Project4  FPGA开发 MIPS 微系统(支持设备与中断)

一、 设计说明

1.  MIPS 微系统应包括：MIPS 处理器、系统桥和 1 个定时器，8 位 7 段数码管、

32 位拨动开关。

2.  MIPS 处理器应支持 MIPS - Lite3 指令集。

a) MIPS-Lite3＝{MIPS-Lite2，ERET、MFC0、MTC0 }。

b) MIPS-Lite2＝{addu，subu，ori，lw，sw，beq，lui，addi，addiu，slt， j，jal，

jr，lb，sb }。

c) 所有运算类指令均可以不支持溢出；所有指令均可以不支持异常。

3.  MIPS 处理器为多周期设计。

4.  MIPS 微系统支持定时器硬件中断。

二、 系统桥与设备

5.  为了支持设备，MIPS 微系统需要配置系统桥。

a) 需要支持 3 个设备，即定时器、8 位 7 段数码管、32 位拨动开关。

b) 定时器的设计规范请参看《定时器设计规范.docx》。

6.  实验设备中的 8 位 7 段数码管由 2 个 4 位 7段数码管组成。

a) 数码管的介绍性材料请阅读：<http://baike.baidu.com/view/556862.htm>

三、 FPGA 内置模块的使用

7.  时钟定制电路

a) 系统时钟为 50MHz，速度过快，请使用 CoreGen 生成适合的时钟模块及

频率。

8.  片内块存储器

a) IM、DM 都请采用BlockRAM。

四、 中断机制

9.  为了支持异常和中断，处理器必须实现 0 号协处理器(CP0)。为此，必须实现

的 CP0 寄存器包括：SR、CAUSE、EPC、PrID。关于这几个寄存器，请大家

阅读《异常中断及协处理器》中的相关内容。

10.  考虑到简化以及与 MARS 模拟器一致，我们将只支持 0x00004180 这个入口地址，即

所有的异常与中断都从这里进入。

a) 你需要修改 NPC 模块，以确保当异常/中断发生时，NPC.NPC 输出

0x00004180。

11. 由于本系统只要求支持设备中断。MIPS内部异常(如指令错误)则不被考虑。

五、 微系统设计

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| PrAddr[31:2] | O | 32 位地址总线(最低 2 位忽略) |
| BE[3:0] | O | 4 位字节使能 |
| PrDIn[31:0] | I | 从 Bridge 模块读入的数据 |
| PrDOut[31:0] | O | 输出至 Bridge 模块的数据 |
| Wen | O | 时钟信号 |
| HWInt[7:2] | I | 6 个硬件中断请求 |

13.  多周期处理器由 datapath(数据通路)和 controller(控制器)组成。

12.  MIPS 处理器需要增加接口信号。以下为参考设计(只列出了新增的信号)：

a) 数据通路应至少包括如下 module：PC(程序计数器)、NPC(NextPC 计算

单元)、GPR (通用寄存器组，也称为寄存器文件、寄存器堆)、ALU(算术

逻辑单元)、EXT(扩展单元)、IM(指令存储器)、DM(数据存储器)、Bridge、

CP0 等。

14. 多周期数据通路应至少包括 PC、NPC、IM、DM 这些。其中：

a) IM：容量为 8KB(32bit×2048字)。

b) DM：容量为 12KB(32bit×3072字)。

15. 微系统中重要的地址范围和入口请见下表。

|  |  |  |
| --- | --- | --- |
|  | 地址或地址范围 | 备注 |
| 数据存储器 | 0x0000\_0000 至 0x0000\_2FFF |  |
| 指令存储器 | 0x0000\_3000 至 0x0000\_4FFF |  |
| PC 初始值 | 0x0000\_3000 |  |
| Exception Handler 入口地址 | 0x0000\_4180 |  |
| 定时器寄存器地址 | 0x0000\_7F00 至 0x0000\_7F0B | 定时器 3 个寄存器 |

a) Exception handler 的代码属于指令存储器。注意 handler 在指令存储器中

的地址位置以及正确编写 modelsim 仿真时指令的初始化文件。

16. 定时器的 ISR 请接入 MIPS 处理器的 HWInt[2]，即最低中断。

六、 文件、层次及模块实例化命名要求

17. 本 project 的顶层文件命名为：mini\_machine.v。

18.  模块层次命名如下：

|  |  |
| --- | --- |
| 第 1 层：  顶层模块名规范 | module mini\_machine(...) ; |
| 第        2        层      ：  mini\_machine 模块对  CPU、桥、定时器的  实例化规范 | xxx U\_MIPS(...) ;  xxx U\_BRIDGE(...) ;  xxx U\_TIMER(...) ; |
| 第 3 层：mips 处理器  对 PC、RF、IM、DM  的实例化规范 | yyy U\_PC(...) ;  yyy U\_RF(...) ;  yyy U\_IM(...) ;  yyy U\_DM(...) ; |
| 第 4 层：PC、IM、DM  模块内部变量命名  规范 | PC 模块：reg [31:2] PC ;  RF 模块：reg [31:0] gpr[31:1] ;  IM 模块：reg [31:0] im[1023:0] ;  DM 模块：reg [31:0] dm[2047:0] ; |

七、 模块定义【**WORD**】

19. 本部分要求与前一 Project 完全相同，请参考之。

请增加 32 位微动开关模块和 4 位 8 段数码管模块(实例化为 2 个，合起来能

够显示 8 个 16 进制字符)。

八、 测试要求

20. 请开发一个主程序以及定时器的 exception handler，实现秒计数显示功能。

整个系统完成如下功能：

1) 主程序通过读取 32 位微动开关值初始化变量 base\_sec，并初始化计数变

量 cnt\_sec 为 0。

2) 主程序将定时器初始化为模式 0，并加载正确的计数初值至预置计数初

值寄存器以产生 1s 的计数周期。

3) 主程序启动定时器计数后进入死循环。该循环不断的读取 32 位微动开关

值，一旦发现与之前的 base\_sec 不同，则更新 base\_sec 并重置 cnt\_sec，

从而实现从新值开始的秒计数。

4) 当定时器倒计数至 0 产生中断后，handler 完成如下功能：重置初

值寄存器从而再次启动定时器的计数器；cnt\_sec 加 1，并计数 base\_sec =

base\_sec + cnt\_sec；将 base\_sec 以十六进制方式从 7段数码管输出。

九、成绩及实验测试要求

21. 实验成绩包括但不限于如下内容：设计的正确性、实验报告等。

22. 实验测试时，你需要重点解读中断实现及软硬件协同机制。

a) 解读不仅应准确，而且应力求简洁。

十、其他要求

23. 打包文件：VerilogHDL 工程文件、code.txt、code.txt 所对应的汇编程序、项

目报告。

24. 本实验要求文档中凡是出现了【WORD】字样，就意味着该条目需要在实验

报告中清晰表达。