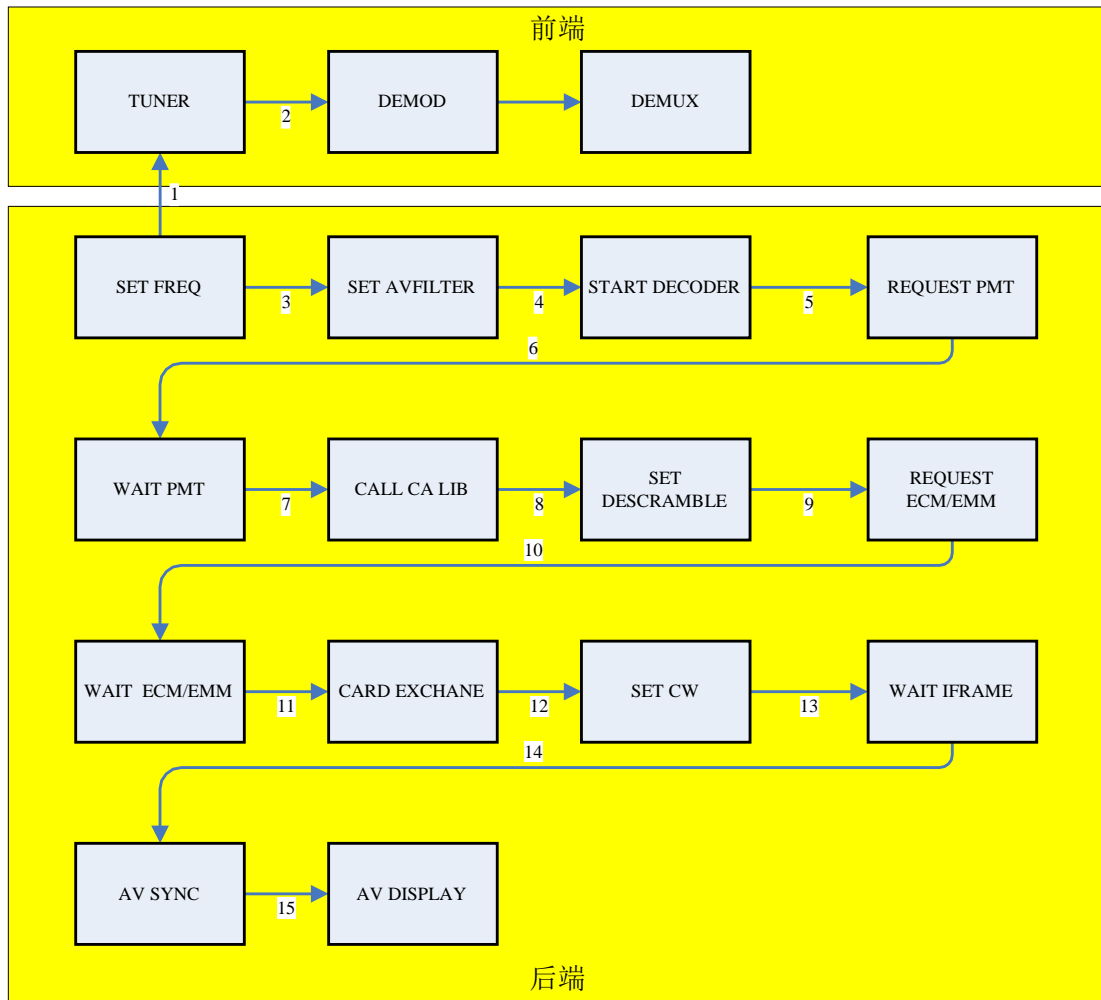


快速换台方案介绍

在机顶盒市场项目中，作为一项最基本的体验，换台速度一直是用户、客户和各位项目成员重点关注的指标。但是在实际开发过程时，很多人不清楚有哪些因素影响换台时间，如何进行提升，下面对此做一些简单的介绍。

如下是加密节目异频点换台过程中，机顶盒需要做的一些主要的动作。另外，还有一些动作没有在流程图中表现出来，例如用户按下换台按键时，实际并不是马上开始切台，而是先绘制 **bannel** 条显示下一个台的节目信息，并静默约 300ms，确认用户不再往下继续按键以后才真正进入换台流程。这样做的目的是为了更快地响应用户的连续切换台动作，避免按键堆积。

除此之外，项目中还会根据具体市场需求，监测 NIT/BAT/PMT/PAT 等等表格，目的是在前端的表格内容更新以后，不必重新搜台，后台自动进行节目信息的更新。换台时，会根据需要对这些表格做启停动作，对换台时间也会有一些负面影响。



