040EB0	00	00	00	00	12	67	6E	72	65	00	00	00	00	00	00	FE	..... gnre.....
00040EC0	FF	00	00	00	00	00	1A	72	74	6E	67	00	00	00	00	00	..... rtng.....
00040ED0	00	00	00	00	00	00	00	00	00	FE	FF	00	00	00	00	00	.....
00040EE0	18	63	6C	73	66	00	00	00	00	00	00	00	00	00	00	00	. clsf.....
00040EF0	00	FE	FF	00	00	00	00	00	0F	6B	79	77	64	00	00	00	. .... kywd...
00040F00	00	55	C4	00	00	00	00	27	6C	6F	63	69	00	00	00	00	. U? ...' loci....
00040F10	00	00	FE	FF	00	00	00	00	00	00	00	00	00	00	00	00	..? .....
00040F20	00	00	00	FE	FF	00	00	FE	FF	00	00	00	00	00	13	61	... .. a
00040F30	6C	62	6D	00	00	00	00	00	00	FE	FF	00	00	01	00	00	lbm.....
00040F40	00	0E	79	72	72	63	00	00	00	00	00	00	00	00	06	FA	.. yrrc.....
00040F50	74	72	61	6B	00	00	00	5C	74	6B	68	64	00	00	00	01	trak... \tkhd...
00040F60	CB	9C	84	90	CB	9C	84	90	00	00	00	01	00	00	00	00	蘊劍蘊劍.....
00040F70	00	00	1B	33	00	00	00	00	00	00	00	00	00	00	00	00	...3.....
00040F80	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00	.....
00040F90	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00	.....
00040FA0	00	00	00	00	40	00	00	00	01	40	00	00	00	F0	00	00	...@....@...
00040FB0	00	00	06	96	6D	64	69	61	00	00	00	20	6D	64	68	64	... mdia... mdhd
00040FC0	00	00	00	00	CB	9C	84	90	CB	9C	84	90	00	00	03	E8	... 蘊劍蘊劍...
00040FD0	00	00	1B	33	00	00	00	00	00	00	00	2D	68	64	6C	72	...3.....-hdlr
00040FE0	00	00	00	00	00	00	00	00	76	69	64	65	00	00	00	00	..... vide...
00040FF0	00	00	00	00	00	00	00	00	56	69	64	65	6F	48	61	6E	..... VideoHan
00041000	64	6C	65	72	00	00	00	06	41	6D	69	6E	66	00	00	00	dler.... Aminf...
00041010	14	76	6D	68	64	00	00	00	01	00	00	00	00	00	00	00	. vmhd.....

## 2 box 结构图



接下来对 h264 编码中有用的几个进行阐述，其它不再描述。

### 3 ftyp (file type box)

如下图所示，开始的四字节 00 00 00 00 18 表示该 box 的 size 为 24 字节（含头），然后 66 74 79 70 是 ftyp 的 BOX TYPE，其它是一些格式兼容等相关信息。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	00	00	00	18	66	74	79	70	33	67	70	34	00	00	03	00	... ftyp 3gp4 ...
00000010	33	67	70	34	33	67	70	36	00	04	0D	C3	6D	64	61	74	3gp4 3gp6 ... mdat
00000020	00	00	09	39	65	88	82	04	83	18	13	80	44	50	34	D7	... 9e ? ? . € DP4
00000030	E0	80	C3	38	0C	2A	7F	72	C0	E2	87	8A	E2	B3	FE	FF	鄧? .* r 棱 噴 獬?
00000040	09	A8	4F	AD	27	9F	A7	FE	D3	5D	09	A8	99	07	FF	A5	. 0 ' 觀. ?
00000050	A4	93	47	A8	2B	1C	B7	FE	6E	DE	D9	74	7A	9C	8F	FF	G +. 服 n 賢 z ?
00000060	6F	6C	FA	3D	41	B6	D7	FF	FB	43	D4	16	54	B7	FF	4F	o 1 ? A ? 處 ? T 0

