1. **系统方案设计**

在SRAM测设的方案设计中，采用软硬件结合的方法，在上位机中利用随机数，定向测试，以及特定bist算法生成测试向量，通过以太网或者UART进行通信，传入下位机FPGA中，FPGA根据收到的测试命令，进行相应的解析，对被测试的SRAM传入相对应的使能信号，地址信号以及数据信号，同时接收SRAM反馈回来的信息，进行处理后通过通信传回上位机，上位机得到相应的反馈信息，根据信息生成相应的日志文件，当测试完成后，上位机根据其传回的日志文件以及传入的测试向量进行比较，从而判断SRAM的功能正确性。其数据流通路如下图所示：



对于结构设计，首先上位机通过UART或者Ethernet进行通信，从memory中获得测试向量，发送到下位机，下位机根据通信进行解包，同时通过CPU的解析，发送给总线的IP相应寄存器进行赋值，使能IP完成命令，IP根据不同的配置产生不同的操作，进而控制SRAM，其系统的结构如图所示：



**二、模块设计**

**1、测试向量生成**

在上位机生成测试向量，下位机接收测试向量，同时根据向量的命令进行相应的操作，其命令格式如表所示：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Cmd | Parameter1 | Parameter2 | Parameter3 | Parameter4 | Parameter5 | Parameter6 | 备注 |
| 1 | addr | data | - | - | - | - | Wrt\_one |
| 2 | addr | - | - | - | - | - | Read |
| 3 | data | - | - | - | - | - | Write\_all |
| 4 | addr | - | - | - | - | - | Read\_d |
| 5 | addr | data | op\_area | inc\_dec | cycle | jump | Wrt\_cfg |
| 6 | addr | area |  |  |  |  | Update |
| 7 | cut\_cnt |  |  |  |  |  | Cut\_cnt |

对于每条命令，当发送的命令序号为1时，控制器对发送的特定地址写入特定的数据；命令序号为2时，控制器会从自己内部的与SRAM地址相对应的寄存器中读出数据；命令序号为3时，控制器会对SRAM所有地址写入相同的数据；命令序号为4时，控制器从SRAM中读出数据，同时直接发送给上位机；命令序号为5时，控制器根据接收到的起始地址以及操作范围，同时根据地址递增或递减，走到边界地址是否回环，以及地址变化是否有间隔跳跃等配置，对几个不同的地址写入不同的数据，如发送2 90 6 1 1 1的命令，则会从2地址开始，向下递减，同时回环，跳跃一个单元，范围为6，即对2，0，3fe，3fc地址写入5a的数据；当命令序号为6时，控制器根据起始地址以及更新的范围，将SRAM中几个连续地址的数据读出到控制器中与SRAM地址相对应的寄存器中去，利用命令序号为2的命令读出，命令序号为7则端开连接。

对于测试向量的种类，主要分为以下5种：

A、定向测试

对所有地址写入同一数据00，ff,5a,a5，并依次读出，判断数据的正确性

B、March-SS算法

其测试方法如下:

|  |  |
| --- | --- |
| 命令 | 备注 |
| ||(w0) | 对所有地址写0 |
| up(r0,r0,w0,r0,w1) | 地址递增，每个单元读0读0写0读0写1 |
| up(r1,r1,w1,r1,w0) | 地址递增，每个单元读1读1写1读1写0 |
| down(r0,r0,w0,r0,w1) | 地址递减，每个单元读0读0写0读0写1 |
| down(r1,r1,w1,r1,w0) | 地址递减，每个单元读1读1写1读1写0 |
| ||(r0) | 对所有地址读0 |

