Lab1 设计一个FIR滤波器分离鸟类声音

**目标**

在完成本实验后，您应当学会：

* 如何使用Vitis HLS构建一个项目
* 在Vitis HLS中进行仿真、综合与IP导出

简而言之，您将掌握使用HLS进行加速核设计与部署到PYNQ的基本流程。出于篇幅限制，本实验仅介绍基本工具操作流程。

**环境要求**

* PYNQ-Z2远程实验室服务或物理板卡Vitis HLS
* Vivado

**实验步骤（Linux版本的HLS）**

# 在Vitis\_HLS中设计FIR

## Linux环境设置（如果您使用校内公共服务器平台的话，这些设置都已配好，直接拷贝即可）

1. 设置以下信息，比如bash环境下修改自己根目录下的**.bashrc文件**（路径请根据自己情况修改）

# Xilinx License

export XILINXD\_LICENSE\_FILE=/mnt/home/software/vivadoML21.2/Xilinx2037.lic

# HLS basic settings

source /mnt/home/tools/xilinx/Vitis\_HLS/2021.2/settings64.sh

# PATH setting

export PATH=$PATH:/mnt/home/tools/xilinx/Vitis\_HLS/2021.2/lnx64/tools/clang/bin

# Library settings

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/mnt/home/tools/xilinx/Vitis\_HLS/2021.2/lib/lnx64.o:/mnt/home/tools/xilinx/Vitis\_HLS/2021.2/lib/lnx64.o/Default

1. 测试自己的环境是否已经OK

输入以下命令，看能否找到：

which vitis\_hls

## 代码脚本设置

新建一个目录 mkdir FIR

进入该目录 cd FIR

将目标C文件拷贝到该目录下 cp -r /mnt/home/students/zhaokang/Lab/ ./

用vim新建一个TCL脚本文件，命名为“run.tcl”，将以下内容输入并保存。

# 1. project name

open\_project prj -reset

# 2. set top name

set\_top fir\_wrap

# 3. add files

add\_files fir.cpp

add\_files -tb fir\_tb.cpp

# 4. solution name

open\_solution "solution1" -reset

# 5. device name

set\_part virtex7

# 6. clock setting

create\_clock -period 13 -name default

config\_export -version 2.0.1;

# 7. C-simulation

csim\_design

# 8. synthesis

csynth\_design

# 9. co-simulation

cosim\_design

## 执行HLS，生成RTL硬件描述

在该目录下，执行命令： vitis\_hls run.tcl

则会自动执行c-simulation, synthesis, co-simulation等过程，并生成日志文件vitis\_hls.log，同时输出到屏幕。

## 查看HLS的生成结果

1. 查看LOG日志

vim vitis\_hls.log

你会看到，HLS先执行c-simulation，并输出结果；之后是详细的综合（synthesis）过程，既包括前端编译结果，也包括后端的调度和资源绑定结果；最后是co-simulation结果。

