

SHANGHAI JIAO TONG UNIVERSITY



**System Verilog project实验报告**

**张铭钰**

**122039910033**

**2023.1.27**

**System Verilog Final Project**

**1 工程内容概述**

本次工程为一个I2C通信用的主设备模块测试，该模块作为主机工作在I2C总线上，通过一个APB接口来接收配置指令，从而驱动I2C总线完成通信交互。提供的sim环境如图1-1所示，其中I2C\_master接收APB配置的指令并通过IOBUF生成I2C通信用的SCL和SDA信号，与一个EEPROM存储器件进行读写访存交互，I2C\_master和EEPROM作为主机与从机工作在SCL/SDA总线上。

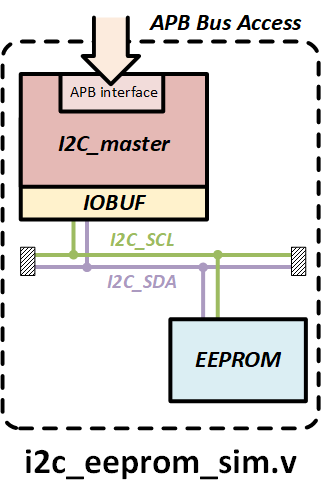


图1-1 待验证sim环境

**2 工程模块分析**

本次工程涉及到的模块主要包括APB接口、I2C\_master和存储器件EEPROM，其中APB接口用于接收配置指令并驱动I2C总线，I2C\_master和EEPROM分别作为主机与从机工作在总线上以完成通信，本节将分析各个模块的工作原理及波形。

**2.1 APB接口**

APB是一种低成本的接口协议，每次传输至少需要消耗两个时钟周期，其端口包括PCLK、PRESETn、PADDR、PSELx、PENABLE、PWRITE、PWDATA、PRDATA、PSLVERR以及PREADY，共十个端口。

通常APB的数据传输事务状态主要过程如图2-1所示，其中PREADY和PSLVERR信号均源于从机，用于等待及错误返回。

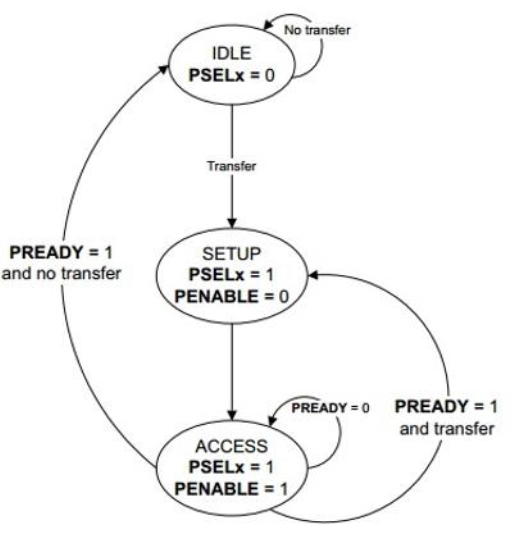
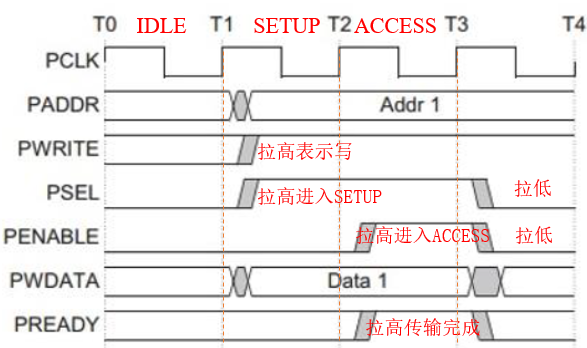


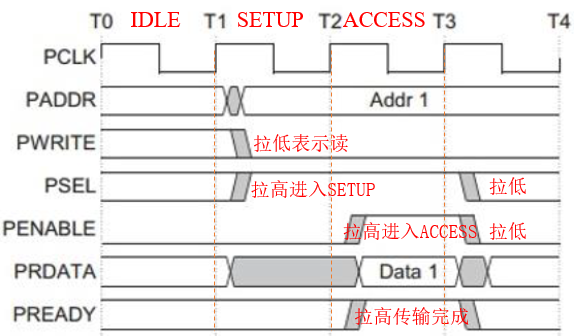
图2-1 APB数据传输状态

如上图所示，APB的默认状态是IDLE，此时无数据传输，PSEL和PENABLE均拉低；当需进行数据传输时，PSELx信号先拉高，APB进入次状态SETUP；停留一个时钟周期后PENABLE信号被拉高，SETUP状态转换到ACCESS状态；若此时PREADY为拉低，表示从机未准备好，则继续维持在ACCESS状态，若PREADY为拉高，表示从机已完成数据传输，可跳转到其他状态：若有连续数据传输则跳转到SETUP，否则直接跳到IDLE。