C、边界测试

对地址为0，1，2，3fd，3fe，3ff的边界地址进行分别写入0，ff,5a,a5进行测试，同时读出该地址以及相邻单元的数据，对比发现数据是否有错误产生

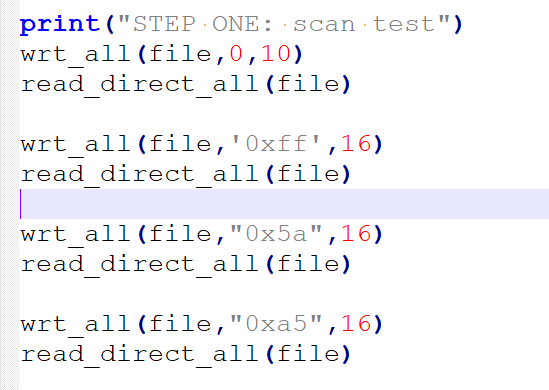
D、随机测试

在地址范围内随机生成地址，以及在数据范围内随机生成数据，对特定地址写入特定数据，同时对该地址以及其左右相邻地址进行读出，发现是否产生错误

E、置信测试

首先在数据范围内生成一个随机数，对所有地址写入这个随机数，然后根据生成特定数量个不同的随机地址，对这些随机地址进行读出，对比数据是否正确

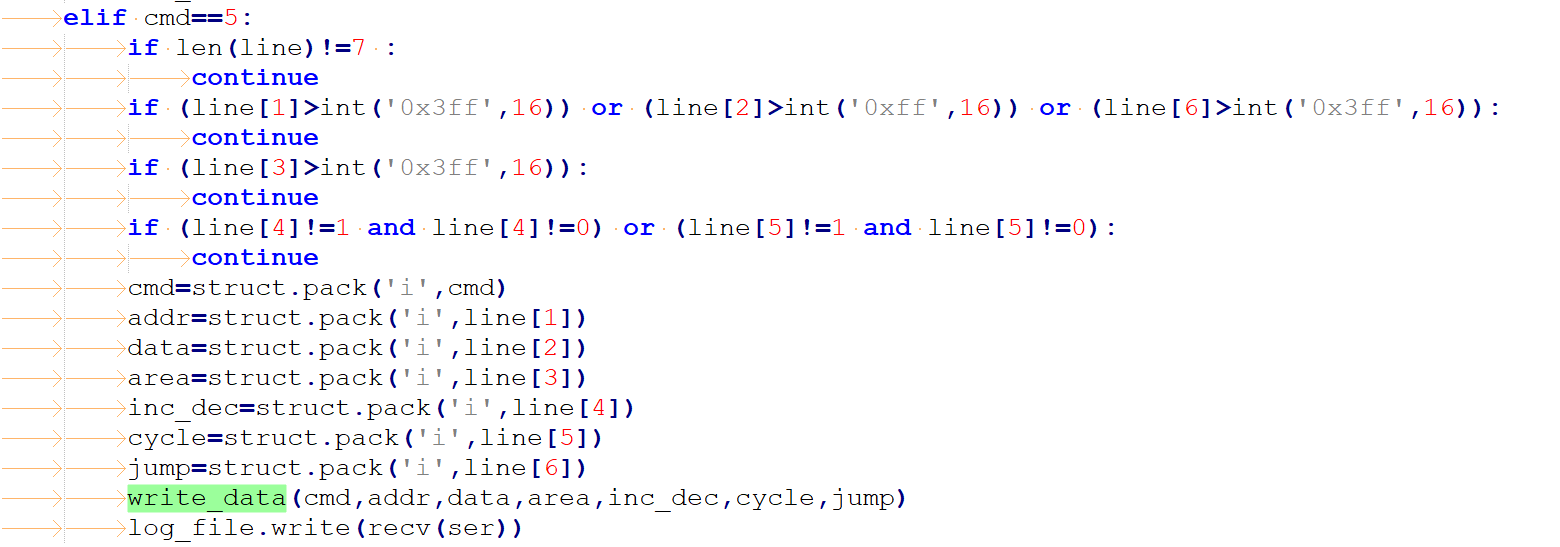
将生成的测试向量写入文本种，等待发送，其代码大致如下所示：



**2、通信方式**

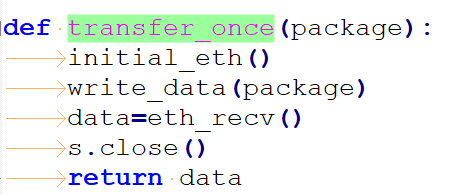
A、UART通信

使用UART进行外设通信，上位机将数据进行字节打包，然后根据根据不同的字节数目以及命令的不同，在下位机有不同的解包形式，根据命令的不同，对IP产生不同的控制，IP根据配置的情况的不同，对SRAM产生不同的操作，其通信代码大致如下：



