**I2 项目Hal UT设计和实现**

**背景：**

对于每个驱动的Hal模块可以做到单独调试。用于做FPGA验证和验收驱动的测试方案；

基本思路：调用任一Hal模块，产生一个输出流，配置控制信息，获取一个输入流。

**实现方式：**

由于Hal是Kernel mode驱动，考虑比较可行的方案是把Hal的UT作为Linux的一个虚拟Module，该Module生成Proc节点。用户通过talent写入命令，并获取信息和文件。



UT test层次图：



**HAL UT基本信息：**

* **覆盖模块**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **VIF** | **VPE** | **VENC** | **VDEC** | **DVIP** | **DSIP** | **AI** | **AO** |

* **HAL UTProc文件路径：**

/proc/Hal/uttest

* **HAL UT Ko文件名：**

haluttest.ko

* **启动方式：**

Insmode /system/modules/haluttest.ko

直接输入挂载驱动或者，配置到init.rc中

* **基本命令格式：**

cat /proc/Hal/uttest

显示ut测试module的基本信息，包括版本，覆盖的模块

**HAL UT调用和参数传递：**

用户命令通过echo方式，把命令和参数传入ut-module的proc节点，EX：

echo vif 2 12 3 5 6 > /proc/hal/uttest

uttest module的基本框架解析命令strings中第一个string，

EX: **vif** 2 12 3 5 6，

这里解析到vif，就调用对应的vif ut模块，然后把第一个空格之后的字符串截取后传递给对应的模块，对应模块提供string的接口。并且解析出argv和argc[];

EX:vif模块拿到“2 12 3 5 6”，这时argv为5，argc[] ={2,12,3,5,6}

第一个参数为testcase，后面为配置参数，有每个模块自己定义

**HAL UT参考代码：**

Linux Module的实现：

<http://blog.csdn.net/a649518776/article/details/7505728>

Proc文件系统解析和实现：

<http://blog.csdn.net/xiaocainiaoshangxiao/article/details/38091459>

http://blog.csdn.net/sty23122555/article/details/51638697