根据上述APB数据传输状态分析可以得出其写/读时序波形如图2-2（a）、（b）所示。



（a） APB写传输时序波形



（b） APB读传输时序波形

图2-2 APB写/读数据传输时序波形

由上述时序波形可以了解到APB接口进行数据传输的流程。

**2.2 I2C模块**

I2C是一种两线式串行总线，其包含两条总线线路：一条串行数据线（SDA）和一条串行时钟线（SCL），均为双向I/O线，数据线SDA的高低电平传输二进制的数据，时钟线SCL通过方波信号提高时钟节拍，每个连接到总线的设备都可以使用唯一地址来识别，通过I2C总线可实现主机与从机间的数据传输，其中起始和终止信号均由主机发出，并等待从机给出一个应答信号。

I2C的可配置寄存器的偏移地址、位宽、读写属性及功能如表2-1所示。

表2-1 I2C的可配置寄存器

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **寄存器** | **偏移地址** | **位宽** | **读写属性** | **功能** |
| I2C\_PRE | 0x00 | 32 | 可读可写 | 对时钟域进行分频得到SCL时钟频率 |
| I2C\_CTR | 0x04 | 32 | 可读可写 | 配置I2C的功能是否打开（第6/7位） |
| I2C\_TX | 0x10 | 32 | 可读可写 | I2C将发送的1字节数据 |
| I2C\_RX | 0x08 | 32 | 只读 | I2C接收的1字节数据 |
| I2C\_CMD | 0x14 | 32 | 可读可写 | 发起命令请求，完成后自动清除 |
| I2C\_STATUS | 0x0C | 32 | 只读 | 监视总线操作，查看总线状态 |

该I2C的基地址为1002\_5000，在此基础上的偏移地址为其内部的寄存器地址，使用地址可实现对内部寄存器的访问。

上表中的I2C内部寄存器主要用于其内部实现数据写/读操作，数据的完整传输时序图如图2-3所示，其中ACK信号由从机发出表示应答。

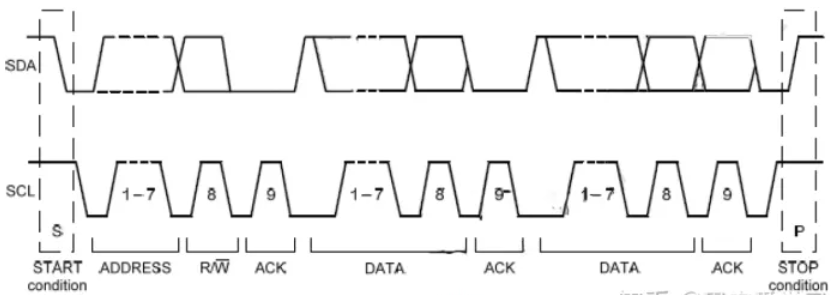


图2-3 I2C完整数据传输时序波形

1. 起始：时钟线SCL为高时，数据线SDA由高到低；
2. 地址传送：主机传送一个7位的地址（ADDRESS），第8位为读/写位（R/W）；
3. 应答信号：主机释放SDA线等待从机的应答信号（ACK）；应答产生时，从机将SDA线拉低并在SCL为高电平时保持低，拉高表示不应答；
4. 数据传送：主机传送一个8位的数据（DATA）；
5. 应答信号：主机释放SDA线等待从机的应答信号（ACK）；如此循环往复；
6. 停止：时钟线SCL为高时，数据线SDA由低到高。

上图中，R/W位为0时表示写，为1时表示读。主机向从机写入一个字节时，起始信号后传送寄存器地址，收到应答信号后开始传送寄存器数据，仍然需要收到应答信号后主机发送停止信号；读一个字节时，首先主机产生起始信号，然后发送从机地址位及一个写数据位，收到应答后发送寄存器地址，仍然需要收到应答信号后主机再发送一个起始信号，然后发送从机地址位及一个读数据位，从机产生应答信号并开始发送寄存器中的数据，通信以主机产生的拒绝应答信号（nACK）及停止信号作为结束。

**2.3 EEPROM模块**

EEPROM作为存储器件，通常用于保存数据，防止掉电之后数据的丢失，可以作为从机通过I2C总线与主机相连，实现二者间的数据交互。

EEPROM的写数据流程：首先写I2C的起始信号，接着写入器件地址，并在读写方向上选择“写”操作；第二步，发送数据的存储地址；第三步，发送待存储的数据。写数据过程中，EEPROM每个字节都会回应一个应答位“0”，用于指示写入数据成功，若无应答位则说明写入数据失败。

EEPROM的读数据流程：首先写I2C的起始信号，接着写入器件地址，并在读写方向上选择“写”操作；第二步，发送待读取的数据地址；第三步，重新发送I2C的起始信号和器件地址，并在读写方向上选择“读”操作；第四步，读取从器件发回的数据，若还想继续读，就发送一个应答位“0”，否则发送非应答位“1”。

从上述分析可以看出，EEPROM可与I2C总线配合完成数据的写/读，而且可以实现双向传输，需注意的是，无论是读操作还是写操作，时钟线SCL始终由主机控制，在写数据时应答信号由从机给出，表示从机是否正确接收数据；读数据时应答信号则由主机给出，表示是否继续读下去。本次实验中的EEPROM为一个8192位的8bit存储器，通过一个13bit的地址进行索引，故每次读写都要通过两个8bit的数据来指明13bit的地址。