### 4 mdat

如下图所示，BOX YPE 为 6D 64 61 74，紧接着的 00 00 09 39 表示 slice 长度

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	00	00	00	18	66	74	79	70	33	67	70	34	00	00	03	00	....ftyp3gp4....
00000010	33	67	70	34	33	67	70	36	00	04	0D	C3	6D	64	61	74	3gp43gp6...mdat
00000020	00	00	09	39	65	88	82	04	83	18	13	80	44	50	34	D7	...9e???.€DP4
00000030	E0	80	C3	38	0C	2A	7F	72	C0	E2	87	8A	E2	B3	FE	FF	鄧?.*r梭噴獬?
00000040	09	A8	4F	AD	27	9F	A7	FE	D3	5D	09	A8	99	07	FF	A5	.O'靚.?
00000050	A4	93	47	A8	2B	1C	B7	FE	6E	DE	D9	74	7A	9C	8F	FF	G+.服n賢z?
00000060	6F	6C	FA	3D	41	B6	D7	FF	FB	43	D4	16	54	B7	FF	4F	ol?A?鷓?T O
00000070	4D	31	EA	17	AF	FD	B4	89	AD	B6	CD	66	8F	50	8D	54	M1?磯於蚰廝島
00000080	7F	DB	DB	6F	6A	1B	0C	A1	EA	11	A5	47	FD	3D	34	F1	踐j..??4
00000090	EA	27	4F	ED	FF	84	D4	2D	F7	FA	69	E9	A7	FA	D4	77	?O?鷓i詞
000000A0	D1	18	0E	06	81	F4	EF	EB	F6	F8	B8	F5	01	04	F9	F6	?..倖橙蝶铭..
000000B0	7F	FD	73	A8	AF	A1	4C	02	E4	17	80	54	16	7C	43	08	s吠L.?€T. C.
000000C0	86	14	52	42	58	77	84	F0	3B	90	3A	02	54	FF	EF	F1	?RBXw勸;.:T锤
000000D0	C1	B1	11	11	10	1C	04	70	41	2A	AF	5C	24	E0	3E	43	帘.....pA*独\$>C
000000E0	43	65	F3	C2	7B	05	A2	92	1E	84	9C	1A	00	2C	62	5F	Ce舐{.?.?.b_

## 5 avcC

如下图所示，红色为 BOX TYPE

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
000410A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000410B0	18	FF	FF	00	00	00	1F	61	76	63	43	01	42	00	1E	FF	...avcC.B..y.
000410C0	E1	00	08	67	42	00	1E	A6	81	41	F9	01	00	04	68	CE	?..gB..1?...h
000410D0	38	80	00	00	00	14	62	74	72	74	00	00	00	00	00	04	8e....btrt.....
000410E0	E2	00	00	04	E2	00	00	00	02	B8	73	74	74	73	00	00	?...?....stts...
000410F0	00	00	00	00	00	55	00	00	00	01	00	00	01	6B	00	00	.....U.....k..
00041100	00	04	00	00	00	34	00	00	00	01	00	00	00	36	00	00	.....4.....6..

## 四、MP4 文件中 h264 的 SPS、PPS 获取

1 【参考依据】ISO/IEC 14496-15 ([下载](#))

2 【综述】在 H264 中，SPS 和 PPS 存在于 NALU header 中，而在 MP4 文件中，SPS 和 PPS 存在于 AVCDecoderConfigurationRecord，首先要定位 avcC。

3 【定义】

①参数集：一组很少改变的，为大量 VCL NALU 提供解码信息的数据。

序列参数集 SPS 作用于一系列连续的编码图像，而图像参数集 PPS 作用于编码视频序列中一个或多个独立的图像。

如果解码器没能正确接收到这两个参数集，那么其他 NALU 也是无法解码的。因此它们一般在发送其它 NALU 之前发送，并且使用不同的信道或者更加可靠的传输协议（如 TCP）进行传输，也可以重复传输。

②关于 AVCDecoderConfigurationRecord 结构定义为

unsigned int(8) configurationVersion = 1;	8bit	版本号, 1
unsigned int(8) AVCProfileIndication;	8bit	sps[1]
unsigned int(8) profile_compatibility;	8bit	sps[2]
unsigned int(8) AVCLevelIndication;	8bit	sps[3]
bit(6) reserved = '111111'b;	6bit	Reserved, 111111
unsigned int(2) lengthSizeMinusOne;	2bit	H264中NALU长度, 计算方法为1+(lengthSizeMinusOne&3)
bit(3) reserved = '111'b;	3bit	Reserved, 111
unsigned int(5) numOfSequenceParameterSets;	5bit	sps个数, 一般为1, 计算方法为numOfSequenceParameterSets&0x1F
for (i=0; i< numOfSequenceParameterSets; i++) {		(sps_size+sps) 数组
unsigned int(16) sequenceParameterSetLength ;	16bit	sps长度
bit(8*sequenceParameterSetLength) sequenceParameterSetNALUnit;}		sps内容
unsigned int(8) numOfPictureParameterSets;	8bit	pps个数, 一般为1
for (i=0; i< numOfPictureParameterSets; i++) {		(pps_size+pps) 数组
unsigned int(16) pictureParameterSetLength;	16bit	pps长度
bit(8*pictureParameterSetLength) pictureParameterSetNALUnit;}}		pps内容