结合上图，我们分析一下流程：

流程 1~2：设置频点，一般耗时 100-300ms 左右，卫星的耗时在信号不好的时候最长可能会达到 3~5s，主要原因是当中心频率锁不定的时候还要尝试临近频点能否锁定，以解决频偏的问题。这个耗时一般和流程 3~15 以多线程的方式同步进行。

流程 3~5：设置新节目的 PID 和解码器参数，并请求新节目的 PMT 表格。这个过程一般耗时在 100-300ms 左右；

流程 6：等待 PMT 的到来，主要是 nagra 等 CA 的要求，必须要取到 PMT 表格以后送给 CA LIB 才能继续后续的操作。这个时候就需要等待 TUNER 锁定以后才会取到当前节目的 PMT，从请求到获取到 pmt 一般耗时 50~200ms，对换台速度有一定的影响。因此很多项目中我们都

是先将 PMT 的主要参数存储在数据库中，需要传送 PMT 时通过参数组合成表格格式后送给 CA。

流程 7~9：在大多数 ca 中，这个流程都有 ca 库来控制。耗时约 50~150ms。

流程 10：等待 ecm 的到来。ecm 发送频率很高，一般都在 100ms 左右，在频点已锁定的情况下，从请求到取得一般耗时约 20~100ms；

流程 11~13：在数据都拿到的情况下，这个过程非常短，耗时在 50ms 以内。

流程 14：等待 I 帧的到来。我们知道，由于视频编码的时候采用了基准帧和参考帧，换台时必须拿到第一个参考帧才能正常显示，而一般情况下，I 帧的间隔在 1~3s 的居多，i 帧间隔越长，换台时间也对应的越长。

流程 16：等待 AV 同步。拿到 I 帧以后，还不能马上显示，需要等到同步以后，才能保证视频的流畅，否则会给用户卡顿的感觉，影响体验。

综上，整个换台过程就介绍完了，一般异频点换台的整个时间约 1~3s 之间都属于正常现象。如下是一个典型的同频点换台的数据：

```
-----test result, times=14-----
key_start      : average=[0000]ms: 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,
key_send       : average=[0000]ms: 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,
pat_come       : average=[0000]ms: 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,
request_pmt    : average=[0000]ms: 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,
request_pat    : average=[0000]ms: 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,
start_decoder  : average=[0000]ms: 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,
video_show    : average=[0000]ms: 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,
vid_sync      : average=[0000]ms: 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,
pcr_come       : average=[0000]ms: 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,
StopChannel   : average=[0004]ms: 0003,0003,0004,0002,0004,0003,0003,0002,0003,0007,0002,0002,0003,0016,
set_freq      : average=[0019]ms: 0017,0017,0019,0018,0019,0019,0017,0018,0019,0026,0020,0016,0020,0031,
tuner_lock    : average=[0020]ms: 0018,0017,0020,0019,0019,0020,0018,0018,0019,0027,0020,0016,0020,0032,
set_avfilter  : average=[0028]ms: 0025,0022,0030,0029,0028,0029,0026,0024,0028,0035,0026,0022,0030,0044,
SUB_Stop      : average=[0187]ms: 0160,0249,0161,0185,0149,0156,0226,0189,0177,0163,0169,0201,0204,0208,
pmt_come      : average=[0161]ms: 0172,0107,0190,0142,0247,0181,0164,0115,0102,0211,0197,0152,0157,0126,
set_ecm_filter: average=[0203]ms: 0206,0197,0214,0167,0268,0203,0232,0168,0155,0233,0219,0207,0188,0179,
ecm_come      : average=[0330]ms: 0265,0393,0423,0243,0403,0429,0305,0248,0209,0374,0261,0378,0312,0324,
ECM_TO_LIB    : average=[0340]ms: 0266,0394,0424,0244,0441,0430,0306,0249,0209,0375,0262,0379,0348,0360,
play_end      : average=[0309]ms: 0274,0406,0262,0254,0347,0248,0446,0299,0249,0278,0433,0285,0252,0265,
open_decoder  : average=[0308]ms: 0274,0406,0262,0253,0346,0248,0445,0299,0248,0277,0432,0284,0251,0265,
SC_TRANSFER   : average=[0507]ms: 0461,0214,0616,0436,0633,0625,0507,0441,0427,0567,0464,0572,0540,0552,
set_cw        : average=[0850]ms: 0749,1216,0922,0724,0917,0910,0792,0728,0720,0854,0749,0853,0829,0848,
Iframe_come   : average=[1773]ms: 1282,1786,1196,1770,1756,1542,2081,2254,1428,2171,0923,2409,1641,2094,
```

