

中期进度汇报

COD19GRP4

概览

- ▶ 成果 (DEMO)

- ▶ 把MAC/IP写死，可以4个网口连接相互ping通
 - ▶ (没有连接ARP表和路由表)

- ▶ 简约的ARP表

- ▶ 线性算法

- ▶ 高效的路由表

- ▶ 32拍Trie树(第五周)
 - ▶ 1/2/4/8/16/32步长可参数化多路Trie树(第7周)

概览

▶ 智能的Testbench

- ▶ Python自动化生成ARP/路由/以太网帧Testcases, 可指定数量和各种类型包的比例
- ▶ Vivado读/写文件 可以显示每个操作预期结果/实际结果/对错

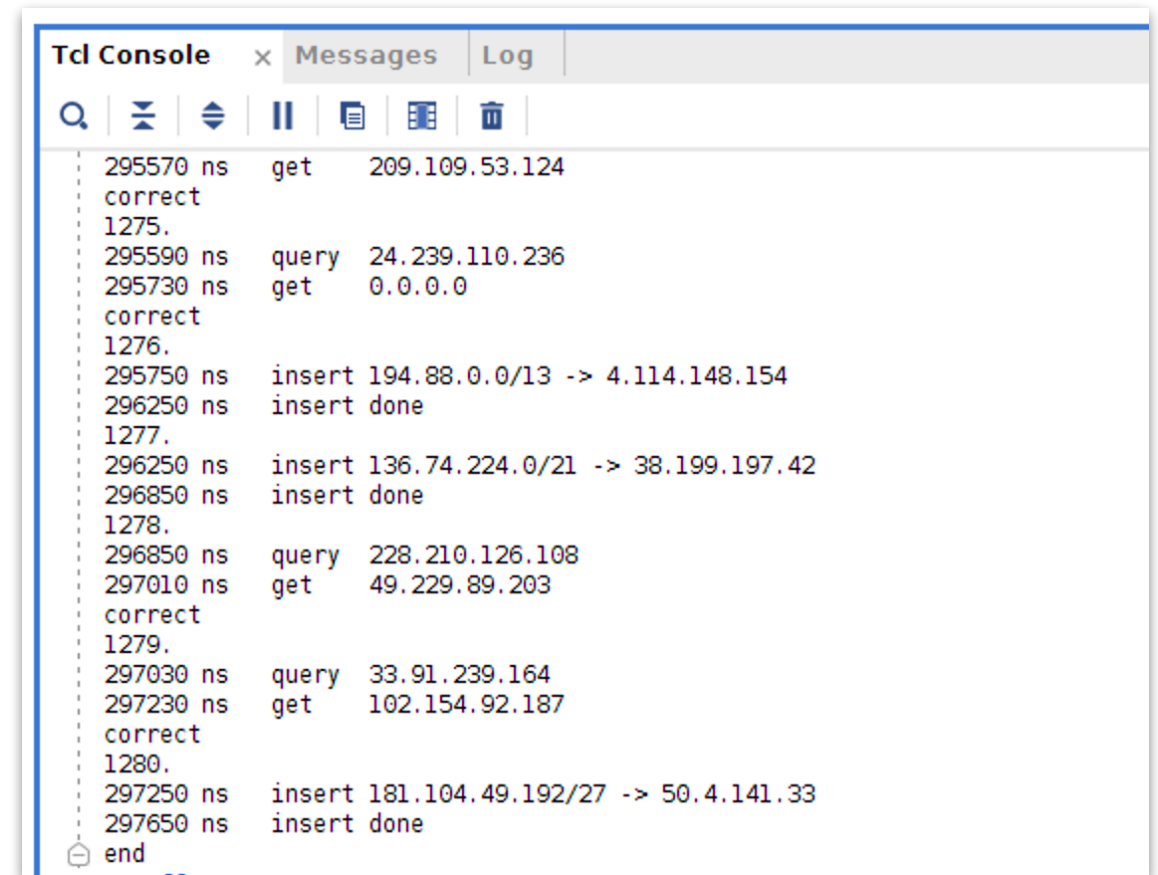
▶ 优美的Debug方式 (DEMO)

