课程实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **课程名称** | 计算机组成原理 | **指导教师** | 钟 旭  李运兰 | **实验日期** | 2021年10月8日 |
| **班 级** | 20计科04班 | **姓 名** | 孙紫铭 | **学 号** | B20200601133 |
| **实验成绩** |  | | | | |

|  |
| --- |
| **一、实验项目名称**  存储器实验 |
| **二、实验目的**  1. 掌握随机存储器RAM的工作特性及使用方法。  2. 了解RAM数据存储和读取的工作原理。  3. LPM存储类元件定制 |
| **三、实验要求**  必开，4学时。一人一组独立完成。 |
| **四、实验环境**  硬件：PC机一台 Intel Pentium 4 CPU 1.2GHz，512MB RAM及以上；  FPGA实验板一块（DDA-III主芯片*CycloneII（EP2C5T144C8））；*  软件：Windows 10，QUARTUS II配套软件 |
| **五、实验内容**   1. 完成两个寄存器内容进行交换，根据电路图完成电路的设计（vhdl） 2. 设计一个拥有八位地址线、八位数据输入输出线的RAM。数据读入和数据输出分开，存储空间为256\*8位。 3. 波形仿真验证或实验板硬件验证 |
| **六、实验原理**  1、寄存器内容交换原理图。  1  2.存储器读写操作原理图 |
| **七、实验步骤**  1.寄存器内容交换  (1)新建源文件  File->VHDL file 输入相应VHDL代码保存  ***library ieee***;  **use ieee**.std\_logic\_1164.all;  entity exp\_bus is //总线实体描述（实体与外部电路进行接口描述）  port(clk: in std\_logic; //端口定义  sw\_bus:in std\_logic; //标准逻辑位 in表示端口方向 输入  r1\_bus:in std\_logic;  r2\_bus:in std\_logic;  r3\_bus:in std\_logic;  k: in std\_logic\_vector(7 downto 0); //标准逻辑矢量，确定赋值方向 (n downto 0) or (0 downto n) 8位信号  lddr:in std\_logic\_vector(3 downto 1);  l: inout std\_logic\_vector(7 downto 0) );  end exp\_bus; //结束实体描述  architecture rtl of exp\_bus is //结构体描述 rtl是结构体名  signal r1,r2,r3,bus\_reg:std\_logic\_vector(7 downto 0):=(others=>'0');  //声明语句  begin //开始功能描述语句 输入输出之间的逻辑关系  ldreg:process(clk,lddr,bus\_reg)  begin //进程  if clk'event and clk='1' then  if lddr(1)='1' then r1<=bus\_reg;  elsif lddr(2)='1' then r2<=bus\_reg;  elsif lddr(3)='1' then r3<=bus\_reg;  end if;  end if; //顺序语句  end process; //结束进程  bus\_reg<=k when sw\_bus='0' else  r1 when r1\_bus='0' else  r2 when r2\_bus='0' else  r3 when r3\_bus='0' else  (others=>'Z');  l<=bus\_reg when (sw\_bus='0' or r1\_bus='0' or r2\_bus='0' or r3\_bus='0') else  (others=>'Z');  end rtl; //结束结构体描述  (2)创建项目  在此基础上建立新的工程   1. 编辑程序   编译  (4) 引脚分配  Assignments->pins     1. 下载至电路板   先将sw\_bus和r1、r2、r3调到无效状态，之后将sw调为有效，将数据存入bus，再将lddr1打开，关上，然后将sw\_bus调为无效，将r1调为有效，可以看到r1中存入我们想存的数据,同理可存入r2，并将r2和r3的数据进行交换。  2.存储器读写操作  (1)创建ram\_io工程  新建文件夹 ram\_io  新建Block，加入LMP\_RAM\_IO,并对其进行参数分配，具体参数及类型如图所示。    File->new->memory initialization file对随机存储器进行初始化    （两者均设置为十六进制）  （参数如图）  文件名保存不能出错，文件名为initial\_file  保存，为ram\_io.block新建工程，编译   1. 创建sw\_pc\_ar工程   新建文件夹 sw\_pc\_ar  VHDL代码如下：  library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  entity sw\_pc\_ar is //sw\_pc\_ar的实体描述  port( clk\_cdu,pcclr,pcld,pcen :in std\_logic; //端口定义  sw\_bus,pc\_bus,ldar :in std\_logic;  inputd :in std\_logic\_vector(7 downto 0);  arout :out std\_logic\_vector(7 downto 0);  d :inout std\_logic\_vector(7 downto 0) );  end sw\_pc\_ar;  architecture rtl of sw\_pc\_ar is //sw\_pc\_ar的结构体描述  signal pc,ar,bus\_reg:std\_logic\_vector(7 downto 0);  begin  seq1:process(clk\_cdu,ldar,bus\_reg) //进程1，描述三者逻辑关系  begin  if clk\_cdu'event and clk\_cdu='1' then  if ldar='1' then ar<=bus\_reg;  end if;  end if;  end process; //结束进程1  seq2:process(clk\_cdu,pcclr,pcld,pcen,bus\_reg) //进程2，描述管脚逻辑关系  begin  if pcclr='0' then pc<=(others=>'0');  elsif clk\_cdu'event and clk\_cdu='1' then  if (pcld='0' and pcen='1') then pc<=bus\_reg; //01时将总线上数据传入pc  elsif (pcld='1' and pcen='1') then pc<=pc+1; //11时pc自增一  end if;  end if;  end process; //结束进程2  bus\_reg<=inputd when sw\_bus='0' else  pc when pc\_bus='0' else  d;  d<=bus\_reg when (sw\_bus='0' or pc\_bus='0') else  (others=>'Z');  arout<=ar;  end rtl;  编译，新建工程sw\_pc\_ar   1. 创建ccq\_8工程   新建文件夹ccq\_8  将两个文件夹中的如下文件加入新建文件夹ccq\_8中    创建新的Block，将ram\_io和sw\_pc\_ar进行符号化（File->Create\Update->Create Symbol Files For Current File)  新建Block文件，加入符号化的ram\_io=和sw\_pc\_ar，并进行连线，如图所示    保存，编译   1. 仿真运行   新建波形仿真文件，按如图所示赋值，并编译运行。  ~T2SO~]KIXBUYI6S`QXMXBN |
|  |
| 1. **实验总结**   **1.寄存器内容交换**  74374是具有三态输出的8个D触发器，74244是三台缓冲器。BIDIR端口为双向端口，既可以实现输入又可以输出信号。bus line（粗线）中含有多个支线，是多个node line（细线）的汇总，从bus line中可以引出node line，也可以使node line汇总为bus line，总线命名一般为X[n…a]，支线为X[n]、X[n-1]、…、X[a]，其中X为名称，n-a为总线中汇总信号的数目  两个寄存器内的数据交换需要借助另一个寄存器作为中转站。例如：互换Reg1和Reg2数据的方法：Reg3<-Reg1；Reg1<-Reg2；Reg2<-Reg3。   1. **存储器读写操作**   Lpm-ram-io为双向端口的RAM  Lpm-ram-io参数设置Lpm-numwords（存储体中的存储单元的数量）、Lpm-width（存储单元的字长）、Lpm-widthad（地址寄存器的位数）  各管脚的作用：  （menmenab为使能端） |
| **九、教师评阅意见** |