如下是一个不同频点换台的数据：

ST: Close VBV					
Check point	777<->999	450<->452	419<->787	452<->454	AVG.
set_freq	405	287	249	305	311.5
tuner_lock	663	546	509	563	570.25
request_pat	668	554	521	575	579.5
pat_come	709	594	566	608	619.25
request_pmt	752	609	620	662	660.75
pmt_come	813	645	665	704	706.75
set_ecm_filter	844	732	708	754	759.5
set_avfilter	845	734	709	755	760.75
open_decoder	847	736	707	758	762
ecm_come	907	807	732	804	812.5
set_cw	1279	1174	1100	1171	1181
vid_sync	2766	2955	2456	3137	2828.5

那么，有哪些提高换台速度的方法呢？

1、尽量缩短上述流程中的每个耗时环节所需要的时间，这些手段还算不上是快速换台，只能说是提高换台速度。一般主要的手段是前端优化和后端优化相结合的方法：

1) 前端优化：优化 tuner 和 demod 驱动，加快锁频速度，使数据更快地到达。同时，锁频动作采用多线程的方式进行，省下锁频本身所耗费的时间；

2) 后端优化：主要从这些环节入手

a) 不在换台时现搜 PMT，采用根据数据库里节目参数复原 PMT 的方法，节省 PMT 请求及等待的时间；

b) 在 AV 同步环节，采用慢同步的方式进行。收到 I 帧以后立刻显示，但不按正常速率显示，而是以非常缓慢的慢放方式显示 I 帧，等到真正同步后再切换到正常速率。给用户的感觉就是换台速度明显提高了。目前 MSTAR、ST 等方案上都集成了这种慢放技术。

2、快速换台之前端并行技术。1 中提到的前端优化是有极限的，不管如何优化，总是需要耗费时间。最优效果的方式是采用全频段 tuner+多路解调技术，在用户最常用的按上下键和

返回键换台时，将用户最有可能要换到的节目频点和数据按照一定的优选算法预先给设置下去，当用户真正切到这个频点时，无需再次锁频，直接对 DEMUX 进行关联后就可以拿到数据了。达到和接近和同频点换台同样的换台速度。其关键在于频点预测和预置。

另外，Fullband tuner 在性能上也有所提升。它的工作原理与普通 tuner 不同，是在初始化之后，就对 1GHz 内的模拟频点都进行采样，然后将特征值存储起来，当需要锁定具体频点的时候，直接将对应的中频取出来使用，缩短了锁频的时间。锁频时间从普通 tuner 的 200ms 左右缩短到 50ms 左右。

3、快速换台之后端并行技术。从上面的流程图中看到，主要花费的时间还是在后端，其中等待 I 帧占用了非常长的时间。因此，后端并行技术的主要思路是将可能被换到的节目首先缓存到内存中，并同步进行解扰，记录下每个节目最近的视音频帧数据。当用户真正换台过去以后，立即将存储的 I 帧取出来作为基础帧，同步进行显示，极大的提升了换台速度。目前接触到的有此技术的芯片厂家主要有 mstar、bcm 和 montage，每家的做法又差别较大。

Mstar: 采用的是局部数据缓存技术，当启动换台流程以后，首先将新节目的视频 PID 数据进行缓存，缓存从上图中流程 3 到流程 13 之间的时间段之内的加扰视频数据。当拿到 CW 以后，尝试从缓存的加密数据中解出 I 帧，以达到快速显示的目的。由于 CW 控制字更换周期一般在 10S 左右，有很大的概率能解出正确的 I 帧数据来。此方案不需要上层应用做特殊的操作流程。用户体验一般。

Bcm: 采用的是预解码技术+Fullband tuner 技术。启动换台流程以后，除需要设置当前将要播放的节目外，再设置多个可能被换台的节目参数，机顶盒将对将要换台的节目进行除解码外的所有流程，使节目处于就绪状态。当真正切台后，只需要将预解码的节目关联到解码器即可完成切台。异频点切换速度在 1s 以内，用户体验较好。但此方案需要节目预测，对现有换台流程的框架影响较大。

Montage: 采用的是多路解扰加缓存技术。当用户换到一个新频点以后，就将该频点下的所

有节目都进行解扰，并缓存最新的约 1-3s 的数据。当换台动作发生以后，如果正好是本频点的节目，直接取出对应节目的音视频数据进行显示。目前同频点换台速度在 300ms 左右，异频点换台在 600~800ms 之间，目前仅仅是使用后端并行技术效果已经很不错，未来搭配 2~4 路解调的 tuner 使用效果会更好。此方案对上层框架影响较小。

总之，随着芯片处理性能的提升和 tuner 技术的发展，未来的趋势是前端并行和后端并行技术相结合的发展方向。目前我们可以做的，就是在不增加硬件成本的基础上，通过优化前后端各个耗时环节流程的方法，来达到提升换台速度的目的。

胡雷

2019年 11 月 14 日