**实验四 仿真激励文件设计**

一、实验目的：

（1）掌握仿真激励文件的基本编写规则和仿真原理；

（2）学习仿真激励文件的设计方法；

（3）分析基本模型与实际工程的区别。

二、实验背景：

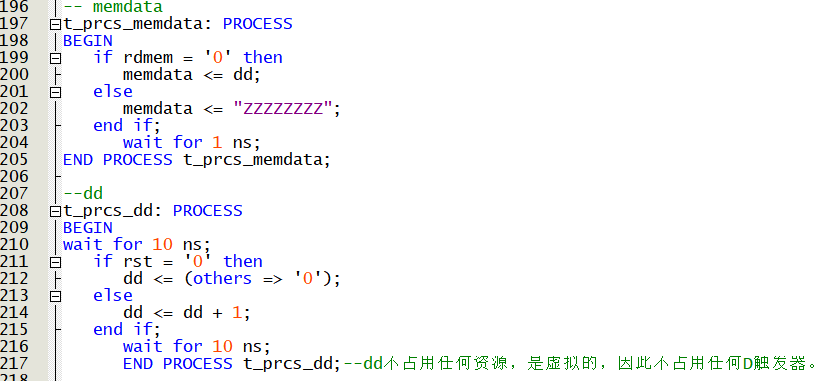
在上个实验中，同学们学习了波形仿真的基本方法，了解了波形仿真时，软件其实先将波形转换为了VHDL语言或者Verilog语言，然后基于语言进行语法分析和仿真。因此本次实验让学生先学习仿真激励的优势，再在开发板上下载和分析工程，从而掌握模型设计与实际使用工程的基本区别、联系及应用。

