

FPGA 快速系统原型设计 中的敏捷实践



by

《FPGA 快速系统原型设计权威指南》

译者之一：杨碧波 riple

Rapid System Prototyping with FPGAs is definitely a book that should be on any FPGA designer's or manager's shelf. Of course, it will likely be open on the desk for quite a while first.

《FPGA 快速系统原型设计权威指南》

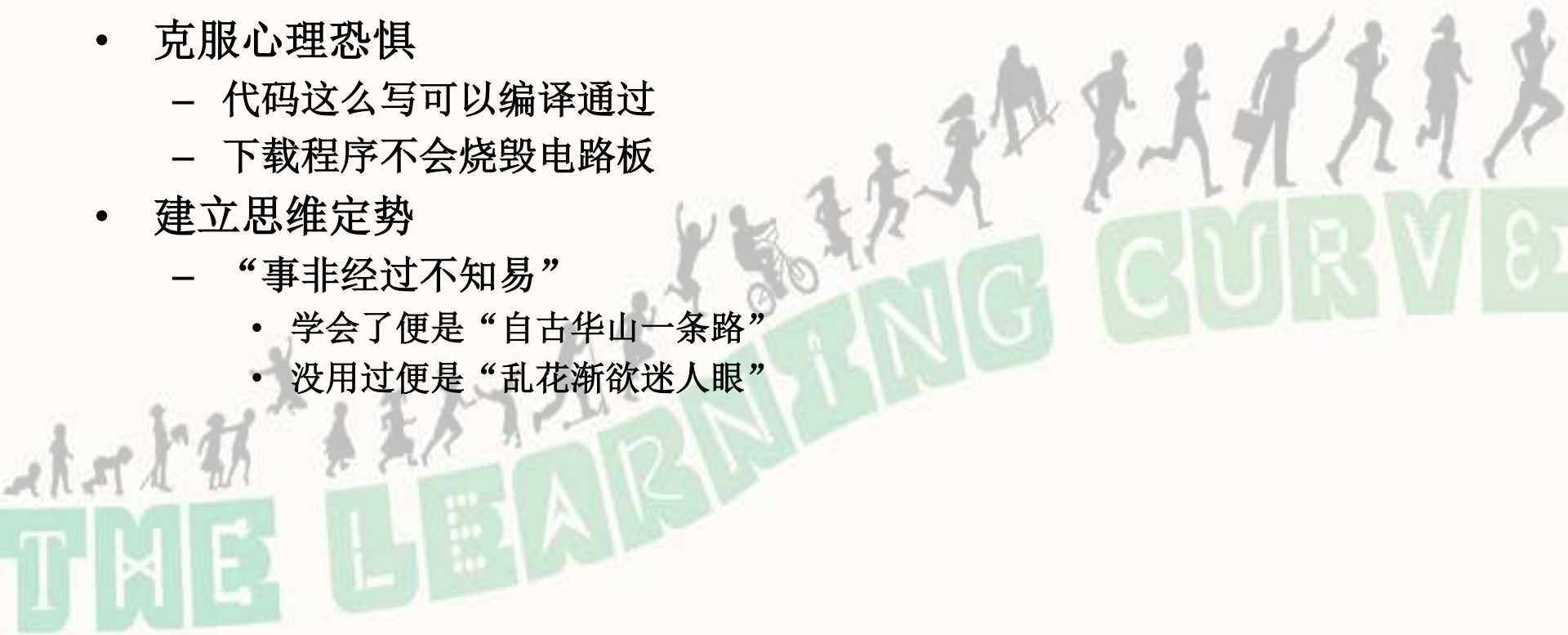
一本常读常新的书

Agenda

- 第一部分：FPGA 工程师的成长
- 第二部分：FPGA开发中的敏捷实践

第一阶段：入门 -> 熟练

- 克服心理恐惧
 - 代码这么写可以编译通过
 - 下载程序不会烧毁电路板
- 建立思维定势
 - “事非经过不知易”
 - 学会了便是“自古华山一条路”
 - 没用过便是“乱花渐欲迷人眼”

