**实验报告**

2019 年 5 月 16 日 成绩：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓名 | 王坤 | 学号 | 17051225 | 班级 | 17052312 |
| 专业 | 计算机科学与技术 | | 课程名称 | 计算机组成原理 | |
| 任课老师 | 章复嘉 | 指导老师 | 章复嘉 | 机位号 |  |
| 实验序号 | 实验5 | 实验名称 | 存储器实验 | | |
| 实验时间 |  | 实验地点 | 1-225 | 实验设备号 |  |
| **一、实验目的与要求** | | | | | |
| 1. 实验目的：    1. 学习MIPS指令系统，熟悉MIPS指令格式及其汇编指令助记符，掌握机器指令编码方法；    2. 学习MIPS汇编程序设计，学会使用MIPS汇编器将汇编语言程序翻译成二进制文件；   了解使用MIPS教学系统模拟器运行程序的方法   1. 实验要求：   按照上述实验内容中的例子，编辑test.asm文件，在PCSpim/MARS中打开它，并使用单步执行和连续执行方式运行该程序；  将下列程序输入一个R\_CPU\_Test.asm文件，并在PCSpim /MARS中单步运行，观察各个寄存器的值，是否和预期的一致 | | | | | |
| **二、实验设计与程序代码** | | | | | |
| 1. 模块设计说明   实验只使用了一个单模块,包含ip核模块  程序直接把开关的低6位作为地址  由于这是两个实验，第一个有写入功能，第二个则要求是只读。本实验报告按照第二个实验来写。实验的源代码没有提供写入的代码段，只有读出的代码段   1. 实验程序源代码及注释等   **`timescale 1ns / 1ps**  **module testmodule(Data\_Input,Data\_Output,SW);**  **input [32:1] Data\_Input;**  **input [6:1] SW;**  **output [32:1] Data\_Output;**  **reg [6:1] mem\_addr;**  **reg [32:1] Data\_Temp;**  **always@(SW[1])**  **begin**  **if(SW[3]) begin Data\_Temp<=Data\_Input; end**  **if(SW[2]) begin mem\_addr<=Data\_Input[6:1]; end**  **end**  **RAM\_B Data\_RAM (**  **.clka(SW[1]), // input clka**  **.wea(SW[6]), // input [0 : 0] wea**  **.addra(mem\_addr), // input [5 : 0] addra**  **.dina(Data\_Temp), // input [31 : 0] dina**  **.douta(Data\_Output) // output [31 : 0] douta**  **);**  **endmodule** | | | | | |
| **三、实验仿真** | | | | | |
| 1. 仿真代码   `timescale 1ns / 1ps  module testmodule(Data\_Input,Data\_Output,SW);  input [32:1] Data\_Input;  input [6:1] SW;  output [32:1] Data\_Output;  RAM\_B Data\_RAM (  .clka(SW[1]), // input clka    .addra(Data\_Input[6:1]), // input [5 : 0] addra    .douta(Data\_Output) // output [31 : 0] douta  );  endmodule   1. 仿真波形      1. 仿真结果分析   读取出来的数和COE文件里面的完全一致 | | | | | |
| **四、电路图** | | | | | |
|  | | | | | |
| **五、引脚配置（约束文件）** | | | | | |
| NET "Data\_Input[31]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[30]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[29]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[28]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[27]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[26]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[25]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[24]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[23]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[22]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[21]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[20]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[19]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[18]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[17]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[16]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[15]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[14]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[13]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[12]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[11]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[10]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[9]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[8]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[7]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[6]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[5]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[4]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[3]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[2]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[1]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[32]" IOSTANDARD = LVCMOS18;  NET "Data\_Input[32]" LOC = T3;  NET "Data\_Input[31]" LOC = U3;  NET "Data\_Input[30]" LOC = T4;  NET "Data\_Input[29]" LOC = V3;  NET "Data\_Input[28]" LOC = V4;  NET "Data\_Input[27]" LOC = W4;  NET "Data\_Input[26]" LOC = Y4;  NET "Data\_Input[25]" LOC = Y6;  NET "Data\_Input[24]" LOC = W7;  NET "Data\_Input[23]" LOC = Y8;  NET "Data\_Input[22]" LOC = Y7;  NET "Data\_Input[21]" LOC = T1;  NET "Data\_Input[20]" LOC = U1;  NET "Data\_Input[19]" LOC = U2;  NET "Data\_Input[18]" LOC = W1;  NET "Data\_Input[17]" LOC = W2;  NET "Data\_Input[16]" LOC = Y1;  NET "Data\_Input[15]" LOC = AA1;  NET "Data\_Input[14]" LOC = V2;  NET "Data\_Input[13]" LOC = Y2;  NET "Data\_Input[12]" LOC = AB1;  NET "Data\_Input[11]" LOC = AB2;  NET "Data\_Input[10]" LOC = AB3;  NET "Data\_Input[9]" LOC = AB5;  NET "Data\_Input[8]" LOC = AA6;  NET "Data\_Input[7]" LOC = R2;  NET "Data\_Input[6]" LOC = R3;  NET "Data\_Input[5]" LOC = T6;  NET "Data\_Input[4]" LOC = R6;  NET "Data\_Input[3]" LOC = U7;  NET "Data\_Input[2]" LOC = AB7;  NET "Data\_Input[1]" LOC = AB8;  NET "Data\_Input[31]" PULLDOWN;  NET "Data\_Input[30]" PULLDOWN;  NET "Data\_Input[29]" PULLDOWN;  NET "Data\_Input[28]" PULLDOWN;  NET "Data\_Input[27]" PULLDOWN;  NET "Data\_Input[26]" PULLDOWN;  NET "Data\_Input[25]" PULLDOWN;  NET "Data\_Input[24]" PULLDOWN;  NET "Data\_Input[23]" PULLDOWN;  NET "Data\_Input[22]" PULLDOWN;  NET "Data\_Input[21]" PULLDOWN;  NET "Data\_Input[20]" PULLDOWN;  NET "Data\_Input[19]" PULLDOWN;  NET "Data\_Input[18]" PULLDOWN;  NET "Data\_Input[17]" PULLDOWN;  NET "Data\_Input[16]" PULLDOWN;  NET "Data\_Input[15]" PULLDOWN;  NET "Data\_Input[14]" PULLDOWN;  NET "Data\_Input[13]" PULLDOWN;  NET "Data\_Input[12]" PULLDOWN;  NET "Data\_Input[11]" PULLDOWN;  NET "Data\_Input[10]" PULLDOWN;  NET "Data\_Input[9]" PULLDOWN;  NET "Data\_Input[8]" PULLDOWN;  NET "Data\_Input[7]" PULLDOWN;  NET "Data\_Input[6]" PULLDOWN;  NET "Data\_Input[5]" PULLDOWN;  NET "Data\_Input[4]" PULLDOWN;  NET "Data\_Input[3]" PULLDOWN;  NET "Data\_Input[2]" PULLDOWN;  NET "Data\_Input[1]" PULLDOWN;  NET "Data\_Input[32]" PULLDOWN;  NET "Data\_Output[31]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[30]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[29]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[28]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[27]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[26]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[25]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[24]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[23]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[22]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[21]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[20]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[19]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[18]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[17]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[16]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[15]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[14]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[13]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[12]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