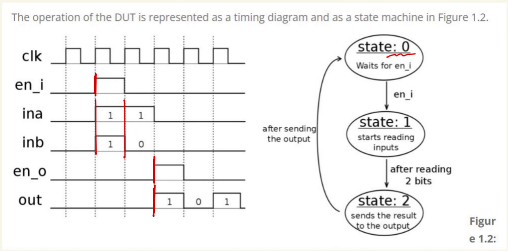
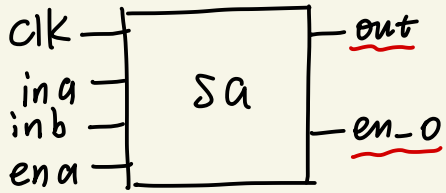
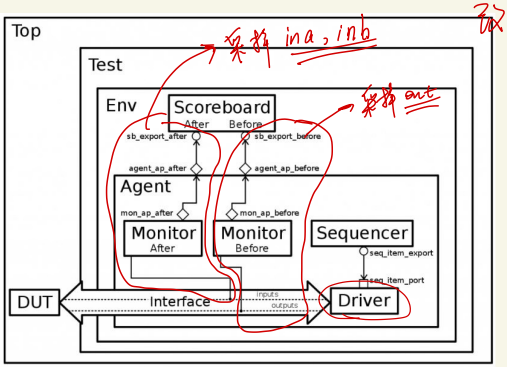
五、小模块1：simpleadder







改成：只用一个 monitor

采样 ina，inb，out 的 transaction ⑨ 为什么要有 simpleadder_configuration

传到 Scoreboard 的 transaction 1

把 transaction 1 复制给 transaction 2

⑧ simpleadder_transaction_3inputs 是干什么的

⑦ 每人写2个 test 和对应的 case 给对方讲解思路

再用 predictor() 函数计算出 ref_model 最后的数 transaction 1 和 transaction 2.

⑩ 思考 reference model 还有没有另外的写法

⑪ 自行查习遗漏的知识点 + 讨论

实战：

同时复习对应的知识点.

SV

① 将 state 改为 enum 类型

② 在 interface 中添加 clocking，modport

③ driver 里的逻辑能否优化

④ monitor 里的逻辑能否优化.

⑤ 自行查习遗漏的知识点 + 讨论

UVM

① 了解 uvm_config_db 和 uvm_resorce_db 的异同

② 为什么 new 函数中可以不写名字.

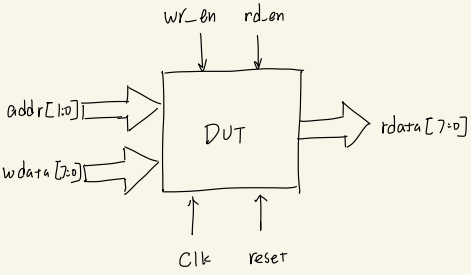
③ 把 predictor() 函数从 monitor 中移动到 scoreboard 里

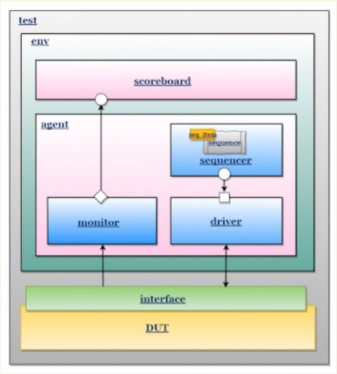
④ 删去 agent 中的两个 TLM port，直接将两个 monitor 和 scoreboard 相连

⑤ 为什么要用 uvm_tlm_analysis_fifo，有何用处？如何用？

⑥ 如何确定 scoreboard 中拼比较的两个包是一一对应的？如何避免错位

小模块2：Memory model





write: addr、wr_en、wdata → driven at the SAME clock cycle

read: addr、rd_en → DUT will respond with the rdata in the NEXT clock cycle

Reset: each addr → 'hFF

① write → any location ---- should same ---- → read

② write → 2'b00 / 2'b01 / 2'b10 / 2'b11 → Read

③ Default value check: Read with no write → Should be 'hFF

④ Reset in the middle of Write/Read

write → any location → Reset → Read → Should 'hFF

注意：

① `uvm_declare_p_sequencer

② wait_for_grant( );

③ send_request(req);

④ wait_for_item_done();

⑤ `uvm_do_with

⑥ set_drain_time(this, 50)

① driver/monitor 逻辑优化

② scoreboard 如何确保一一对应

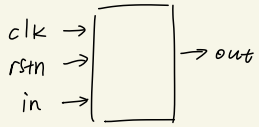
③ reference model 优化

④ 自己写n个case跑一跑

⑤ 写 covergroup

小模块3：det_1011

every clock → input

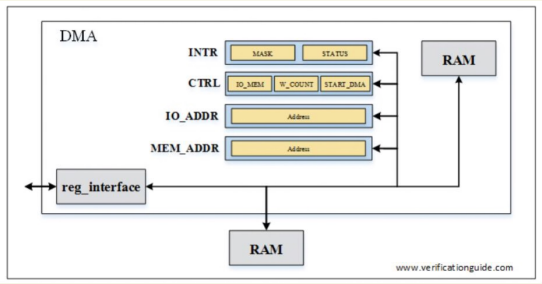


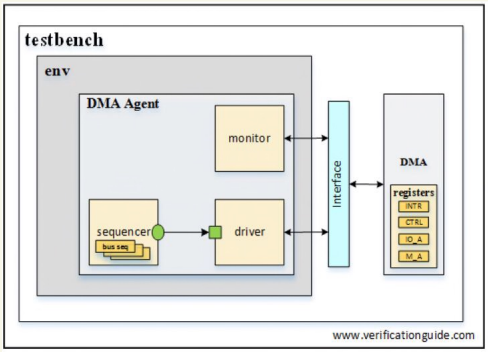
match → out = 1

① 看懂输出波形
② driver 优化
③ monitor 优化
④ scoreboard 优化
⑤ reference model 优化
⑥ 多写2个 case（sequence + test）