**3 测试平台设计**

针对上述DUT的分析，本次实验搭建了一套测试平台用于实现其验证，结构如图3-1所示，主要模块包括待验证DUT、Interface、SRC\_AGENT、SCOREBOARD、ASSERTION以及COVERAGE，本节将对这些模块进行介绍。

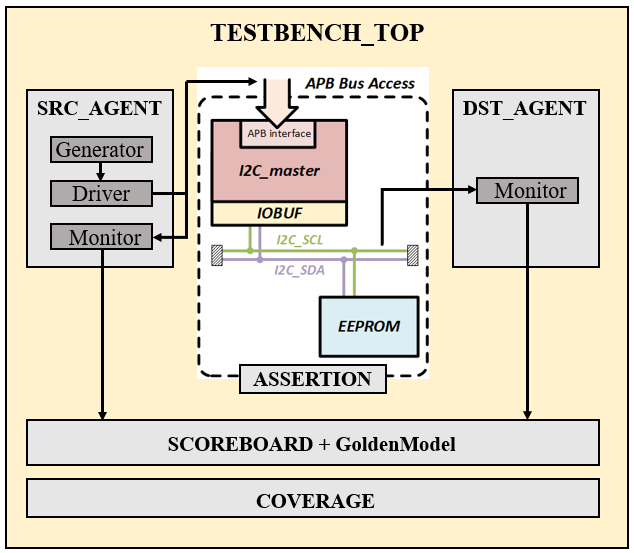


图3-1 测试平台框架

**3.1 常量定义**

首先对工程中需用到的常量进行定义，以便后续复用。测试平台构建过程主要用到的常量如图3-2所示，这些常量包括I2C总线寄存器地址，起始/结束/写/读/初始状态，开始/结束读、开始/结束写状态，使能/中断信号以及状态寄存器信号。提前对这些常量进行定义，后续构建模块时可直接复用，避免出现错误。

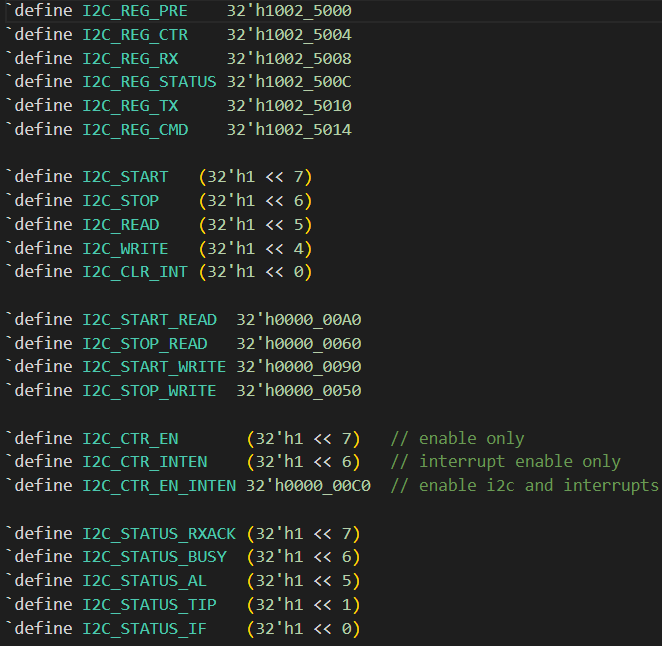


图3-2 实验过程使用到的常量

**3.2 Interface接口定义**

接口提供了一种将多个信号组合在一起表示为单个端口的方法，接口中的modport则对模块内的接口访问进行限制。本实验中需使用接口的模块包括src\_agent、i2c\_eeprom、scoreboard、assertion以及coverage，因此通过modport对上述模块可访问的端口进行了分类，如图3-3所示。

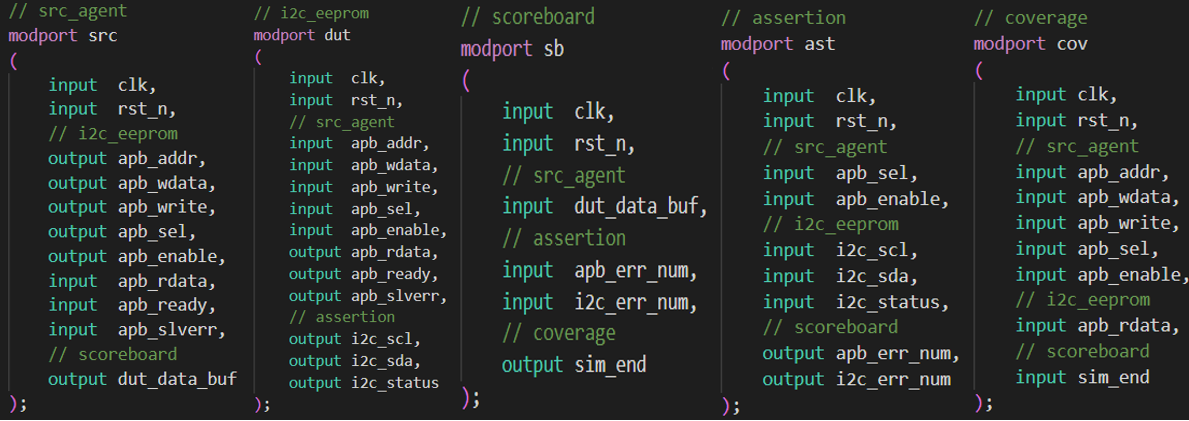
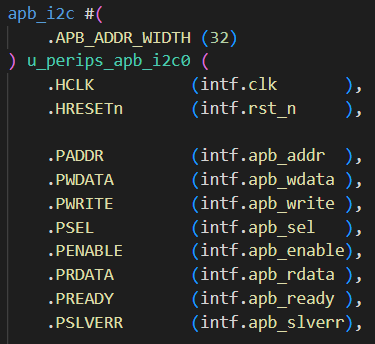


