# **FPGA初级设计与开发培训计划**

# 培训目标：

使研究生掌握FPGA设计文档的撰写方式，基本开发流程以及各类开发/仿真/测试工具的使用。从而能够在软硬件协同团队内部独立或协同承担较简单的硬件逻辑设计与开发的任务。

1. 在设计文档撰写中：能够根据项目设计需求，在对应模板指导下，对整体系统逻辑进行迭代式模块分解，并对各模块进行接口与内部状态机设计。另外，能够进行简单的寄存器编址以及读写协议设计。
2. 在FPGA开发流程中：在单时钟域场景下，能够以“一段式”方式使用verilog编写包含状态机的硬件模块，使用FIFO, RAM等常用IP核，并初步理解组合逻辑与时序逻辑之间区别以及满足高频率时序的基本编码原则。
3. 在开发/仿真/测试工具使用上：学习low-level的testbench的编写，学会vivado/modelsim等基于波形图的仿真工具使用，学会基本tcl脚本的使用以及FPGA项目Makefile的编写，另外初步了解myhdl等基于python等高级语言的仿真工具，学习altera/xilinx平台中signal tap/debug core的使用，学习阅读时序报告并且分析关键路径等。

# 知识预备：

1. Verilog 2001中[基本语法](https://www.runoob.com/w3cnote/verilog-basic-syntax.html) (可否使用verilog撰写一个无符号加法器模块)；
2. 有限状态机 (可否用有限状态机表示一个[UDP/TCP报文分类器模块](https://github.com/fast-codesign/FAST-ANT/blob/master/hw%20code/gpp.v))；
3. 计算机网络基本知识。

# 环境预备：

1. Vivado开发环境（2018.1以上）

<https://www.xilinx.com/support/download.html> （需要下载full product, 而非updater）

1. Linux操作系统（18.04以上）
2. Make, gcc等通用编译环境。

# 计划安排：

培训计划为期5天（1周）共分为三个部分：系统设计与设计文档撰写（1天）；FPGA开发流程（2天）；开发/仿真/测试工具使用（2天）。研究生在培训完成后设计并实现一个可在FAST平台上运行的独立模块：

1. 在FAST UM中实现一个能够产生发送UDP 报文的发包器；
2. 在FAST UM中实现一个能够对UDP/TCP报文分别计数的计数器；
3. 也可自选自定义模块。

**Day1: 设计文档撰写**

1. 硬件设计原则（divide and conquer）

参考：[FAST-ANT网络测试仪设计文档](https://github.com/Winters123/antDev/blob/master/doc/0926ANT-DevFile.pdf)

1. 设计文档构成（总体设计，接口设计，子模块设计，寄存器地址等）
2. 设计文档中制表、作图规范：
   1. 模块接口信号列表；
   2. 模块功能流程图；
   3. 模块连接关系图；
   4. 有限状态机。

**Day2: FPGA基本开发流程1**

1. “一段式”有限状态机编写（使用案例编写）；（[一段式vs两段式/三段式](https://blog.csdn.net/Jackiezhang1993/article/details/85045621)）
2. 时序逻辑与组合逻辑区别及对应always语句块编写方式：
   1. Verilog代码层面阻塞、非阻塞时序逻辑区别；([阻塞非阻塞语句区别分析](https://www.cnblogs.com/xd-elegant/p/3982274.html))
   2. 布局布线后reg变量所生成的器件区别（锁存器，触发器）；
   3. always语句块编写方式（使用案例编写）。

**Day3: FPGA开发流程2**

1. 同步FIFO，RAM等IP核的调用方式（GUI方式与tcl脚本方式）以及时序特点：
   1. 练习1：阅读FAST的UM的[Data Cache](https://github.com/fast-codesign/FAST-ANT/blob/master/hw%20code/data_cache.v)，理解FIFO使用场景；
   2. 练习2：阅读FAST-ANT的[发包模块](https://github.com/fast-codesign/FAST-ANT/blob/master/hw%20code/pgm.v)，理解RAM使用场景；
   3. 练习3：使用GUI/tcl两种方式在vivado/quartus中生成FIFO/RAM的IP核。
2. 介绍AXI-Stream, AXI-Lite, AXI等几种常用数据传输协议接口信号含义以及使用方式
   1. [AXI-Lite时序介绍](https://www.realdigital.org/doc/a9fee931f7a172423e1ba73f66ca4081)
   2. [AXI-Stream时序介绍](https://blog.csdn.net/better_xiaoxuan/article/details/74000480)
   3. [AXI Specification](https://developer.arm.com/documentation/ihi0022/latest)

**Day4: 相关工具使用1**

1. 学习testbench模块的编写与测试：（使用案例编写）
   1. 学习testbench激励模块的编写方法；（[一种示例](https://blog.csdn.net/qq_26652069/article/details/96422293)）
   2. 学习在vivado simulation/modelsim中使用testbench对模块进行时序仿真；
2. 了解[myhdl](https://github.com/myhdl/myhdl)等基于python等高层次语言的非波形图仿真框架（对提升仿真效率有帮助）

**Day5: 相关工具使用2**

1. 学习使用vivado/quartus对仿真后的工程代码进行综合并阅读报告：
   1. 学习如何使用vivado/quartus进行综合与布局布线；
   2. 学习如何阅读时序报告并在时序不满足时定位关键路径：
      1. 分辨跨时钟域与单时钟域时序违例的区别；
      2. 分辨走线延迟与逻辑延迟的区别；
      3. 学习缓解时序违例的代码撰写风格/原则。
   3. 学习如何使用debug core/signal tap对特定信号进行debug。

**参考资料：**

1. [FAST 2.0平台原理与应用](https://github.com/fast-codesign/paper-base/blob/master/FAST%E5%B9%B3%E5%8F%B0%E5%8E%9F%E7%90%86%E4%B8%8E%E5%BA%94%E7%94%A82.0.pdf)
2. [FAST UA编程入门手册](https://github.com/fast-codesign/paper-base/blob/master/FAST%E5%B9%B3%E5%8F%B0%E5%8E%9F%E7%90%86%E4%B8%8E%E5%BA%94%E7%94%A82.0.pdf)
3. [FAST论文, IWQoS 2019](https://github.com/fast-codesign/paper-base/blob/master/FAST-final.pdf)