小模块4：dma

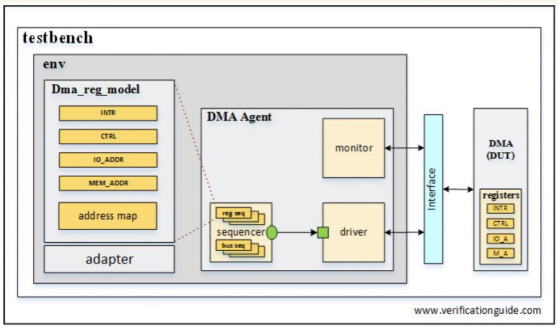
sequence { write / read





Reg Name	Address	Field Description			
INTR	h400	31:16		15:0	
		MASK		STATUS	
CTRL	h404	31:10	9	8:1	0
		RESVD	IO_MEM	W_COUNT	START_DMA
IO_ADDR	h408	31:0			
		ADDRESS			
MEM_ADDR	h40C	31:0			
		ADDRESS			

- Writing register classes
- Writing register package
- Instantiation of register classes in register package
- Writing Adapter class
- Integrating register package and adapter in environment
- Accessing registers with RAL



A1

Simpleadder → DMA → cnn

AHB_SRAMC

MCU

一. 目的，预期？

2022 11月、12月

2023 1月、2月、3月、4月、5月

秋招（面试，笔试）

上手

二，通过项目

① (入职后) 编写测试 case

sequence，test

☆如何测试功能点

项目

②(秋招会问) 搭建框架，组件连接…

前期准备工作，公司有专人搭环境

但可体现对项目的理解和基础知识掌握度

↓协议

(driver, monitor, scoreboard)

↓

(SystemVerilog, UVM, Others)

…

边角知识

Verilog

数电

行业认知

③ 精通 1-2个"简历"项目

三，下一步的方向

① 基础知识实战化

simpleadder

memory

det_1011

dma_reg

router_REG

FIFO(sync、async)

…

小模块

一周过一个

实操

<不停止搜寻新的项目>

mcdf

I2C，uart → 参考

AHB2APB

AHB_SRAMC <简单>

DMAC <带奇存器> <中等>

Cnn <困难>

MCU <困难>

大项目

2到3个月一个

四、合作方式

我的提议：①②同步并行

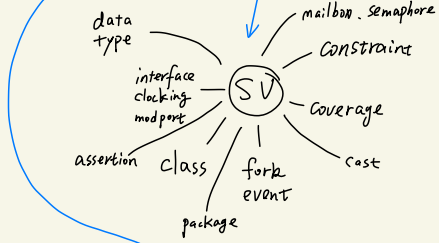
学生思维　工程思维

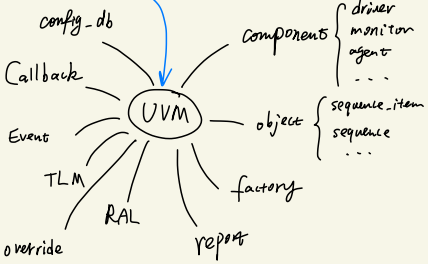
A2

自行参考我们手上的所有资料　基于每一个小模块做一些小实验　实现练手的目的

绿牌 UVM实战　CV网站　B课网视频　…

路科 V2.1 V2.2 V2.3　IEEE的SV官方手册　UVM官方文档





选同一个培训班项目做

（可以讨论视频里的内容）

先学习其规范性

利于下一步全凭自己验DMA

自行参考充分利用<不可写进简历的项目>

- mcdf
- I2C
- uart
- AHB2APB
- SPI
- …

以手上目前资料，建议AHB_SOAMC（路科）　可写进简历

不排除以后有更好的新开培训班项目

cnn，MCU难度较大，暂放

随时想一些（可以上手的小实验）

视频中、项目中任何想不通的

随时开短会

微信Voice/Video

交流讨论

（一次几分钟，高效率）

每周四固定开一次长会

总结+计划

（2小时左右）

① 基础知识实战化

基础扎实笔试、都会

② 精通1-2个"简历"项目

项目精通，面试不慌

一. 实战项目

BIG

△ AHB_SRAMC

☆ DMAC

☆ cnn

☆ 继续寻找新项目(盗版 or 开源)

☆ FIFO(sync、async)

small

△ simpleadder

△ memory

△ det_lo11

△ dma_without_ral

△ dma_with_ral

☆ router_reg

☆ I2C, uart, AHB2APB, SPI

⊗ mcdf (太杂乱,性价比太低)

⊗ MCU(无代码)

B

二. 基础知识

△ 精通 System Verilog

△ 精通 UVM

✓ 熟练操作工具: VCS(QuestaSim), DVE(Verdi), TCL
Linux 命令行, bash 脚本, Makefile
vim, Regular Expression, Perl/Python

✓ 除验证以外的必备知识: 数电、Verilog
<DC、ICC>
(数集,设计方法学, FPGA)
(模电,模集,射频)

集训目标

三. 刷笔试题

✓ 牛客网

✓ 其他资料

✓ 四. 修改简历

✓ 五. 模拟面试

标志含义

△ 正在进行

☆ 即将进行

✓ (比较轻松的任务)在状态佬总时进行

⊗ 舍弃掉