Project1 Logisim 开发单周期 MIPS 处理器

一、 设计说明

1. 处理器应支持的指令集 MIPS-C7：addu，subu，ori，lw，sw，beq，lui。
   a) addu，subu 可以不支持实现溢出。
2. 处理器为单周期设计。

二、 设计要求

3. 顶层设计视图包括如 Figure1 所示的部件，即 Controller(控制器)、IFU(取指令单元)、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)、DM(数据存储器)、EXT(扩展单元)、多路选择器及 splitter。

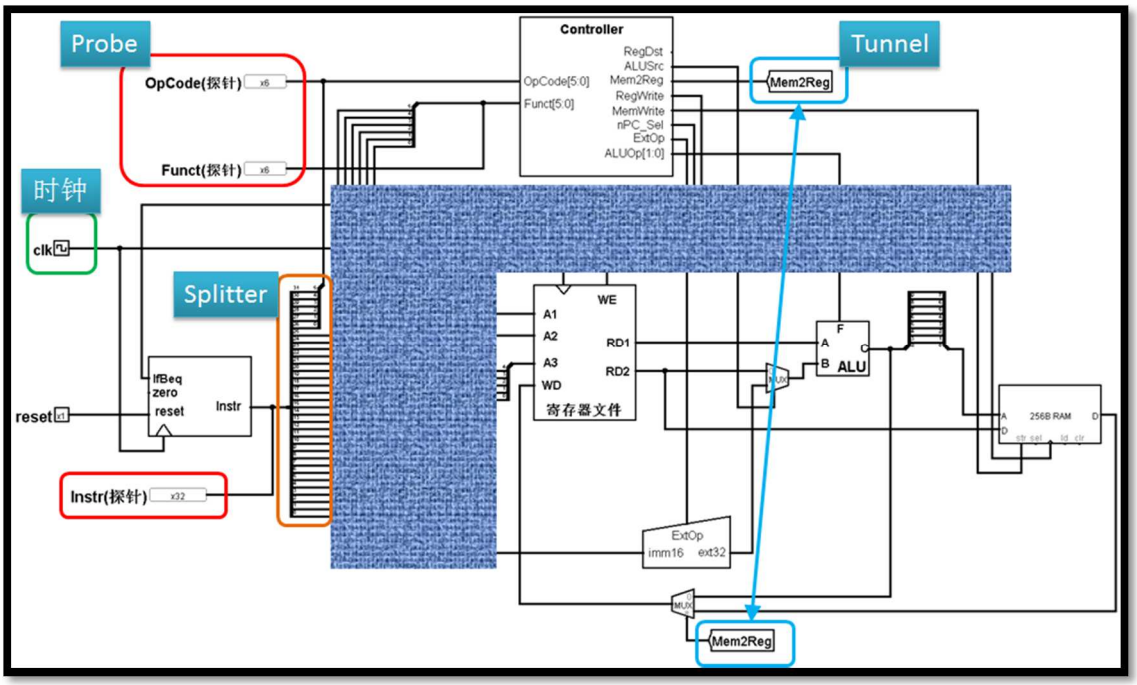

Figure1 顶层设计

   a) 顶层设计视图的顶层有效驱动信号包括且仅包括：clk、reset。
   b) 必须采用模块化和层次化设计。整个设计文件目录结构应类似于 Figure2。

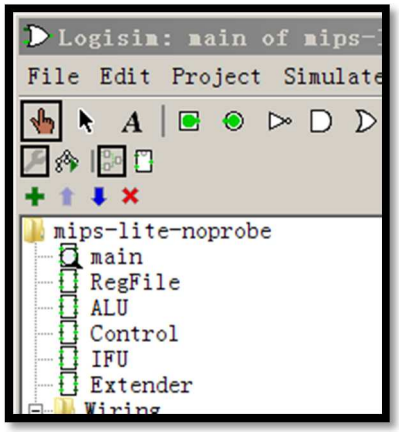

Figure2 设计层次(仅供参考)