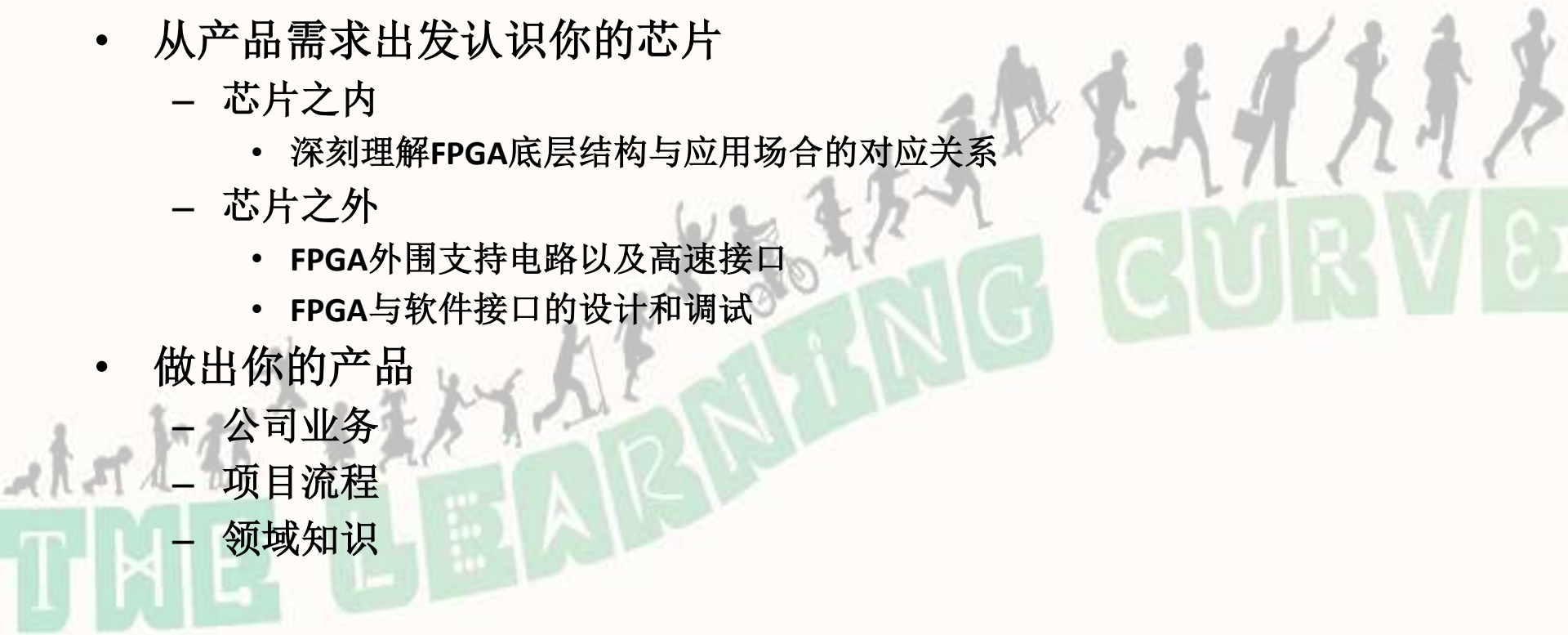


第二阶段：提高 -> 精通

- **EDA工具——从感性认识到理性认识**
 - 站在EDA工具的角度理解FPGA开发流程
 - EDA工具各个流程的意义，各个阶段的输入输出
 - EDA工具输出的报告：Infor和warning
- **HDL语言——积累设计模式**
 - 跨时钟域处理电路
 - 有限状态机电路
 - 特定领域的通路电路
 - 片上总线
- **FPGA芯片——从逻辑特性到物理特性**
 - 搞定时序分析和约束
 - 看懂RTL视图和Technology视图

第三阶段：从业 -> 专业

- 从产品需求出发认识你的芯片
 - 芯片之内
 - 深刻理解FPGA底层结构与应用场合的对应关系
 - 芯片之外
 - FPGA外围支持电路以及高速接口
 - FPGA与软件接口的设计和调试
- 做出你的产品
 - 公司业务
 - 项目流程
 - 领域知识



本书的定位

- 本书的定位不是导航，而是俯瞰
 - “天下事，仰而跂之则难，俯而就之甚易”
 - 读书是作者与读者交流的过程
 - HDL语法书、EDA工具书，越读越薄
 - 开发方法书，启发读者的思路，引发读者的共鸣，举一反三，常读常新
 - 一本好书，恰如冰山一角，能不能看到水面以下的部分，是对读者的考验和挑战
 - 从“精通”到“专业”
- 本书中没有说到的一点
 - 技术+领域=领域专家