- ▶ LED/7端数码管显示自动机状态
- ▶ 通过硬件按钮控制转发逻辑(如丢包)
- ▶ 内嵌逻辑分析仪接到队列RX和TX

▶ 规范的Git版本控制

▶ (不) 勤奋的组员

- ▶ 还把ALU/SRAM/UART小作业写了



```
Tcl Console x Messages Log
[Icons]
295570 ns get 209.109.53.124
correct
1275.
295590 ns query 24.239.110.236
295730 ns get 0.0.0.0
correct
1276.
295750 ns insert 194.88.0.0/13 -> 4.114.148.154
296250 ns insert done
1277.
296250 ns insert 136.74.224.0/21 -> 38.199.197.42
296850 ns insert done
1278.
296850 ns query 228.210.126.108
297010 ns get 49.229.89.203
correct
1279.
297030 ns query 33.91.239.164
297230 ns get 102.154.92.187
correct
1280.
297250 ns insert 181.104.49.192/27 -> 50.4.141.33
297650 ns insert done
end
```

概览

▶ 智能的Testbench

- ▶ Python自动化生成ARP/路由/以太网帧Testcases
- ▶ Vivado读/写文件 可以显示每个操作预期结果/实际结果/对错

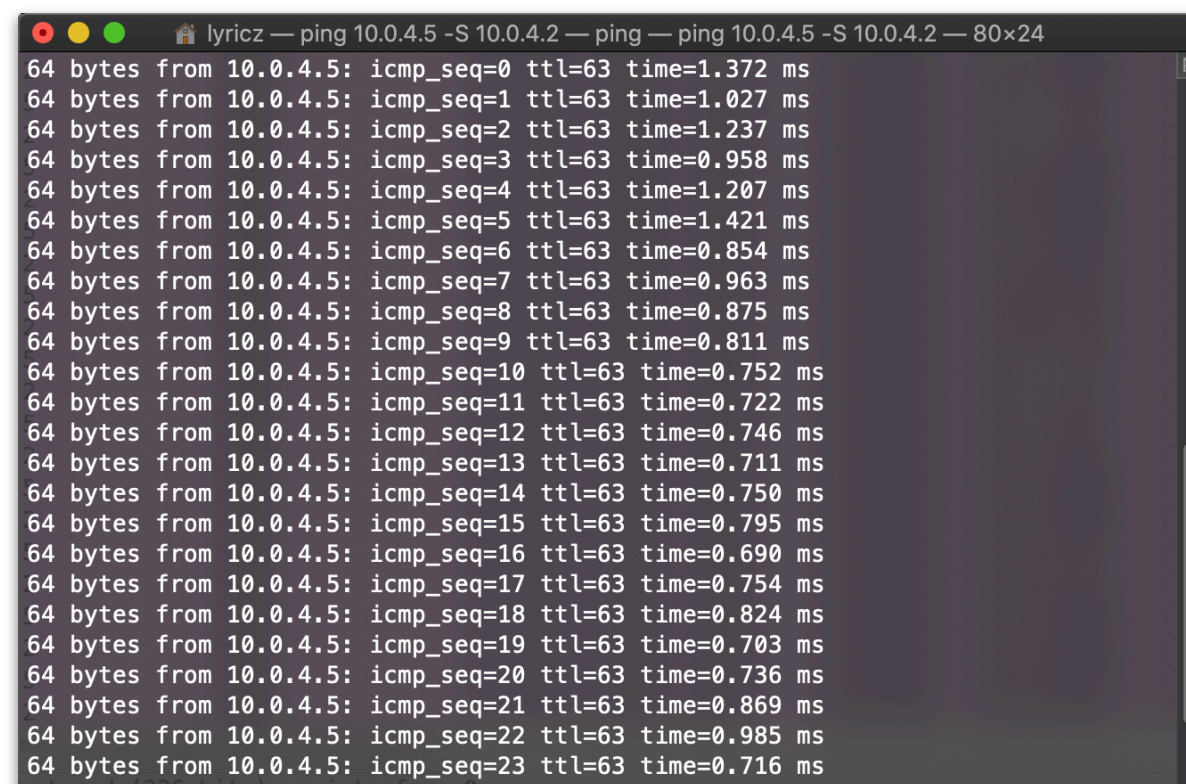
▶ 优美的Debug方式 (DEMO)