图3-3 各模块接口分类

**3.3 DUT顶层端口修改**

由于使用了interface对信号进行组合分类，因此待验证sim环境顶层模块的（i2c\_eeprom\_sim）端口连接需要修改为interface连接的端口，同时拉了一些内部信号出来用于后续验证，具体修改过程如图3-4所示。



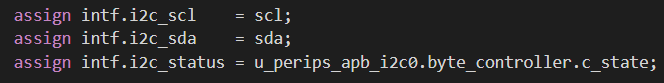


图3-4 DUT顶层模块修改

**3.4 SRC\_AGENT驱动设计**

这一模块的主要作用是驱动DUT完成写/读操作，驱动的实现经历了三个过程：首先进行数据的产生（Generator），其次施加激励（Driver），最后进行数据的观测（Monitor），接下来分别介绍各部分包含的要素及功能。

**3.4.1 Generator**

为产生数据，定义了三个Queue：bit [12:0] eeprom\_addr [$]用于存储/读出地址；bit [7:0] eeprom\_wdata [$]用于存储写入的数据；bit [7:0] eeprom\_rdata [$]用于进行数据的读出。接着进行了类的定义，该类中包含一个task，如图3-5所示。

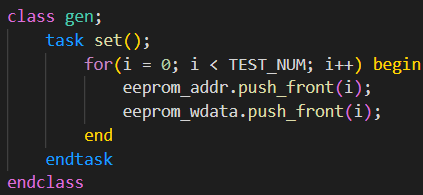


图3-5 Generator对象要素

如上图所示，gen类中含有一个task，其主要实现将256~0的数据依次存储在eeprom\_addr [$]和eeprom\_wdata [$]中，借助Queue中push\_front的方法产生数据并进行存储。

**3.4.2 Driver**

这一部分用于实现对APB接口的驱动，首先定义了一个基类，其中包含对信号进行初始化的task以及写/读的task，如图3-6所示。

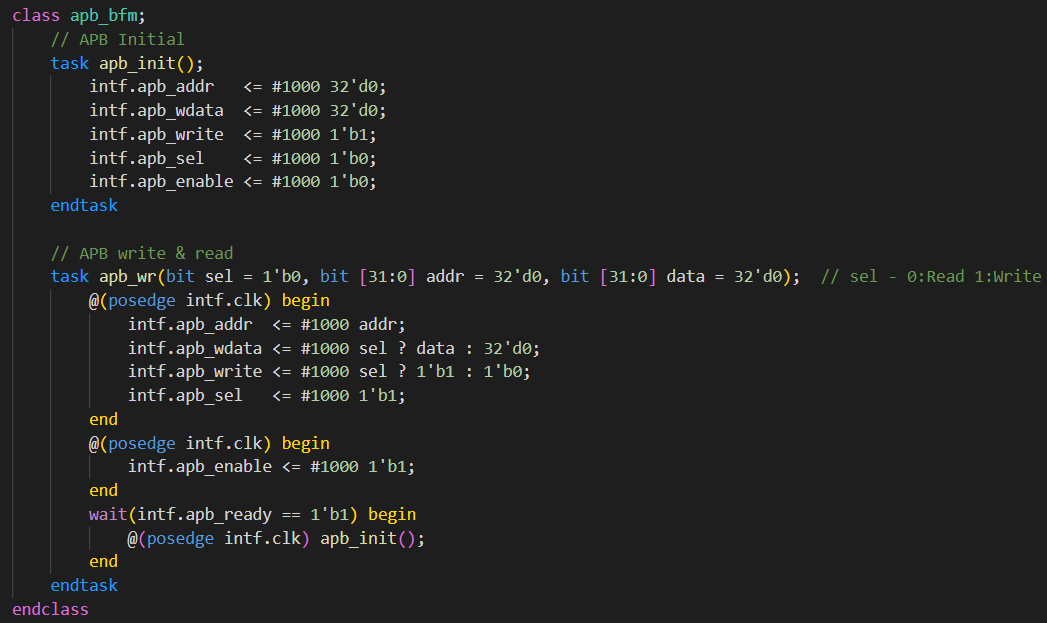
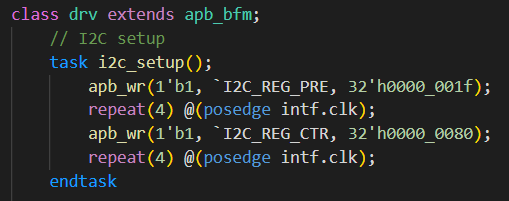
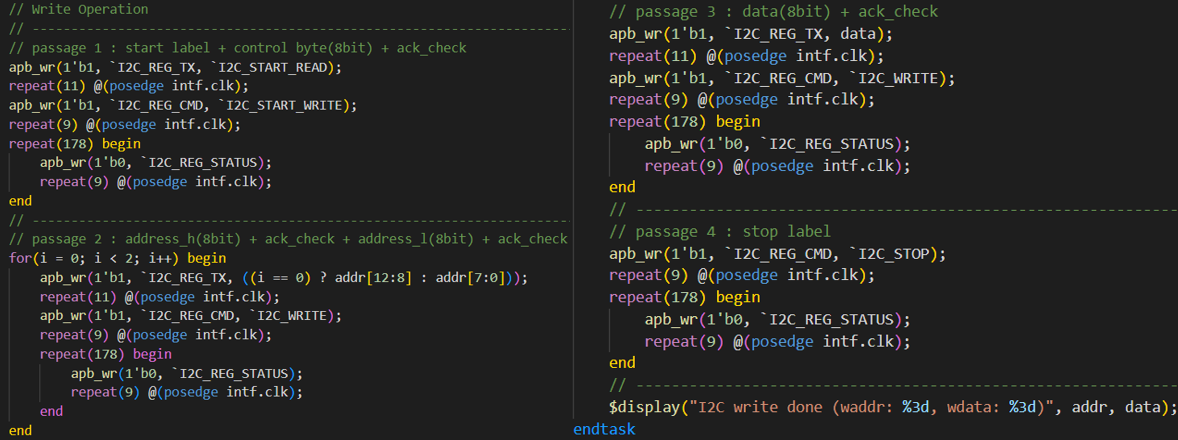