[11]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[10]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[9]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[8]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[7]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[6]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[5]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[4]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[3]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[2]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[1]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[32]" IOSTANDARD = LVCMOS18;  NET "Data\_Output[32]" LOC = R1;  NET "Data\_Output[31]" LOC = P2;  NET "Data\_Output[30]" LOC = P1;  NET "Data\_Output[29]" LOC = N2;  NET "Data\_Output[28]" LOC = M1;  NET "Data\_Output[27]" LOC = M2;  NET "Data\_Output[26]" LOC = L1;  NET "Data\_Output[25]" LOC = J2;  NET "Data\_Output[24]" LOC = G1;  NET "Data\_Output[23]" LOC = E1;  NET "Data\_Output[22]" LOC = D2;  NET "Data\_Output[21]" LOC = A1;  NET "Data\_Output[20]" LOC = L3;  NET "Data\_Output[19]" LOC = G3;  NET "Data\_Output[18]" LOC = K4;  NET "Data\_Output[17]" LOC = G4;  NET "Data\_Output[16]" LOC = K1;  NET "Data\_Output[15]" LOC = J1;  NET "Data\_Output[14]" LOC = H2;  NET "Data\_Output[13]" LOC = G2;  NET "Data\_Output[12]" LOC = F1;  NET "Data\_Output[11]" LOC = E2;  NET "Data\_Output[10]" LOC = D1;  NET "Data\_Output[9]" LOC = B1;  NET "Data\_Output[8]" LOC = B2;  NET "Data\_Output[7]" LOC = N3;  NET "Data\_Output[6]" LOC = M3;  NET "Data\_Output[5]" LOC = K3;  NET "Data\_Output[4]" LOC = H3;  NET "Data\_Output[3]" LOC = N4;  NET "Data\_Output[2]" LOC = L4;  NET "Data\_Output[1]" LOC = J4;  NET "SW[6]" IOSTANDARD = LVCMOS18;  NET "SW[5]" IOSTANDARD = LVCMOS18;  NET "SW[4]" IOSTANDARD = LVCMOS18;  NET "SW[3]" IOSTANDARD = LVCMOS18;  NET "SW[2]" IOSTANDARD = LVCMOS18;  NET "SW[1]" IOSTANDARD = LVCMOS18;  NET "SW[6]" LOC = R4;  NET "SW[5]" LOC = AA4;  NET "SW[4]" LOC = AB6;  NET "SW[3]" LOC = T5;  NET "SW[2]" LOC = V8;  NET "SW[1]" LOC = AA8;  NET "SW[6]" CLOCK\_DEDICATED\_ROUTE = FALSE;  NET "SW[5]" CLOCK\_DEDICATED\_ROUTE = FALSE;  NET "SW[4]" CLOCK\_DEDICATED\_ROUTE = FALSE;  NET "SW[3]" CLOCK\_DEDICATED\_ROUTE = FALSE;  NET "SW[2]" CLOCK\_DEDICATED\_ROUTE = FALSE;  NET "SW[1]" CLOCK\_DEDICATED\_ROUTE = FALSE; | | | | | |
| **六、思考与探索** | | | | | |
| 1. 实验结果记录：   QTSpim的编译结果  00000827, 0001102b, 00421820, 00622020 ,00832820, 00a33020, 00463804, 00a64820, 01264004, 00284826 ,01215020 ,01075822, 00e86022, 012c6824, 012c7025 ,00c77825, 00c78027, 00e38820, 02289004, 02239804, 00f3a004,  0281a820 ,0255b025, 0296b820, 0296c022, 02d4c822, 0241d026, 02d4d82b, 0354e02b, 02c2e820 ,0282f022, 017af820  存取器模块的读写结果  Input\_Data: 32'h 0000\_0001  Output\_Data: 32'h 0000\_0827  Input\_Data: 32'h 0000\_0002  Output\_Data: 32'h 0001\_102b  Input\_Data: 32'h 0000\_00ff  Output\_Data: 32'h 017a\_f820   1. 实验结论：   实验的结果与预计完全一致   1. 问题与解决方案：   问题一:书上给的QTSpim的初始地址有误  解决方案:书上给出来的初始地址为 0x0040000,这个地址在输入到QTSpim的初始地址的时候并不能写入,系统也并不会给自动补全高位,因此应该输入完整的8位数0x00400000;  问题二:在运行R\_CPU\_Test.asm这个文件的时候,书上给出来的初始地址有误  解决方案:在输入初始地址的时候,应该跳过系统给出来的前10条指定,直接从我们写入的指令的地址位置开始,即地址:0x00400024,而不是书上给出来的0x0040000;  问题三: 在运行R\_CPU\_Test.asm这个文件的时候,会一直报溢出的错误  解决方案:目前还没有解决方案,这个是程序的问题,要解决的话,只能重新去设计程序了,但是这个并不会影响最后生成的机器码,机器码是正确的   1. 思考题： 2. 实验结果与预计完全一致 3. 本实验由于是存储器和汇编译码这两个实验一起做的,在进行汇编译码的时候已经进行过读写操作的验证,结果完全正确,与写入的值完全一致 | | | | | |