个人的成长+成功的项目

- 给刚刚从业的**FPGA**工程师的几点忠告
 - 珍视你的第一份工作
 - 从工作中学习，在工作中成长
 - 记工作日志，写博客
 - **FPGA**工程师学点软件，学点“不可综合语句”
 - 数据的分析和可视化
 - 工具之间的“胶水”
 - 使用**Linux OS**
 - 脚本
 - 免费工具
 - 开源精神

Agenda

- 第一部分：FPGA 工程师的成长
- 第二部分：FPGA开发中的敏捷实践

Agile for FPGA

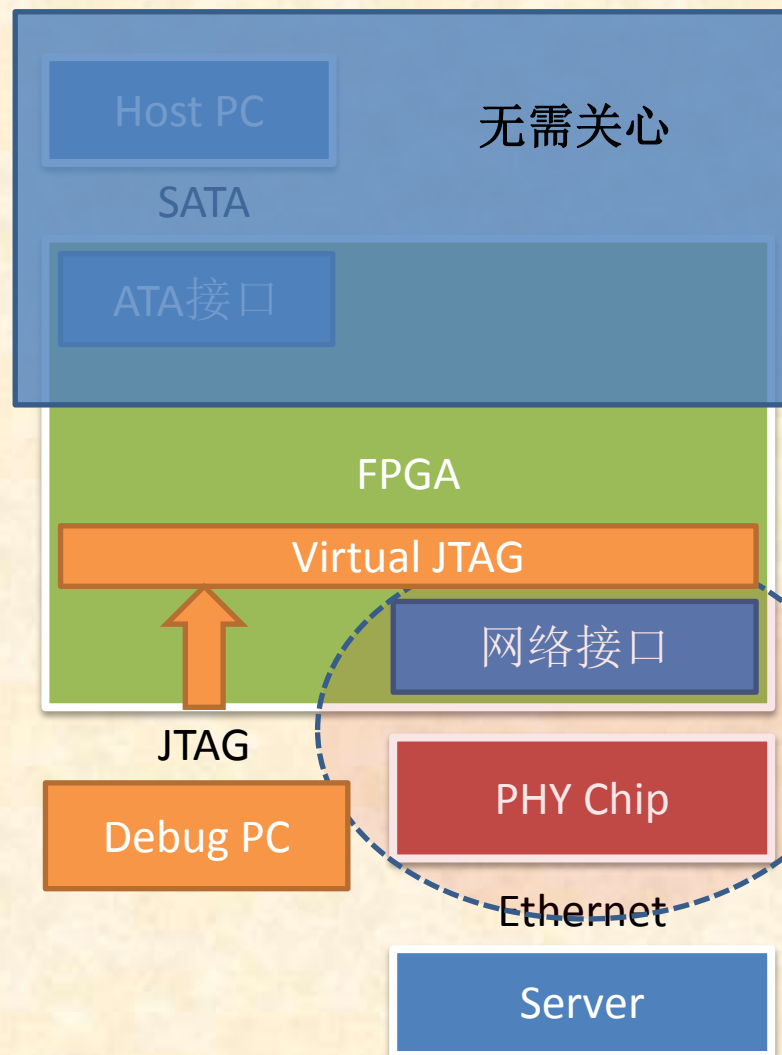
拥抱敏捷

- **FPGA开发本质上是敏捷的**
 - 相对于芯片开发：灵活 + 快速 = 敏捷
 - 可以随时更新- 更短的迭代周期
 - 可以上板验证 - 更快的反馈速度
- **敏捷开发的思想需要结合FPGA的自身特点**
 - 相对于软件开发：
 - 上板调试代价更高 - 仿真验证的必要性
 - 距离真实世界更近 - 开发验证、调试工具的必要性
 - » 验证、调试工具不是最终产品的一部分
 - » 开发工具、积累工具，合理重用
- **刻意地模仿芯片或者软件开发都是不可取的**

