# 1 AVS IP 软件验证流程

## 9800验证流程解析

 Firmware启动后阻塞等待；上位机软件读取yuv文件，生成testvector并通过PCIE发送到DDR，然后通知Firmware；Firmware设置寄存器控制AVS3 IP编码；AVS3 IP从DDR上取testvector进行编码，将avs3放到DDR上并通知Firmware；Firmware通知上位机；上位机从DDR取avs3码流进行比对check。

## 1.2 H3 3500版本firmware设计



H3 3500 avs firmware设计思路如下所示：

1. 上位机将YUV存放到DDR指定位置，同时将此视频流对应的cfg配置信息一并作为YUV头信息放到DDR指定位置上；
2. firmware在上位机放置完成后，取出视频流配置信息，生成reg的结果后，启动AVS；
3. AVS将根据reg中配置信息的地址偏移量，从ddr指定位置取yuv，（如果为P帧还需要取上一帧的重构帧）进行处理，firmware将为AVS指定所需的DDR空间如ctu\_ctrl\_input、map\_cu、refp等memory的地址；
4. AVS处理完成后将重构帧和码流写到DDR指定位置（地址由firmware确定）；
5. AVS触发中断后上报，firmware检测到中断，由上位机取AVS输出的数据进行check，check无误后继续下发下一帧YUV file，(如果视频流不改变，且无需改变配置，不需要重新传递给firmware配置信息，firmware会自行计算)；
6. 重复第一步骤，直到一个视频流结束。

firmware内部执行流程如下所示：



## 1.3 目前firmware存在的问题

* ~~无bs data length信息，上位机不知道该取多长，~~
* **加寄存器标识**
* 重构帧里面有padding信息，而所取参考帧不需要padding信息，因此造成了视频流无法连续播放的问题，所以采用1.1所示方法验证IP。
* ~~目前参考帧寄存器只有一个，意味着无法参考多帧，在代码中，参考帧列表是一个大小为17的数组，是否以后重构帧要保留17帧？参考帧寄存器要增加为17个？~~
* **参考是两帧， buf 最大只能有两帧，（可以存相邻或更远的帧）。**
* ~~如果按照重构帧为之后视频流的参考帧这样的做法，重构帧和参考帧长度计算方法不一致。参考帧大小：~~

|  |
| --- |
| size = (ctx->info.pic\_height + 2 \* ctx->refp[0][0].pic->padsize\_luma) \* (ctx->refp[0][0].pic->stride\_luma); |

重构帧大小：

|  |
| --- |
| size = sizeof(pel) \* ctx->pic[1]->stride\_luma \* ctx->pic[1]->height\_luma |

* ~~为什么一个视频流的第一个LCU，ctu\_ctrl\_input\_ipf\_enable\_flag、ctu\_ctrl\_input\_tscpm\_enable\_flag、ctu\_ctrl\_input\_amvr\_enable\_flag、ctu\_ctrl\_input\_emvr\_enable\_flag、ctu\_ctrl\_input\_affine\_enable\_flag一定为0，后续LCU才会根据cfg文件进行配置。~~
* 目前firmware为关闭知识图像下完成，知识图像部分逻辑缺失。
* 当前AVS只经过IPPP帧的测试，firmware也在这种情况下完成，参考帧的逻辑为当前帧只能参考前一帧。LDP 有两个配置。
* 无B帧验证。
* 寄存器存在重复性功能，如fetch\_ctx\_io\_ctx\_ptr、ctu\_ctrl\_input\_ptr均为当前帧计数指针。
* lamda 统一写法
* 1.4 后续firmware所需文件

1. IP spec, 及register说明。
2. 寄存器在不同宏定义下值的来源及变化关系。
3. 测试向量生成方法及相对应版本的代码。
4. cycle 个数要记录下来