- ▶ LED/7端数码管显示自动机状态
- ▶ 通过硬件按钮控制转发逻辑(如丢包)
- ▶ 内嵌逻辑分析仪接到队列RX和TX

▶ 规范的Git版本控制

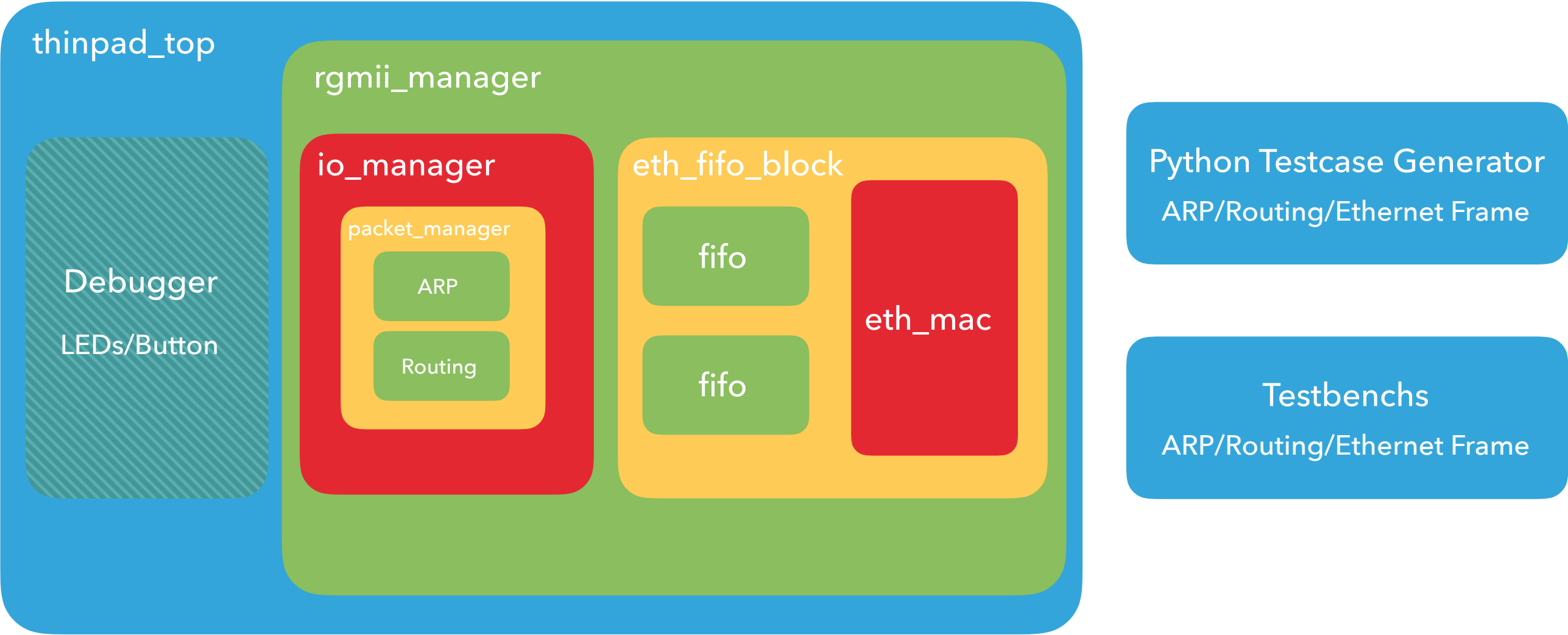
▶ (不) 勤奋的组员

- ▶ 还把ALU/SRAM/UART小作业写了



```
lyricz — ping 10.0.4.5 -S 10.0.4.2 — ping — ping 10.0.4.5 -S 10.0.4.2 — 80x24
64 bytes from 10.0.4.5: icmp_seq=0 ttl=63 time=1.372 ms
64 bytes from 10.0.4.5: icmp_seq=1 ttl=63 time=1.027 ms
64 bytes from 10.0.4.5: icmp_seq=2 ttl=63 time=1.237 ms
64 bytes from 10.0.4.5: icmp_seq=3 ttl=63 time=0.958 ms
64 bytes from 10.0.4.5: icmp_seq=4 ttl=63 time=1.207 ms
64 bytes from 10.0.4.5: icmp_seq=5 ttl=63 time=1.421 ms
64 bytes from 10.0.4.5: icmp_seq=6 ttl=63 time=0.854 ms
64 bytes from 10.0.4.5: icmp_seq=7 ttl=63 time=0.963 ms
64 bytes from 10.0.4.5: icmp_seq=8 ttl=63 time=0.875 ms
64 bytes from 10.0.4.5: icmp_seq=9 ttl=63 time=0.811 ms
64 bytes from 10.0.4.5: icmp_seq=10 ttl=63 time=0.752 ms
64 bytes from 10.0.4.5: icmp_seq=11 ttl=63 time=0.722 ms
64 bytes from 10.0.4.5: icmp_seq=12 ttl=63 time=0.746 ms
64 bytes from 10.0.4.5: icmp_seq=13 ttl=63 time=0.711 ms
64 bytes from 10.0.4.5: icmp_seq=14 ttl=63 time=0.750 ms
64 bytes from 10.0.4.5: icmp_seq=15 ttl=63 time=0.795 ms