Agile Cases

案例分享

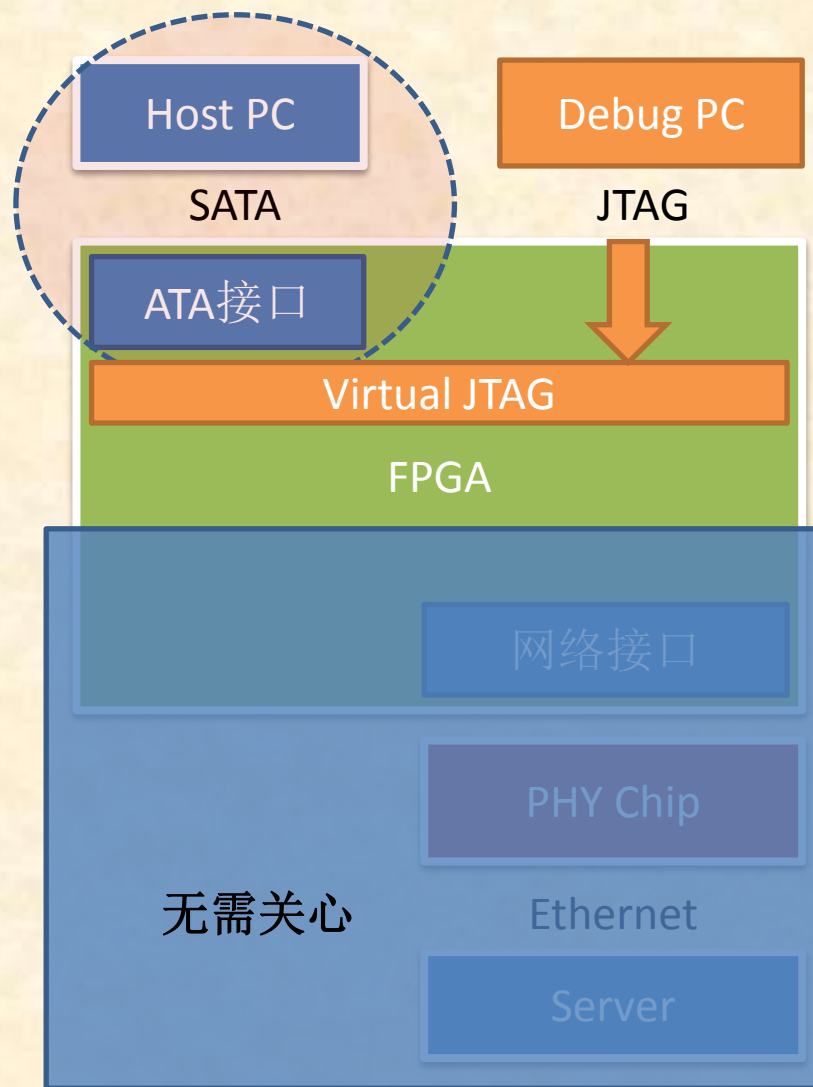
- 案例一：用Virtual JTAG加速上板验证
 - 问题1：
 - 缺少外部芯片的仿真模型。
 - 解决1：
 - 用VJ虚拟出一个“串口控制台”，通过接口逻辑层，向外部芯片发送“写”和“读”命令。验证接口逻辑层，积累操作外部芯片的脚本。
 - 通过这一接口，把最终产品中需要用状态机来实现的控制电路，暂时替换为可以用PC机软件来实现的脚本；把FPGA上的硬件调试转化为了PC端软件调试，加速了调试过程。
 - 敏捷思想：
 - 把项目中的风险，在早期的迭代中加以解决。