图3-6 Driver基类定义

如上图所示，apb\_init用于对APB接口的输入信号进行初始化，设置初始地址、写入数据为0，apb\_write拉高表示写，初始sel及enable信号均拉低。apb\_wr用于配置输入APB接口的数据及信号：输入地址即为写入APB的地址；当sel信号拉高时，写入数据即为输入数据，否则写入数据为0；当sel拉高时，apb\_write信号为拉高，表示写，否则apb\_write信号拉低，表示读；在每个时钟周期上升沿sel和enable信号均拉高，当检测到ready信号为1时，对数据进行初始化，拉低sel和enable信号。

在这一基类的基础上，结合I2C总线的读写规范，定义了一个派生类，其中包含对I2C总线进行初始化的task、I2C模块向从机写数据的task和I2C模块从外部从机读数据的task，如图3-7所示。





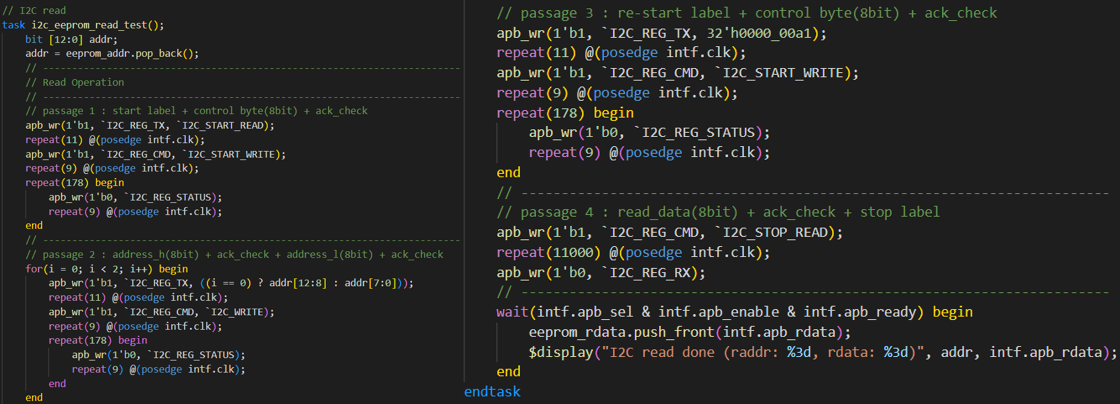


图3-7 Driver派生类定义

如上图所示，i2c\_setup主要功能在于初始化I2C模块的序列：首先配置I2C\_PRE寄存器来设置所需的SCL时钟频率，然后向I2C\_CTR中写入一个值（0x80）来使能I2C的功能。

i2c\_eeprom\_write\_test主要功能在于通过I2C模块向从机写入数据：第一步，“开始标志”，控制信号及“应答”；第二步，从机地址高位，“应答”，从机地址低位，“应答”；第三步，写数据，“应答”；第四步，“停止标志”。

i2c\_eeprom\_read\_test主要功能在于通过I2C模块从外部从机读取数据：第一步，“开始标志”，控制信号及“应答”；第二步，从机地址高位，“应答”，从机地址低位，“应答”；第三步，“开始标志”，控制信号及“应答”；第四步，读取数据，“应答”及“停止标志”。