64 bytes from 10.0.4.5: icmp_seq=16 ttl=63 time=0.690 ms
64 bytes from 10.0.4.5: icmp_seq=17 ttl=63 time=0.754 ms
64 bytes from 10.0.4.5: icmp_seq=18 ttl=63 time=0.824 ms
64 bytes from 10.0.4.5: icmp_seq=19 ttl=63 time=0.703 ms
64 bytes from 10.0.4.5: icmp_seq=20 ttl=63 time=0.736 ms
64 bytes from 10.0.4.5: icmp_seq=21 ttl=63 time=0.869 ms
64 bytes from 10.0.4.5: icmp_seq=22 ttl=63 time=0.985 ms
64 bytes from 10.0.4.5: icmp_seq=23 ttl=63 time=0.716 ms
```

结构



一些坑

- ▶ 夯实基础概念少走弯路（可能除了我们组大家都知道）
 - ▶ KSZ8795是个交换机，FPGA是要写的路由器（之前一直以为整个是一个路由）
 - ▶ 为了测试ping是不是通的，子网掩码可以设置成255.255.255.255，不然直接会被交换机扔回去（交换机会学习到MAC和Port的映射）
- ▶ VLAN从1开始编号不是0
- ▶ 对于ping请求Windows有防火墙，要先关了
- ▶ 如何使用有bug的macOS Catalina连接Thinpad
 - ▶ 使用最新版的Parallel Desktop **15**，把Thinpad分配给虚拟机
- ▶ ILA监控两个FIFO的信号非常方便调试
 - ▶ Open Synthesis Design → Debug Set Up
 - ▶ 或者IP Catalog里面添加
- ▶ PLL分频之后locked信号可以作为后面所有模块的复位
- ▶ 如何用一台电脑的两个网口测试
 - ▶ ping -S或者ping -I指定网口
 - ▶ 或者一个网口分配给虚拟机
- ▶ 12V电源线转USB淘宝有卖，再连一个USB/RJ45的Hub即可一根Type-C连接到电脑非常方便，断电也能调试
- ▶ 调自闭了问几位大哥或者出去换个脑子

后面打算

- ▶ 把ARP表和路由彻底连上
- ▶ 更快的ARP算法
- ▶ 改成流水
- ▶ 造CPU

DEMO

谢谢