B、以太网通信

根据以太网的格式进行通信，在后面的数据段种将命令进行打包，以字节的形式进行传输，在下位机根据int，double，char等字节数目的不同，首先解析出命令，然后根据命令即可获得有效数据包的长度，根据长度进行解包，获得相应的命令信息，从而控制IP，其代码大致如下：



对于通信，可以根据生成的测试向量进行自动通信，也可进行手动输入命令以及相应的参数进行自定义的命令的传送，从而控制IP以及SRAM。

**3、IP设计**

CPU根据通信外设获得相应的数据写入到IP的寄存器中，IP在系统中需要挂在在总线上CPU方可进行相应的控制，在IP设计中，为方便挂在系统，将总线模块与功能拆开，便于复用。在设计中共实现了AXI与AVALON两种总线接口进行设计，其结构如下所示：



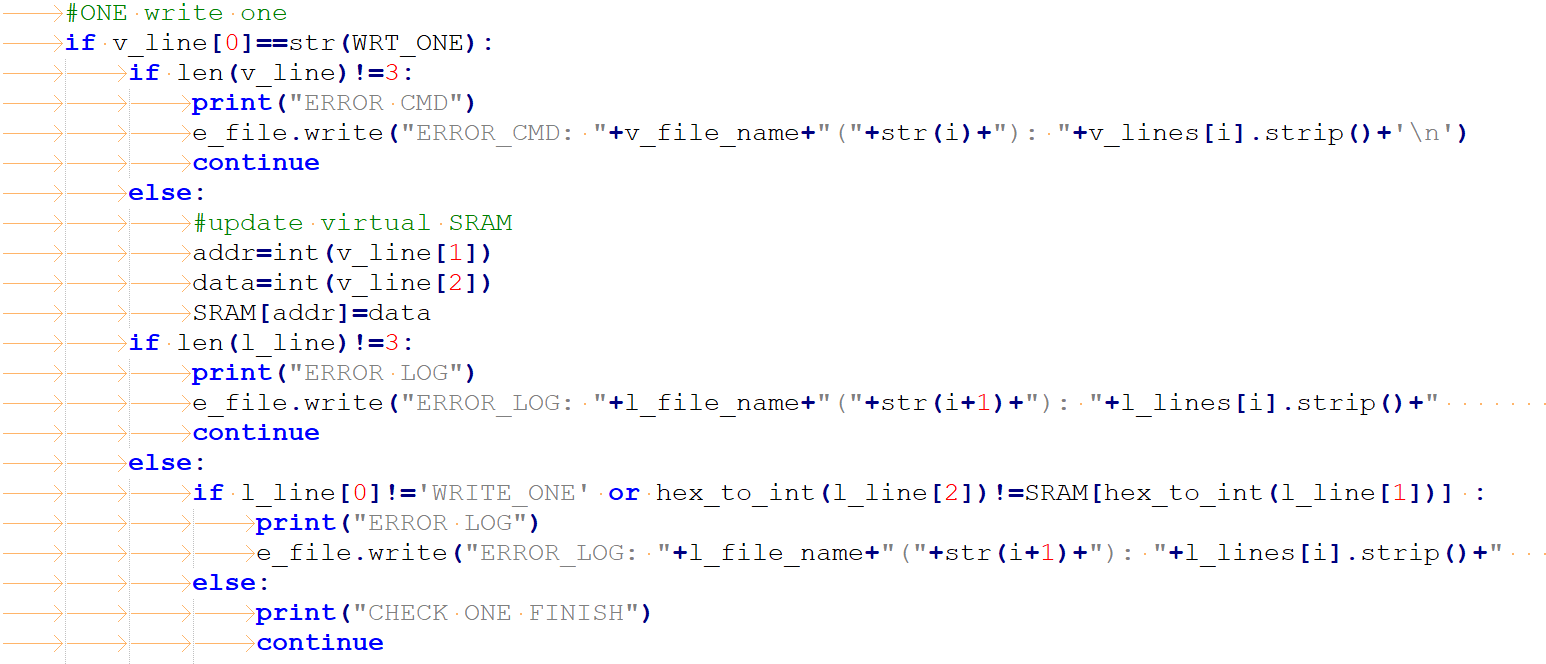
其IP寄存器如下所示：

|  |  |  |
| --- | --- | --- |
| Name | Function | I/O |
| Sta\_addr | 输入起始地址，写入或更新 | Input |
| Area\_cfg | 写操作以及更新操作范围，jump为0，即为写操作次数+1 | Input |
| Op\_cfg | [2]:直接读出 [1]:配置递增递减 [0]:是否回环  [31:22] jump步长 | Input |
| Send | 要写入的数据或读出的地址 | Input |