Agile Cases

案例分享

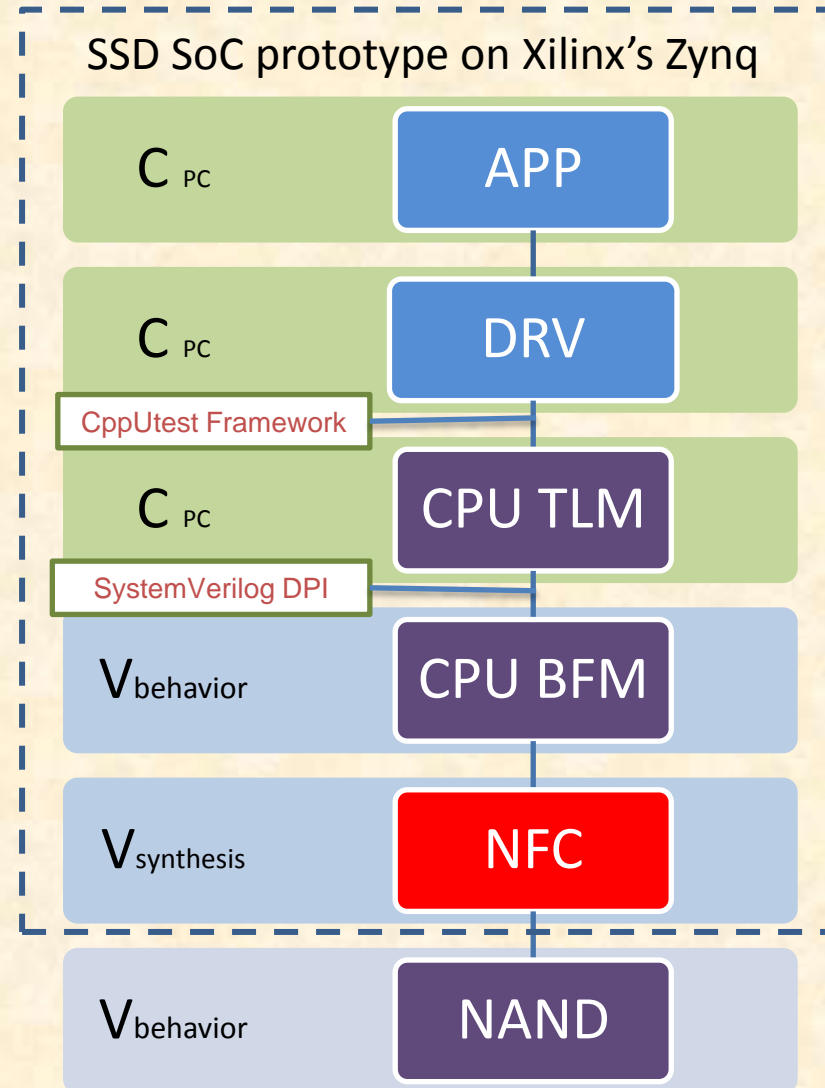
- 案例一：用Virtual JTAG加速上板验证
 - 问题2：
 - 外部设备行为复杂，测试向量覆盖率不够、仿真速度慢。
 - 解决2：
 - 用VJ设计一个真实行为捕获模块，丰富测试向量。
 - 用VJ虚拟未实现的关键逻辑模块，上板测试调试。
 - 敏捷思想：
 - 反思回顾，发现开发瓶颈，持续改进方法、工具。
 - 提早集成、提早测试。



Agile Cases

案例分享

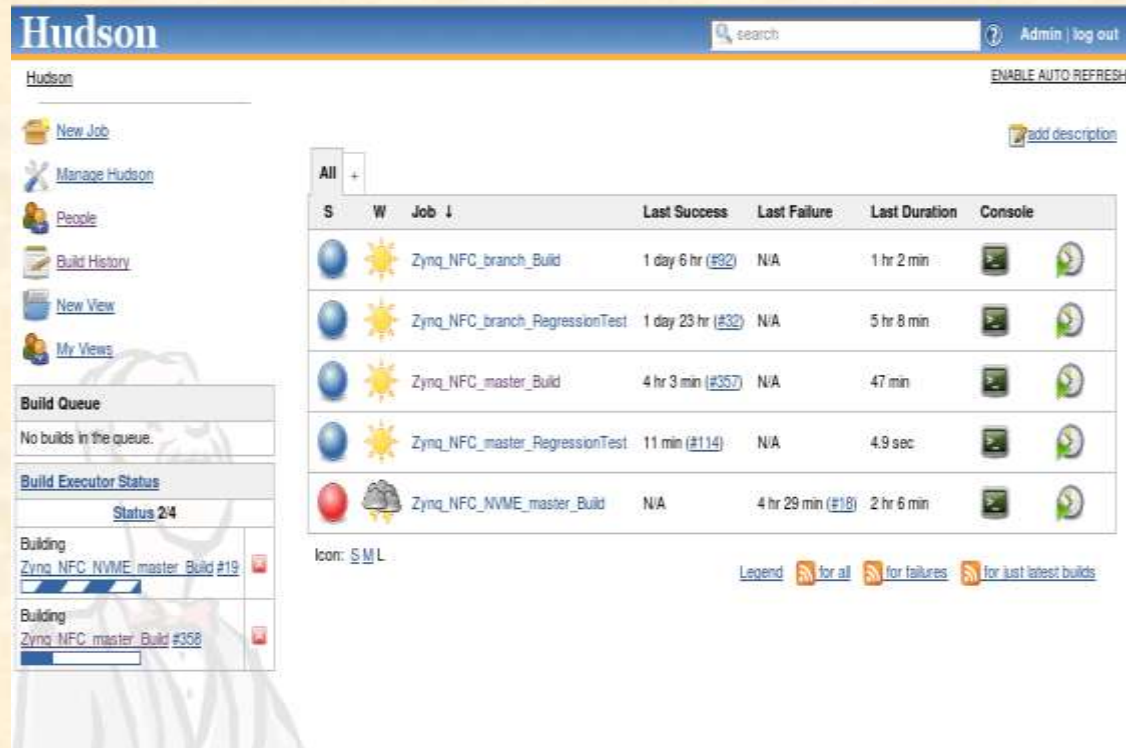
- 案例二：FPGA开发中引入回归测试和持续构建
 - 问题1:
 - 软件的驱动层（C）和FPGA的接口层（V）需要测试和调试
 - FPGA需要回归测试来保证持续开发的质量
 - 解决1:
 - 用SystemVerilog DPI连接C和Verilog
 - 用CppUtest测试框架搭建FPGA回归测试环境
 - 敏捷思想:
 - 回归测试