**3.4.3 Monitor**

这一部分用于对读出数据进行采集以便实时监控DUT的变化，定义了一个类，其中包含一个task对数据进行采集，如图3-8所示。

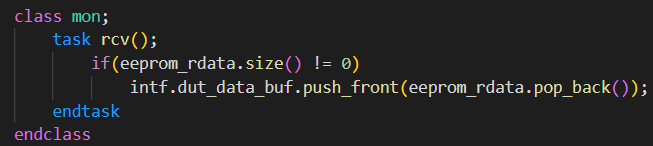


图3-8 Monitor类定义

如上图所示，在类mon中定义的rcv对读出数据进行收集，DUT的变化都可以在这一部分被采集并输出，从而便于检验。

**3.5 SCOREBOARD功能验证设计**

这一模块主要是在monitor的基础上设计scoreboard配合Golden Model完成验证，其中Golden Model用于模拟DUT的功能，对DUT的结果进行预测，并把结果送入scoreboard与DUT的结果进行比较。由于在上一模块中数据的产生是通过Queue的push\_front操作，因此使用Golden Model模拟这一操作，如图3-9所示。

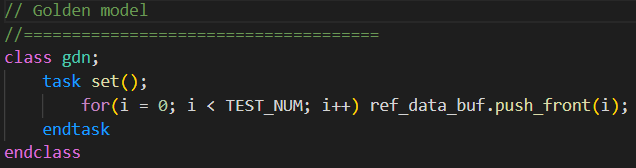


图3-9 Golden Model定义

如上图所示，定义bit [7:0] ref\_data\_buf [$]，通过push\_front操作产生数据，与src\_agent中的数据产生方式相同，从而可实现对数据的观测与比较。

对这一模块的例化是通过将从DUT中读出的数据与参照数据进行对比，相同则表示比对正确，不同则错误数加一，最终将测试次数与错误次数进行输出，以便观测数据的产生与读出是否正确，该模块还连接了assertion模块，用于指示断言的失败次数。

**3.6 ASSERTION断言设计**

断言（assertion）被放在verilog设计中，方便在仿真时查看异常情况。本实验中设计了两个断言，一个用于验证APB接口的时序正确性，另一个用于验证I2C时序正确性及总线内部的状态转换，如图3-10所示。

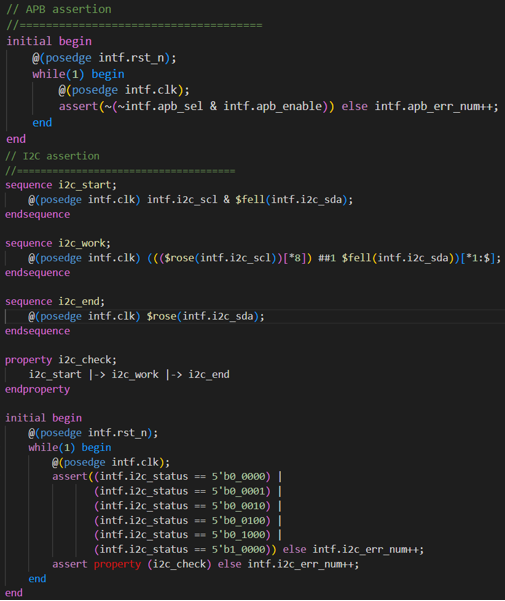


图3-10 断言设计

如上图所示，APB的数据传输状态需sel与enable同时拉高，对APB的这一时序特性进行断言，若通过则表明其可以正常实现数据传输。对I2C总线的断言设计主要关注其时序特性及内部的状态转换，从图3-4中可以看出上图中的i2c\_scl和i2c\_sda即scl和sda线，i2c\_status即I2C\_master中的c\_state，断言I2C总线的时序状态（开始到工作到结束）以及可以实现所有状态的跳转。

**3.7 COVERAGE覆盖率设计**

覆盖率用于衡量设计中已测部分与未测部分的比例，可分为代码覆盖率及功能覆盖率，本实验中主要对I2C内部的寄存器进行覆盖率统计，由表2-1可知其内部主要包含六个寄存器：I2C\_REG\_PRE、I2C\_REG\_CTR、I2C\_REG\_RX、I2C\_REG\_STATUS、I2C\_REG\_TX以及I2C\_REG\_CMD，对仿真过程被使用到的寄存器进行功能覆盖率统计，如图3-11所示。

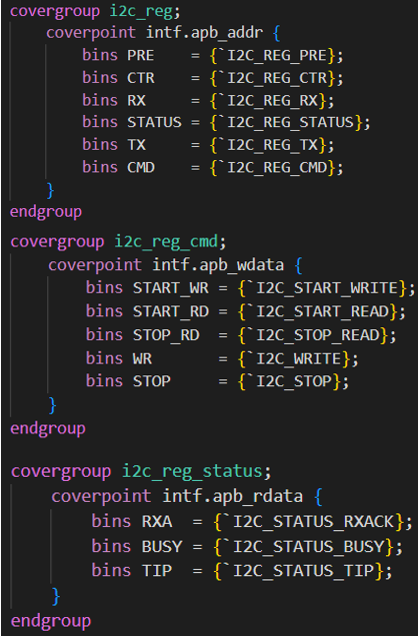
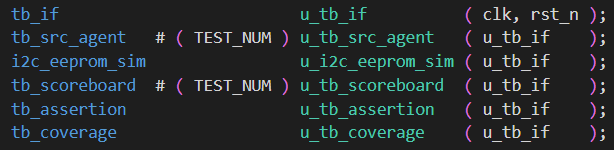


