

asyn_fifo 待测设计有 6 个模块：

wptr_full.v、 sync_w2r.v、 sync_r2w.v、 rptr_empty.v、 fifomem.v、 asyn_fifo.v

Design:

asyn_fifo.v 是顶层模块，在它里面例化了前 5 个模块。

wptr_full.v 是产生写地址和写指针的模块

rptr_empty.v 是产生读地址和读指针的模块

sync_w2r.v 是将写时钟域传来的指针同步进入读时钟域的模块

sync_r2w.v 是将读时钟域传来的指针同步进入写时钟域的模块

fifomem.v 是存储部分

UVM Verification:

top_tb.sv 是整个验证平台的顶层模块（测试用具）。在它里面连接了待测设计和验证平台。

my_if.sv 定义了一个接口，用来连接待测设计和验证组件。

my_transaction.sv 定义了验证平台中传递信息的事务。它里面包含一些变量。

my_driver.sv 发送激励。

in_monitor.sv 在信号输入端口监测。

out_monitor.sv 在信号输出端口监测。

my_sequencer.sv 是 UVM 搭建的验证平台中不可或缺的部件，负责中转由 sequence 传来的 transaction。

i_agt.sv 是一个容器类，里面实例化了 my_driver、my_sequencer 和 in_monitor。

o_agt.sv 也是一个容器类，里面实例化了 out_monitor。

my_model 是参考模型，由于 fifo 实现的功能较为简单，此处的参考模型仅仅是把输入的数据复制一遍然后输出。

my_scoreboard.sv 是计分板，用于比较待测硬件的输出和参考模型的输出是否一致。

my_env.sv 是一个容器，它里面实例化了 i_agt、o_agt、my_model、my_scoreboard 等部件。

base_test.sv 它里面实例化 my_env，同时也会规定信息打印的规则。

my_case0 和 my_case1 是两个测试用例。my_case0 可以运行成功，my_case1 运行不成功，运行不成功的原因是 uvm-1.2 版本不支持 default_sequence 的使用或者是 uvm-1.2 版本支持，但是我不会用。

fifo_rst_mon.sv 和 fifo_chk_rst.sv，前者用来检测复位信号，后者用来检测复位后寄存器的复位值是否正确。由于这个是我第一次用 UVM 搭建的验证平台，所以许多地方还不成熟，比如 my_model 仅仅是为了验证平台的完整而添加的，实际上并不需要。

Makefile 使用说明：

make COMPILE 命令编译

make SIMULATION 命令仿真运行

make URG 产生覆盖率报告

make clean 清除运行过程中的冗余文件