中期进度汇报

COD19GRP4

概览

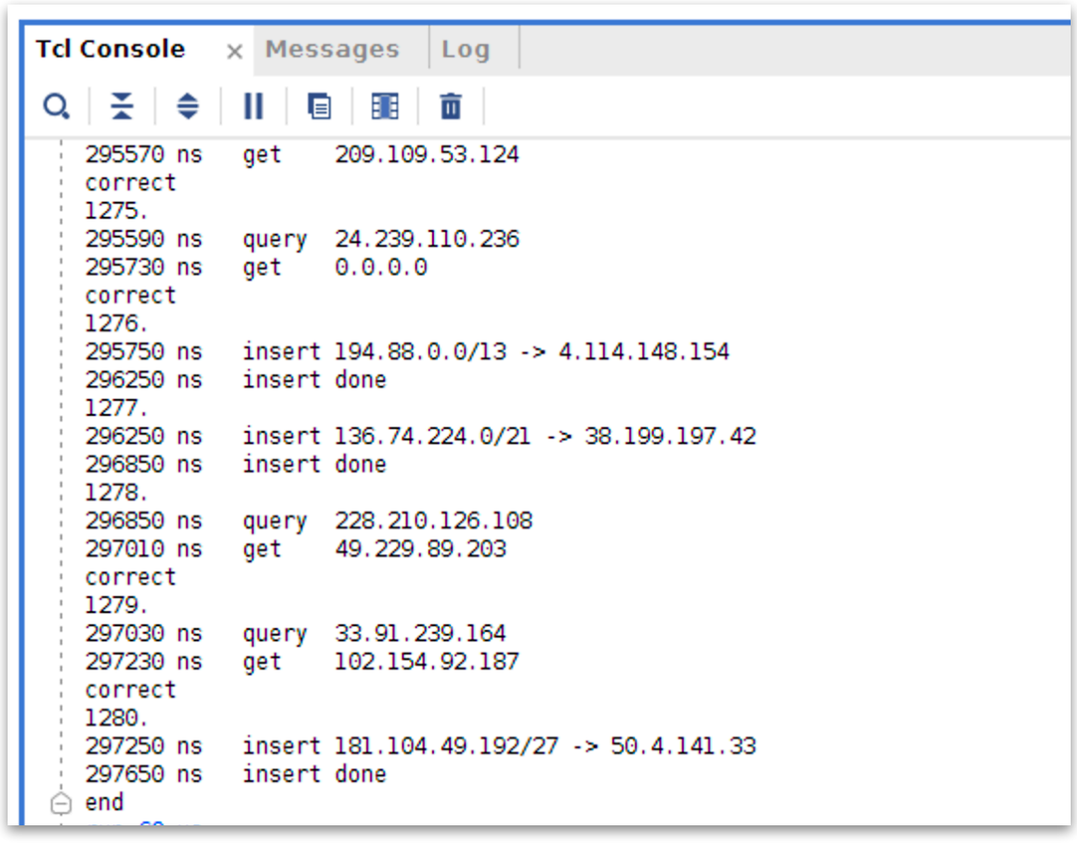
- 成果 (DEMO)
  - 把MAC/IP写死，可以4个网口连接相互ping通
  - (没有连接ARP表和路由表)

```
64 bytes from 10.0.4.5: icmp_seq=5 ttl=63 time=1057.393 ms
64 bytes from 10.0.4.5: icmp_seq=6 ttl=63 time=1001.291 ms
64 bytes from 10.0.4.5: icmp_seq=7 ttl=63 time=1248.631 ms
64 bytes from 10.0.4.5: icmp_seq=8 ttl=63 time=1008.399 ms
64 bytes from 10.0.4.5: icmp_seq=9 ttl=63 time=1000.940 ms
64 bytes from 10.0.4.5: icmp_seq=10 ttl=63 time=1003.808 ms
64 bytes from 10.0.4.5: icmp_seq=11 ttl=63 time=1022.109 ms
64 bytes from 10.0.4.5: icmp_seq=12 ttl=63 time=1278.183 ms
64 bytes from 10.0.4.5: icmp_seq=13 ttl=63 time=1727.503 ms
64 bytes from 10.0.4.5: icmp_seq=14 ttl=63 time=1980.193 ms
```

- 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)