| Enable | [0]配置使能[1]配置读写0写1读 | Input |
| Outp\_addr | 输出地址 | Output |
| Outp\_data | 输出数据 | Output |
| Status | [7:0]当前状态 | Output |

根据总线的不同偏移地址情况不同，但几个寄存器之间的相对偏移顺序相同。

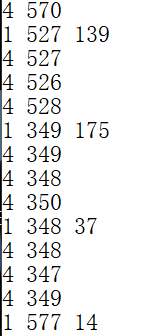
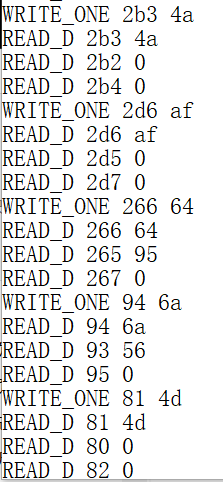
**3、检错方式**

FPGA中的IP不会对数据进行进一步的检错，只会检查是否溢出等极端错误情况，然后把数据以及地址信息反馈回上位机，在上位机进行相应的检错操作，上位机根据反馈信息，生成相应的log文件，上位机首先虚拟一个SRAM，根据发送出去的测试向量，进行与待测SRAM相同的操作，然后根据反馈回来的含有真实SRAM信息的日志文件，与虚拟SRAM的完全正确的情况进行对比，发现真实SRAM是否与虚拟SRAM的黄金模型有所出入，从而进行检错，其代码大致如下：

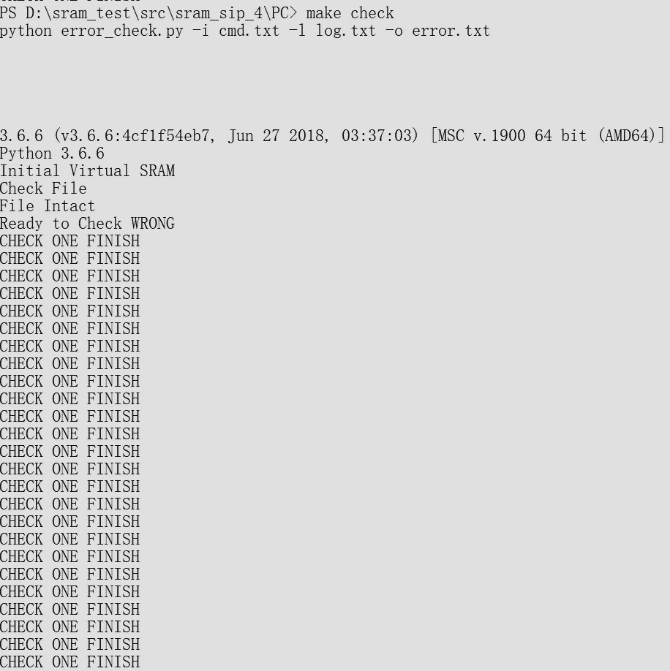


**三、测试结果**

对SRAM在1M-60M不同的频率的功能进行测试，其测试命令与log日志如下所示：

经过实测，SRAM在1M-50M作用下，功能正确，无错误产生，其检测结果如下：



在60M的作用下，由于外接线的问题，使得波形不稳定，造成了少量的数据错误，其错误结果如下：