4 【实例分析】 数据如上 avcC 图所示, 现在对数据进行详细分析

61 76 63 43	avcC BOX_TYPE
01	版本号
42	AVCProfileIndication
00	profile_compatibility
1E	AVCLevelIndication
FF	NALU长度, 为4
E1	sps个数, 低五位, 为1
00 08	sps长度, 为8
67 42 00 1E A6 81 41 F9	sps内容
01	pps个数, 为1
00 04	pps长度, 为4
68 CE 38 80	pps内容

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

## 五、MP4 文件中的 H264 data /NALU slice

1 【参考】H264 官方文档(下载) + 毕书一新一代视频压缩编码标准(下载)

2 【综述】

- ① 在 MP4 格式文件中, H264 slice 并不是以 00 00 00 01 来作分割, 而是存储在 mdat box 中。
- ② H264 基本码流由一些列的 NALU 组成。原始的 NALU 单元组成: [start code] + [NALU header] + [NALU payload]

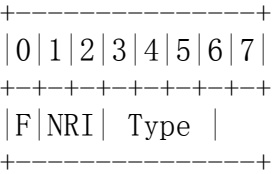
start code	1 字节	00 00 01 或 00 00 00 01	需要添加的
NALU header	1 字节	如下 3	通过 mdat 定位



③ H264 基本码流结构分两层：视频编码层 VCL 和网络适配层 NAL，这样使信号处理和网路传输分离

VCL	负责高效视频内容表示
NAL	以网络所要求的恰当方式对数据进行打包和发送

3 【定义】 NALU header



F	1bit	forbidden_zero_bit，在 H.264 规范中规定了这一位必须为 0
NRI	2bit	nal_ref_idc，0~3，指示这个 NALU 的重要性
Type	5bit	nal_unit_type，NALU 单元的类型 0 没有定义 1-23 NAL单元 单个 NAL 单元包。 24 STAP-A 单一时间的组合包 25 STAP-B 单一时间的组合包 26 MTAP16 多个时间的组合包 27 MTAP24 多个时间的组合包 28 FU-A 分片的单元 29 FU-B 分片的单元 30-31 没有定义

特别的，当值为 7 和 8 分别为 SPS 和 PPS。  
毕书（[下载](#)）（P191）上的定义为：

表 7.20 nal\_unit\_type 语义

nal_unit_type	NAL类型	C
0	未使用	
1	不分区、非 IDR 图像的片	2, 3, 4
2	片分区 A	2
3	片分区 B	3
4	片分区 C	4
5	IDR 图像中的片	2, 3
6	补充增强信息单元（SEI）	5
7	序列参数集	0
8	图像参数集	1
9	分界符	6
10	序列结束	7
11	码流结束	8
12	填充	9
13..23	保留	
24..31	未使用	

nal\_unit\_type=5 时，表示当前 NAL 是 IDR 图像的一个片，在这种情况下，IDR 图像中的每个片的 nal\_unit\_type 都应该等于 5。注意 IDR 图像不能使用片分区。



表 7.20 nal\_unit\_type 语义

nal_unit_type	NAL类型	C
0	未使用	
1	不分区、非 IDR 图像的片	2, 3, 4
2	片分区 A	2
3	片分区 B	3
4	片分区 C	4
5	IDR 图像中的片	2, 3
6	补充增强信息单元 (SEI)	5
7	序列参数集	0
8	图像参数集	1
9	分界符	6
10	序列结束	7
11	码流结束	8
12	填充	9
13..23	保留	
24..31	未使用	

nal\_unit\_type=5 时，表示当前 NAL 是 IDR 图像的一个片，在这种情况下，IDR 图像中的每个片的 nal\_unit\_type 都应该等于 5。注意 IDR 图像不能使用片分区。

#### 4 【实例分析】数据分析，数据如上图 mdat 所示

6D 64 61 74	mdat BOX TYPE
00 00 09 39	silce 长度， 2361

接下来的 65 就是 NALU header，可以由 65&0x1F 来求的后五个 bit，从而得知此 slice 为 I frame  
注意，mdat 与 silce 之间有可能存在若干占位符，我在 TCL 手机测试时就出现了连续的 00 的占位符，这样后面用 H264 硬编码时会比较麻烦一点。

## Ref/Related

- 1 相关资料和工具在文中链接下载
- 2 <http://www.52rd.com/Blog/wqyuwss/559/4/>
- 3 <http://blog.csdn.net/szu030606/article/details/5943279>
- 4 <http://blog.csdn.net/k1988/article/details/5654631>
- 5 <http://www.cppblog.com/czanyou/archive/2008/11/26/67940.html>
- 6 <http://krdai.info/blog/sps-pps-in-mp4-format.html>
- 7 <http://www.cnitblog.com/zouzheng/archive/2007/04/04/25155.html>
- 8 <http://bbs.chinavideo.org/viewthread.php?tid=10273>