4. IFU：内部包括 PC、IM(指令存储器)及相关逻辑。
   a) PC：用寄存器实现，宽度为 30 位。PC 应具有复位功能。
   b) IM：容量为 32bit×32 字，用 ROM 实现。
   c) 说明：由于 IM 地址仅为 5 位，因此需要用 Splitter 连接 PC 低位地址与 IM 地址。
5. GPR：以 32 个 32 位具有写使能的寄存器为基础，辅以多路选择器。
6. ALU：实现加法及减法时，允许使用 Logisim 内置的 Adder 及 Subtractor。也欢迎以门电路为基础自行开发。
7. EXT：可以使用 Logisim 内置的 Bit Extender。
8. DM：容量为 32bit×32 字，用 RAM 实现。
   a) DM 应采用双端口模式，即设置 RAM 的“Data Interface”属性为“Separate load and store ports”。
9. 必须有时钟源，即如 Figure1 中绿圈所示。
   a) 只有设置了时钟源，系统才能自动运行，从而让程序连续运行。
10. PC 复位后初值为 0x0000_3000，目的是与 MARS 的 Memory Configuration 相配合。
    a) 教师用测试程序将通过 MARS 产生，其配置模式如 Figure3 所示。因此你也需要用 MARS 编写测试程序并生成能够在 Logisim 中运行的代码。

Figure3 MIPS 存储配置模式(MARS memory configuration)

三、 模块定义

11. 为了完成设计，你应该仿照下面给出的 IFU 模块定义，自行给出所有功能部件的模块定义。

    a) IFU、GPR、ALU、EXT、DM、Controller。

12. IFU 模块定义(参考样例)

(1) 基本描述

IFU 主要功能是完成取指令功能。IFU 内部包括 PC、IM(指令存储器)以及其他相关逻辑。IFU 除了能执行顺序取值令外，还能根据 BEQ 指令的执行情况决定顺序取值令还是转移取值令。

(2) 模块接口

信号名	方向	描述
IfBeq	I	当前指令是否为 beq 指令标志。 1：当前指令为 beq

		0：当前指令非 beq
Zero	I	ALU 计算结果为 0 标志。 1：计算结果为 0 0：计算结果非 0
clk	I	时钟信号
Reset	I	复位信号。 1：复位 0：无效
Instr[31:0]	O	32 位 MIPS 指令

(3) 功能定义

序号	功能名称	功能描述
1	复位	当复位信号有效时，PC 被设置为 0x00000000。
2	取指令	根据 PC 从 IM 中取出指令。
3	计算下一条指令地址	如果当前指令不是 beq 指令，则 PC←PC+1 如果当前指令是 beq 指令，并且 zero 为 0，则 PC←PC+1 如果当前指令是 beq 指令，并且 zero 为 1，则 PC←PC+sign_ext(当前指令 15..0) [注]PC 取地址为 4 字节，固低 2 位地址可以去除。

四、 控制器设计

13. 请仿照下图给出 MIPS-Lite1 指令集的单周期控制器真值表。【WORD】

See MIPS Green Sheet							
func	10 0000	10 0010	n/a				
op	00 0000	00 0000	00 1101	10 0011	10 1011	00 0100	
	add	sub	ori	lw	sw	beq	
Control Signals: RegDst	1	1	0	0	X	X	
ALUSrc	0	0	1	1	1	0	
MemtoReg	0	0	0	1	X	X	
RegWrite	1	1	1	1	0	0	
MemWrite	0	0	0	0	1	0	
nPC_sel	0	0	0	0	0	1	
ExtOp	X	X	0	1	1	X	
ALUctr<2:0>	Add	Subtract	Or	Add	Add	Subtract	
All Supported Instructions							