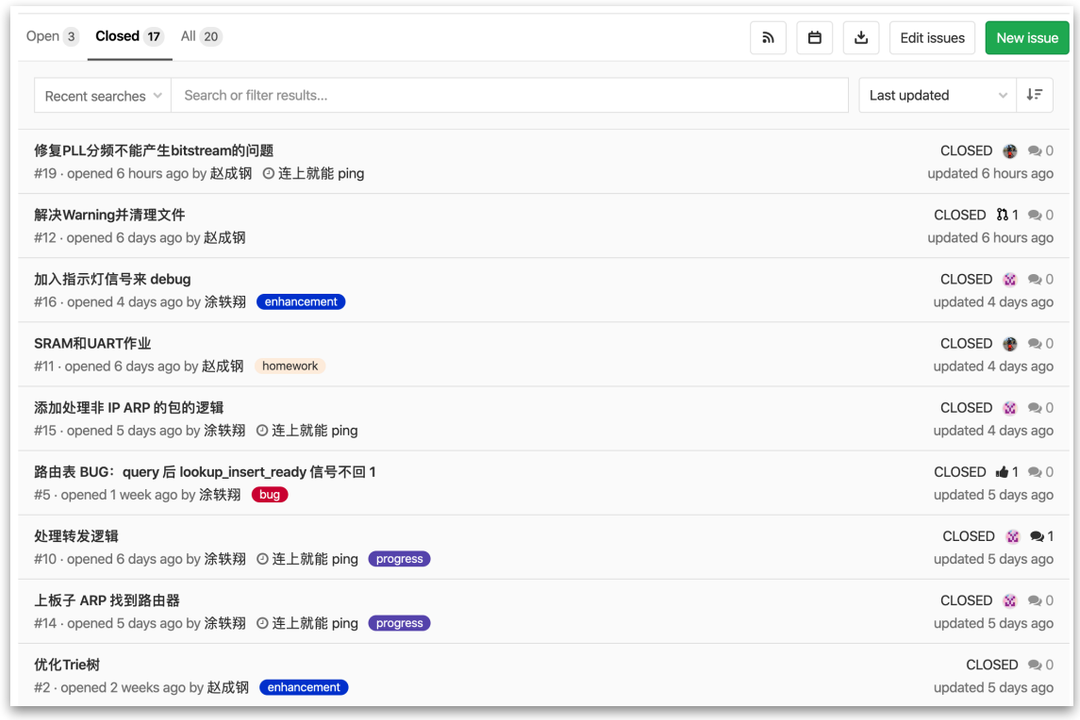
概览

- 智能的Testbench
  - Python自动化生成ARP/路由/以太网帧Testcases
  - 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

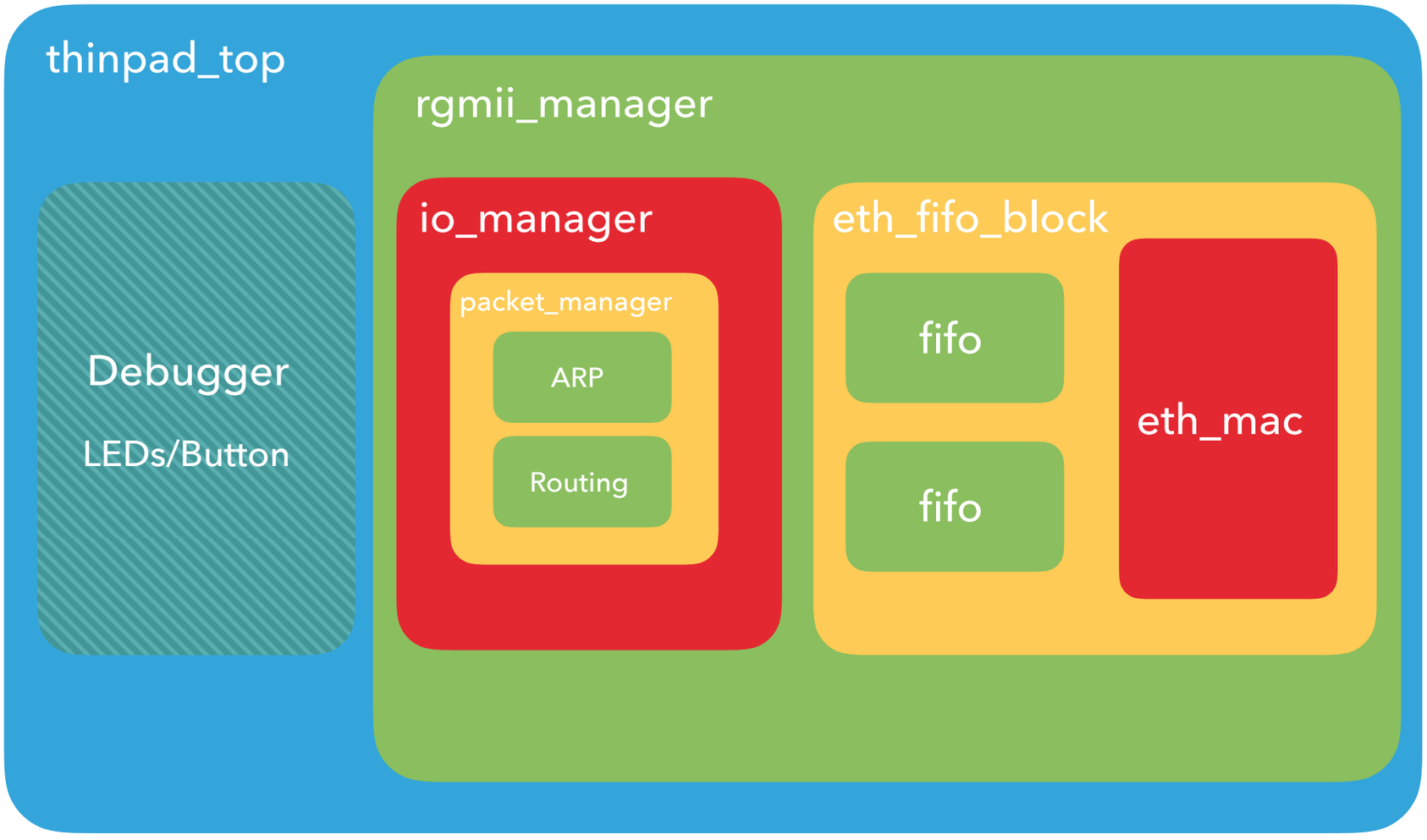


概览

- 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



结构



Python Testcase Generator

ARP/Routing/Ethernet Frame

Testbenchs

ARP/Routing/Ethernet Frame

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