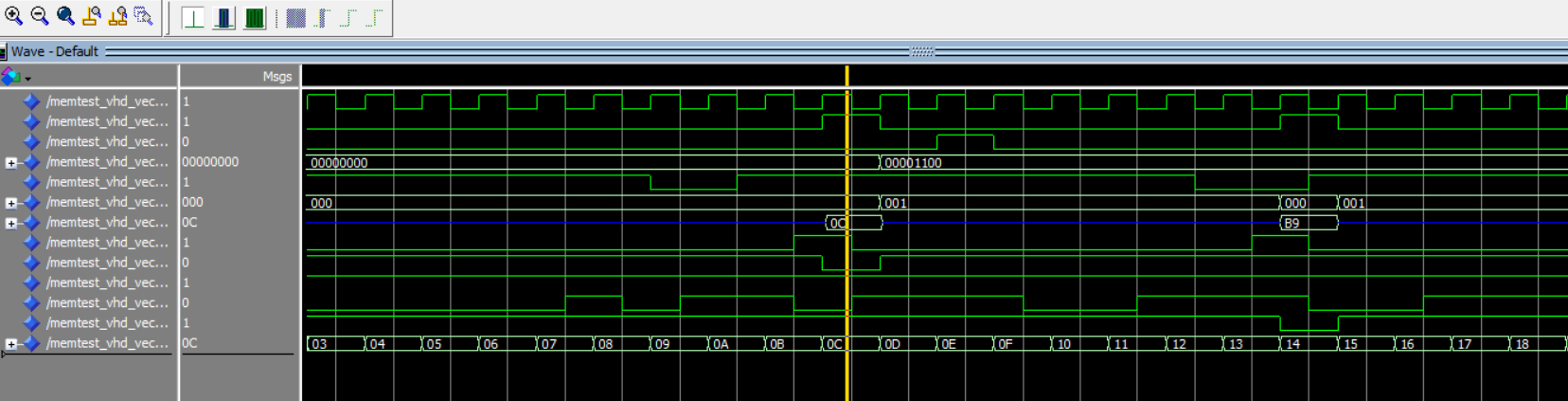
三、实验内容：

学习使用VHDL代码进行激励文件的编写和仿真。

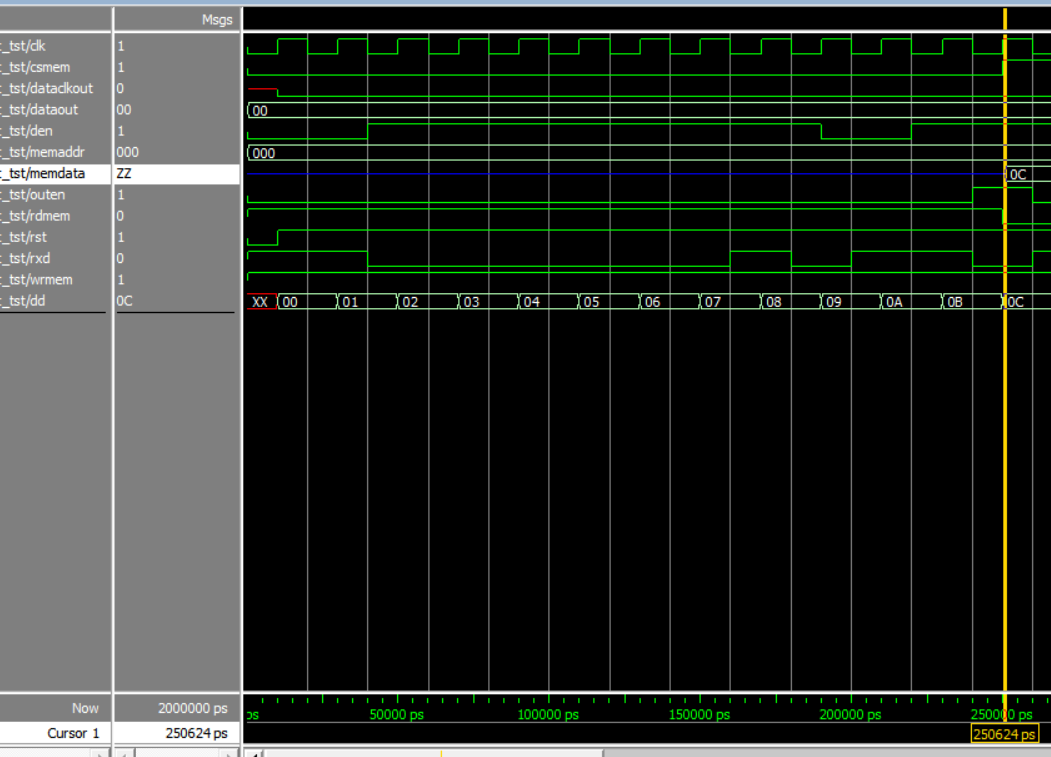
1. 实验步骤：
2. 将memdata激励代码进行优化

由于memdata实际为一个整体，因此在激励文件设计中应当将他作为整体进行赋值操作，同时，memdata应当在rdmem为0时进行赋值，其他时间为高阻态。并且我们希望memdata每次赋值应当不同，只要我们设置额外的变量dd，让memdata和dd保持变化即可。完整代码如下：





可以看到dd已经随时间变换，并且当rdmem为0时，对memdata赋值。需要注意的是，memdata并不是在时钟上升沿来临后立刻变化和dd保持一致，而是需要经过一定的延时。此外由于激励仿真仅是方阵，并没有对实际电路器件的使用，因此dd占用的触发器个数为0。



1. 按键防抖、串口代码等部分。

按键防抖首先有一个移位寄存器，每毫秒移位一次，只有当连续10个毫秒全部按压按键之后才认为按下按键，否则认为是出现了抖动，不进行操作。要完成以上目标要存在分频器将系统时钟分频道毫秒和秒级别。另外还存在4-7译码器驱动数码管，数码管为低电平有效，即输入为低电平时才点亮。

由于实际的串行输入并没有时钟，所以需要通信双方约定好数据传输的速率等，达成相同的物理协议之后才能进行通讯。

1. 思考题
2. 尝试修改激励代码，按自己的想法，仿真更复杂的输入激励情况。

设计模任意值计数器，当计时器输出为1时，才对memdata进行赋值操作，否则不变。

1. 写出自己设计的按键防抖动代码，或者仿真网上查到的按键防抖动代码并仿真分析；

思路：时钟周期为0.01ms，当检测到按键被按下之后（某个变量的高低电平发生变化），count开始计数，count到达5000（即存在一段50ms的按键按下状态）才认为按键被按下，否则认为是抖动。

LIBRARY IEEE;  
USE IEEE.STD\_LOGIC\_1164.ALL;  
USE IEEE.STD\_LOGIC\_ARITH.ALL;  
USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;  
ENTITY xiaodou IS  
  PORT(

pressin:in std\_logic;--按键  
  clk:in std\_logic--时钟频率，此处pin连接clk2，设基准频率为100kHZ  
  );  
  end entity;  
  ARCHITECTURE behav OF xiaodou IS  
  signal count:integer;--计数  
  signal DIN1:std\_logic;--代表DIN按键，若为1，表示按键按下，若为0，表示按键未按下  
  begin  
  process(clk)  
  begin  
  if(clk'event and clk='1') then--上升沿  
 if(DIN='0') then--按键初始为1  
 count<=count+1;  
 if(count=5000) then--对应现实50ms  
 DIN1<='1';--判断是否有一段持续50ms的'0'状态，若有即代表按下  
 else  
 DIN1<='0';--若无代表没按下  
 end if;  
 else  
 count<=0;  
 end if;  
 end if;  
 end process;  
end behav;

1. 仿真下载工程中的memtest代码，分析与../step1/下memtest的区别和原因。

仿真得到的结果并不会出现rdmem和wrmem数据丢失的情况，而上次的memtest会出现丢失的状况。由于sram中memtest加入了两个状态，在完成读或者写状态后都会回到初态，并且记住读或写请求，所以每次读写请求不会丢失。