Agile Cases

案例分享

- 案例二：FPGA开发中引入回归测试和持续构建
 - 问题2：
 - 多人协作开发中代码的质量保证
 - FPGA需要自动构建来实现及时发布
 - 解决2：
 - 用Tcl脚本实现项目编译、仿真的自动化
 - 用Hudson CI来实现多个仿真、编译工程的自动构建和健康度报告
 - 敏捷思想：
 - 每日构建



The screenshot displays the Hudson CI web interface. The main content area shows a table of build jobs with columns for status (S), warning (W), job name, last success, last failure, last duration, and console links. The jobs listed are:

S	W	Job ↓	Last Success	Last Failure	Last Duration	Console
●	☀	Zynq_NFC_branch_Build	1 day 6 hr (#92)	N/A	1 hr 2 min	
●	☀	Zynq_NFC_branch_RegressionTest	1 day 23 hr (#32)	N/A	5 hr 8 min	
●	☀	Zynq_NFC_master_Build	4 hr 3 min (#357)	N/A	47 min	
●	☀	Zynq_NFC_master_RegressionTest	11 min (#114)	N/A	4.9 sec	
●	☀	Zynq_NFC_NVME_master_Build	N/A	4 hr 29 min (#18)	2 hr 6 min	

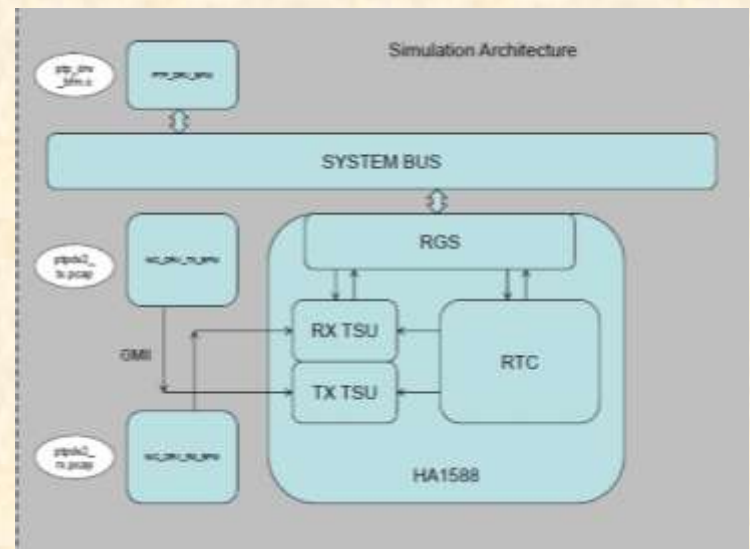
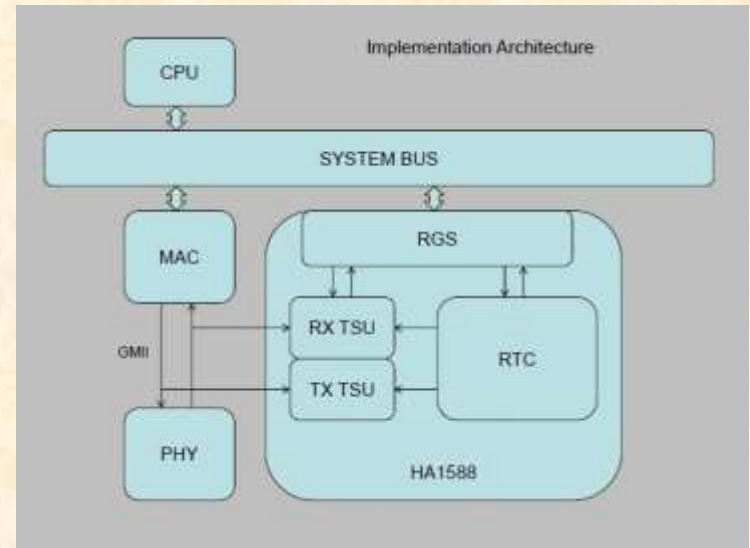
Below the table, there are links for 'Legend', 'for all', 'for failures', and 'for just latest builds'. The left sidebar contains navigation links such as 'New Job', 'Manage Hudson', 'People', 'Build History', 'New View', and 'My Views'. The 'Build Queue' section shows 'No builds in the queue.' and the 'Build Executor Status' section shows 'Status 2/4' with two active builds: 'Zynq_NFC_NVME_master_Build #19' and 'Zynq_NFC_master_Build #358'.