图3-11 覆盖率设计

如上图，第一个covergroup用于统计使用到的寄存器，由于PRE、TX、RX和CTR寄存器只用到了一个功能，故寄存器使用到了就表示其功能被使用；而CMD和STATUS寄存器区分了子功能，需进一步统计各自的功能覆盖率，下面的两个covergroup用于这两个寄存器功能覆盖率的统计。

**3.8 TOP顶层例化**

TestBench的顶层主要实现对各个模块的例化，并定义时钟及复位信号，如图3-12所示。



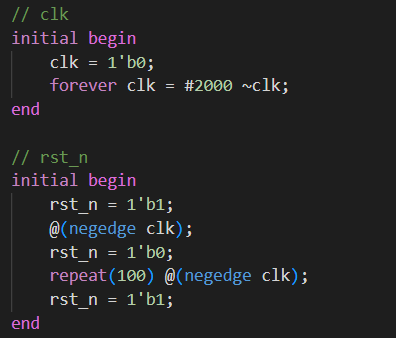


图3-12 顶层例化及时钟定义

**4 工程仿真结果**

将待验证sim环境与上一节构建的测试平台导入编译器中进行仿真，观察APB能否被正确驱动（读/写）、scoreboard对比情况、assertion断言通过情况、覆盖率以及时序波形。

**4.1 数据读写测试结果**

本实验中写入的地址及数据为0~255依次写入，相应的，读出数据也应是0~255依次读出，写入及读出结果如图4-1所示（部分数据在图中省略）。

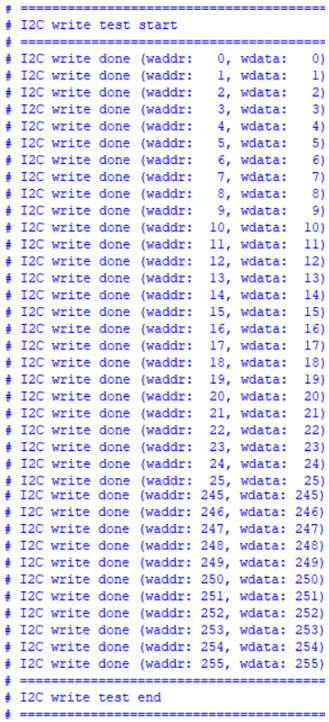
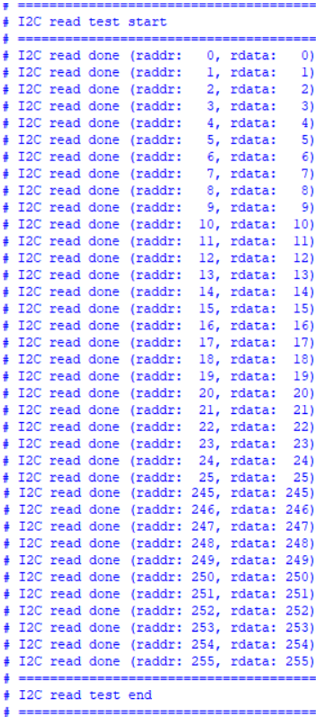
 

图4-1 读写测试结果

从上面两幅图可以看出写入数据及地址从0~255依次变化，写入及读出数据结果保持一致，说明对于该DUT的写/读操作被正确实现。

**4.2 SCOREBOARD测试结果**

本实验在scoreboard中构建了golden model模拟数据的产生并将读出数据与从DUT中观测到的读出数据进行比对，若不同则错误数加一，该模块还连接了assertion断言模块用于统计断言的错误次数，测试的打印结果如图4-2。

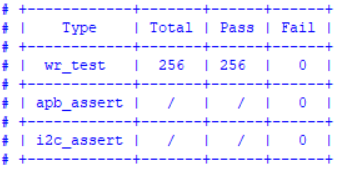


图4-2 scoreboard测试结果

如上图所示，数据从APB传到I2C总线再传输到EEPROM，对于从EEPROM读出的数据进行采集并与golden model模拟的数据进行对比，可以看到共进行了256次比对测试，错误数为0，说明数据传输过程没有问题。该比对中还加入了对断言错误次数的测试，设计了两部分断言，分别针对APB时序、I2C时序及其内部的状态转换，可以看到两部分断言的错误次数均为0，说明全部通过。