1. 查看生成的RTL文件

cd prj/solution1/syn/

ls

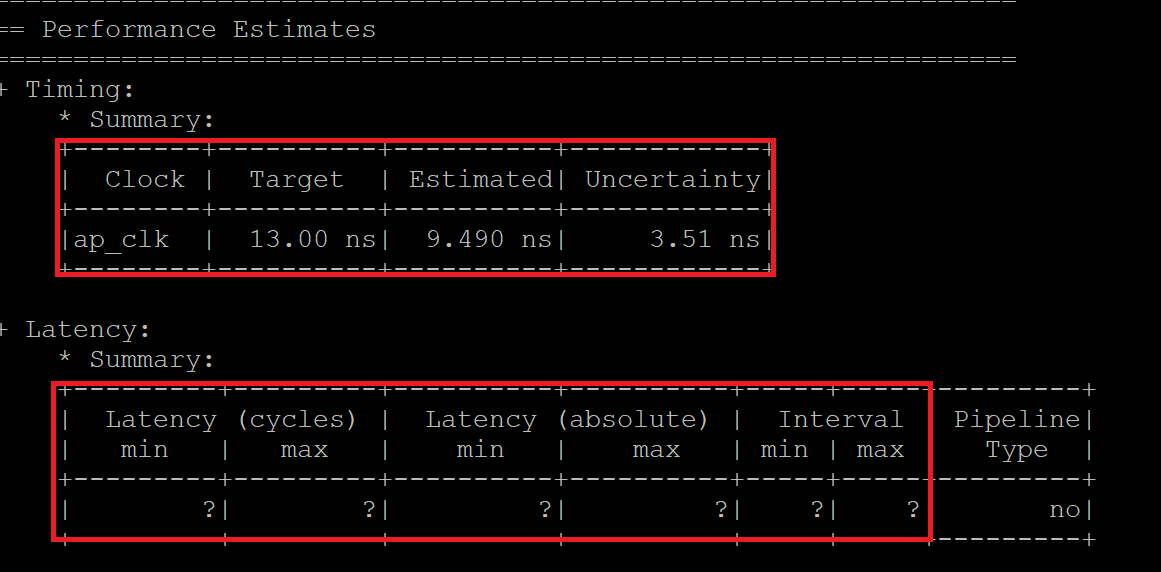
你会看到，除了报告report外，生成了verilog和VHDL两种RTL硬件描述。

1. 查看综合synthesis报告，评估其性能

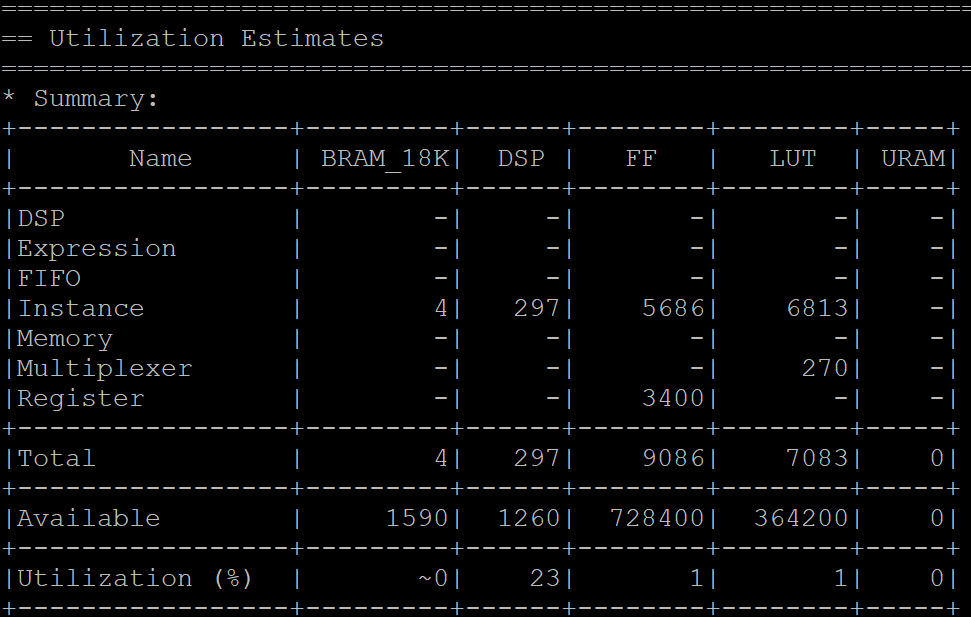
cd prj/solution1/syn/report/

打开顶层module名字对应的rpt报告文件，比如该顶层文件名叫“fir\_wrap”，则打开fir\_wrap\_csynth.rpt： vim fir\_wrap\_csynth.rpt

可以看到其预估的timing值、latency值：



也可以看到预估的各类资源的使用量：



1. 查看更精确的性能latency结果，即co-simulation结果

cd prj/solution1/sim/report/

vim fir\_wrap\_cosim.rpt

可以看到精确的latency值：

