

李枫 (独立开发者, hkli2012@126.com)

目的

由于市场对硬件产品功能、性能、开发周期的要求越来越高, Core/SoC的设计和验证也越来越复杂, 传统的低级 HDL(Verilog/

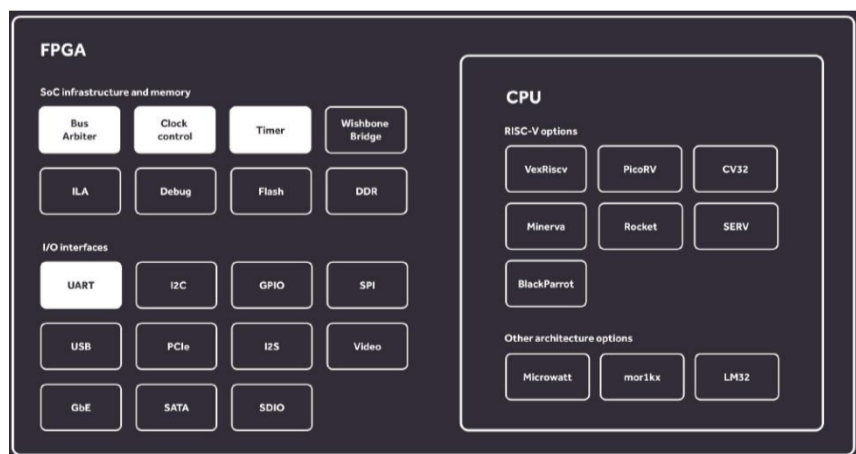
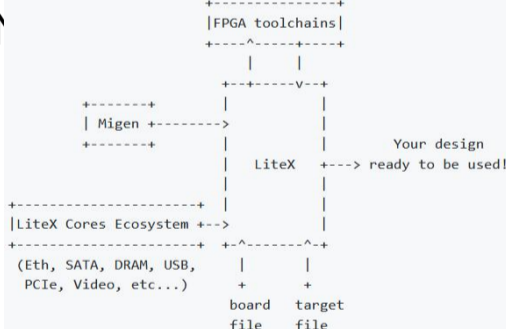
VHDL) 已不能很好的满足相关需求, 而HLS之类的“中级” HDL虽有长足的发展, 似乎仍未能解决好两大问题: 对硬件更高层次的抽象和较低的学习曲线。近几年, 以eDSL(嵌入式领域专用语言)和Core/SoC生成器为代表的新硬件设计开发方法异军突起, 很好地适应了硬件开发越来越“软件化”的总体趋势。

方法

eDSL是新一代的“高级” HDL, 它通过结合软件世界中的宿主语言和DSL的理念, 将自身嵌入到一门流行的宿主语言中, 从而可以最大程度地利用宿主语言已有的代码库和生态, 而借助宿主语言的流行性又可以显著地降低开发者的学习曲线。eDSL具备了对硬件更高层次的抽象能力, 从而实现了更高的生产效率和更好的代码复用。目前主要有三大类eDSL: 一类以Scala作为宿主(以Chisel和SpinalHDL为代表), 另一类基于Haskell(如Bluespec等), 还有一类正是我们关注的重点——以Python作为宿主。

LiteX/Migen

Migen是基于FHDL(嵌入Python的eDSL)的EDA工具箱, 它还包含了MiSoC/nMigen之类的衍生和扩展来更好的简化硬件设计/开发。而LiteX则是一个基于MiGen的Core/SoC构建器, 它包含了SoC设计框架和一组IP库/实用程序来高效地创建SoC和进行全定制FPGA设计。LiteX的生态正日益成熟, 已支持各种常见的硬件IP 和多种RISC-V软核, 并在真正的硬件产品开发中使用(如LibreSoC、↑



刚发布的LiteX 2021.04中已加入了对lowRISC Ibex的初始支持, 并且将硬件平台/目标板相关文件解耦到独立的项目litex-boards, 从而更好地将自身作为SoC框架。

结果

我们试图将基于LiteX的RISC-V开发工作迁移到像树莓派这样的开源硬件平台, 由于架构不同(树莓派等开源硬件多是ARM平台, 而LiteX官方目前只支持X86), 这样的迁移工作中会遇到很多问题需要解决。但随着苹果在下一个十年转向ARM, 以及基于苹果M系列/AWS Graviton等ARM芯片的桌面/云计算系统日益增多,

ARM生态圈正迎来非常重要的分水岭。而像树莓派这样的产品系列已经拥有了强大的生态以及在专业程序员/业余爱好者中的广泛影响力。所以将LiteX移植到ARM会是很有意义的尝试。

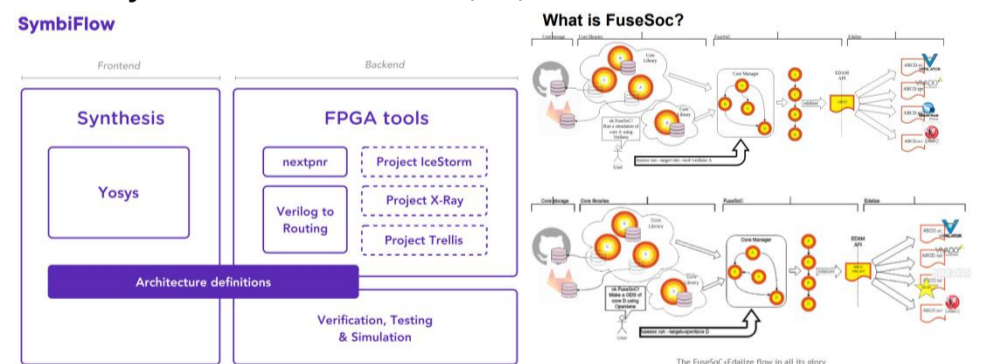
我们试验了Renode开源全系统仿真器在ARM上的移植(官方同样只支持X86)。Renode类似于QEMU, 但除了少量底层代码外主要由C#编写。Renode具有灵活的插件机制和很好的可扩展性(如可以与Wireshark/GDB/Verilator等集成), 它对RISC-V的支持非常全面, 特别适合软硬件协同开发的RISC-V。



我们还在iCESugar-Pro FPGA开发板(使用Lattice ECP5开源RISC-V Linux)上实践了LiteX, 并正在准备一个官方虚拟机镜像(X64)之外的面向树莓派4的开发环境。



虽然现在包括LiteX在内的基于Python的EDA工具越来越多(如SymbiFlow、FuseSoC等), 但Python的性能一直是弱点。我们探索了多种Python新运行时并在ARM上进行了实验, 以期望未来能进一步提升这些Python EDA项目的效率。



结论

- 1) 像Chisel/Rocket Chip和MiGen/LiteX这样基于新一代HDL的Core/SoC 框架为硬件开发带来了更高的生产率, 它们已经在RISC-V等新兴硬件设计开发领域获得了广泛和真实的应用;
- 2) 开源EDA工具越来越多, 生态也愈加繁荣;
- 3) 由于Python自身的特点——简洁、易用、高产出, 强大的生态, 以及在专业和非专业开发人员中的影响力等, 使得Python在构建开源EDA工具链中正在发挥越来越重要的作用;
- 4) ARM软硬件生态正迎来新一轮爆发期, 将传统的基于X86主机的跨平台开发转向ARM, 为开发者带来了更多的选择和新的体验。