Agile Cases

案例分享

- 案例三：用FPGA领域外的工具来辅助验证
 - 问题：
 - 网络接口缺少测试向量
 - 解决：
 - 用WireShark的PCAP包格式作为仿真的输入输出格式
 - 敏捷思想：
 - TDD，以可验证的最终交付物为目标，杜绝半成品

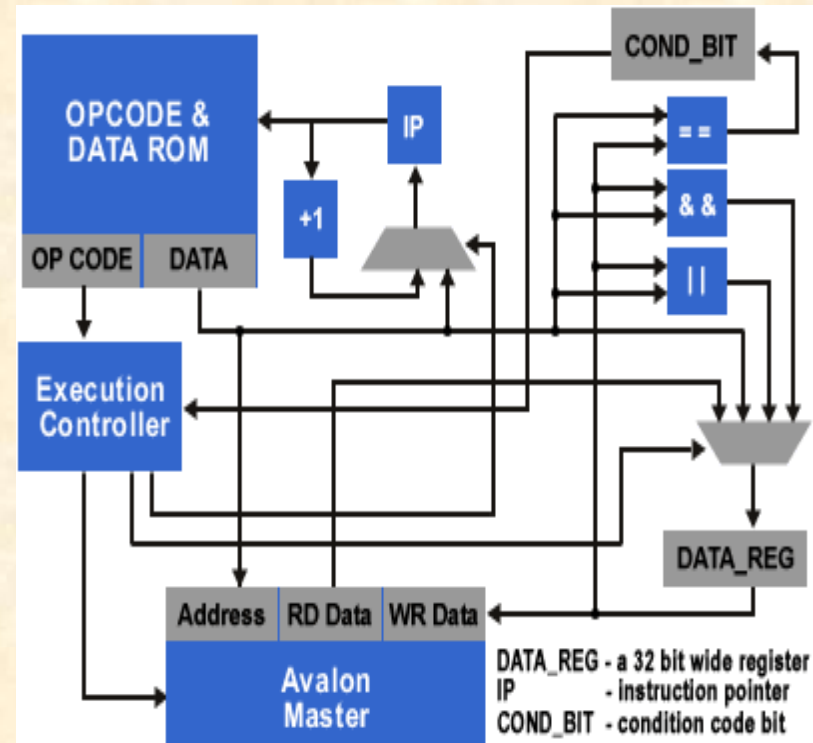
IEEE 1588 Hardware Assisted PTP



Agile Cases

案例分享

- 案例四：用“软”FSM代替“硬”FSM
 - 问题：
 - NAND Flash操作序列复杂、命令繁多、厂家之间存在差异，子状态机过多，导致层次化状态机过于庞大
 - 解决：
 - 用Micro-Sequencer来实现“软状态机”，用存储器中可以动态改写的“微码”代替用状态机实现的“硬连线电路”
 - 敏捷思想：
 - 个人和团队的持续改进



Thank You !

&

Questions?