**4.3 COVERAGE测试结果**

本实验对I2C内部的寄存器进行覆盖率的统计，反映了其内部寄存器的功能覆盖率，在编译器中可以看到覆盖率结果如图4-3。

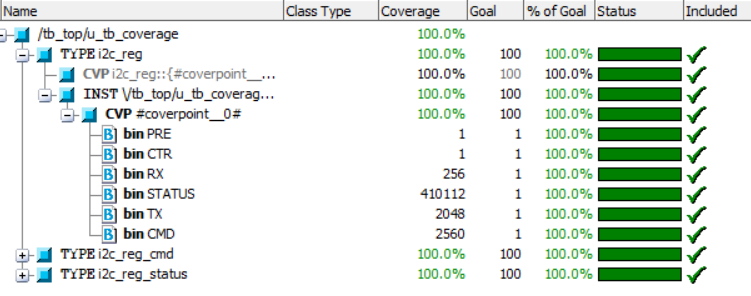
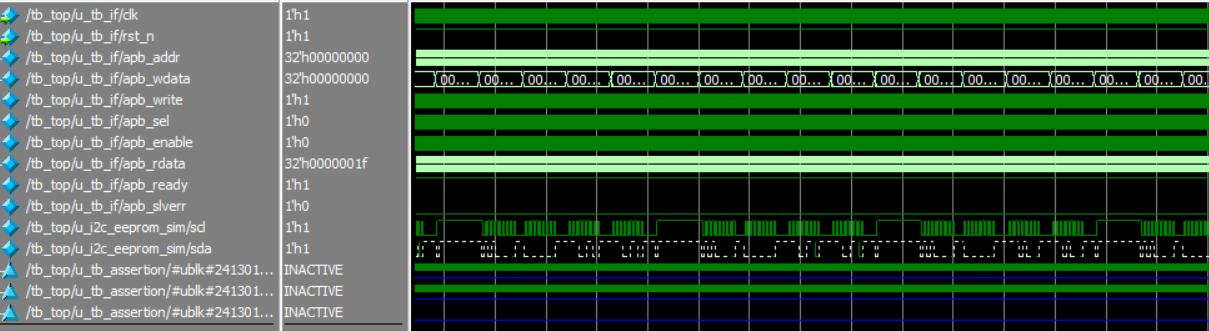


图4-3 覆盖率测试结果

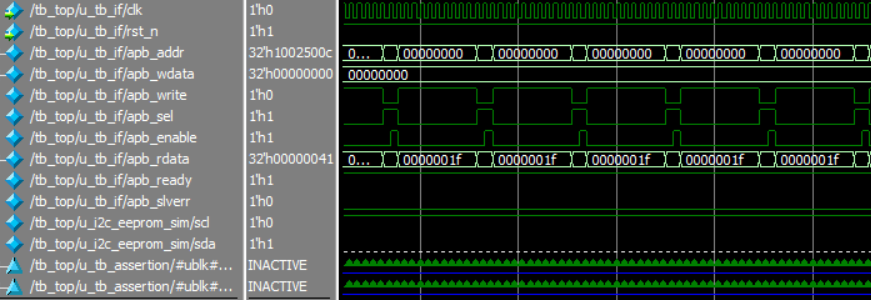
从上图可以看出I2C内部的六个寄存器在仿真过程中全部被使用，对于只有一个功能的PRE、CTR、TX和RX寄存器而言功能覆盖率为100%，而具有子功能的CMD和STATUS寄存器的功能覆盖率也达到了100%，满足了测试要求。

**4.4 时序波形结果**

本实验中DUT具有APB接口，需验证其时序波形是否正确，仿真时序波形结果如图4-4（a）、（b）所示。



（a） 时序整体波形



（b） 时序局部放大波形

图4-4 时序波形输出结果

从上图可以看出，该测试平台完成了对DUT的写入及读出，当apb\_write拉低时表示读，apb\_sel信号首先拉高进入SETUP状态，然后apb\_enable拉高进入ACCESS状态进行数据的读出，当检测到apb\_ready信号为高表示数据传输完成，apb\_sel与apb\_enable信号拉低，输出的时序波形图正确的反应了这一过程。

**5 工作总结与感悟**

综上所述，在本次测试平台架构的实验中我主要完成的内容有以下几个部分：

1. **驱动设计**：通过interface连接测试平台及DUT，并构建了SRC\_AGENT对DUT进行驱动，主要包括Generator产生数据、Driver施加激励以及Monitor采集数据，进行了256次写入及读出操作，数据从APB传到I2C总线再传输到EEPROM，对最终从EEPROM读出的数据进行采集。
2. **功能验证设计：**在monitor的基础上设计了SCOREBOARD模拟数据的产生，并与采集到的数据进行比较以观测数据传输过程有没有出现错误。对256个写入数据进行了对比测试，错误次数为0。
3. **断言设计：**对于APB接口的时序特性、I2C总线的时序特性以及I2C内部的状态转换进行了断言，结果全部通过。
4. **覆盖率设计：**对I2C内部使用到的寄存器统计了功能覆盖率，只有一个功能的寄存器被使用到即功能被覆盖，具有子功能的寄存器则单独统计了各个功能的覆盖率，最终结果为100%。

**工作感悟：**在本次实验的过程中，我理解了APB接口协议、I2C总线的工作原理及时序波形，并在以往lab的基础上尝试自己一点点搭建测试平台去验证DUT，我学习了用SystemVerilog语言构建每个模块并最终汇总为一个整体，从不同方面测试DUT的功能正确性。本次project对我来说是一项重大挑战，这个过程中经历了许多迷茫与困惑，但坚持下来会发现收获颇丰，这是一次书本知识与实践能力的融会贯通，极大的锻炼了我的能力，也增强了我对这门课的认知与体会。这次实验有些地方还有待完善，我会在以后的学习及研究生活中进一步改善与精进。

最后想感谢任课老师蒋老师的耐心教学与指导，感谢助教们的辛苦付出与及时的答疑解惑，感谢一起互相讨论交流的小伙伴们，大家的帮助促进了我对这门课的理解与掌握，今后我会更加努力，不辜负老师和师兄们的付出。