Figure4 控制信号真值表

a) 结合真值表，请给出数据通路每个功能部件的每个控制信号的布尔表达式。

b) 表达式中只能使用“与、或、非”3 种基本逻辑运算。

c) 每个控制信号的表达式应该是指令 opcode 域与 funct 域的函数。

d) 对于多位的控制信号(如 ALUCtr)，应诸位给出其逻辑表达式。

14. 请在 Logisim 中完成控制器设计。

a) 控制器整体结构需要仿照 Figure 5 实现。

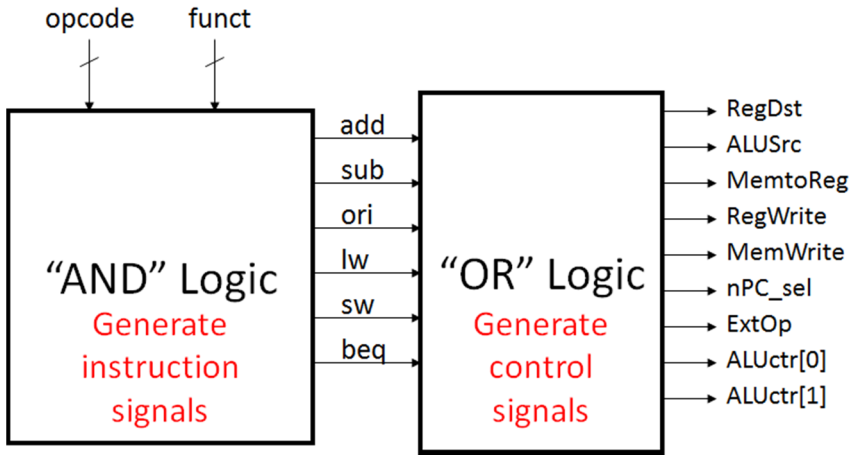

Figure 5 控制器基本结构

b) 控制信号必须仿照下图方式实现。

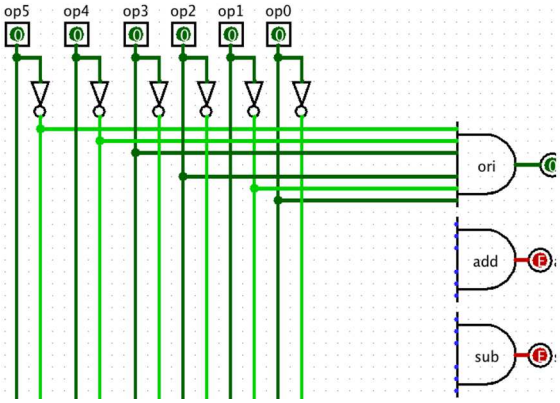

Figure6 与阵列：译码产生指令标识

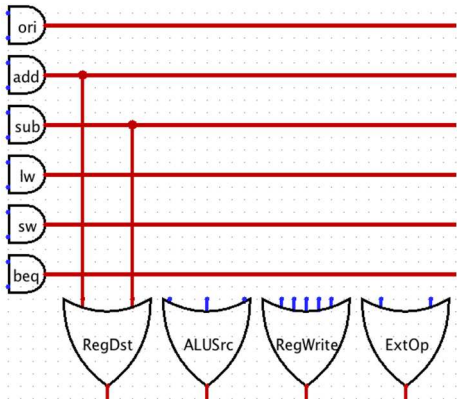

Figure7 或阵列：产生最终的控制信号

五、 测设要求

15. 构造 1 个至少 20 条以上指令的测试程序，并加载至 IFU 中运行通过。

a) MIPS-Lite1 定义的每条指令至少出现 1 次以上。

16. 详细说明你的测试程序原理。【WORD】

a) 应明确说明测试程序的测试期望，即应该得到怎样的运行结果。

b) 每条汇编指令都应该有注释。

六、 问答【WORD】

17. 请充分利用 Figure4 中的 X 可以将控制信号化简为最简单的表达式。
18. 对于 Figure6、Figure7 中的与或阵列来说，1 个 3 输入与门最终转化为 2 个 2 输入与门，1 个 4 输入与门最终转化为 3 个 2 输入与门，依次类推。或阵列也类似计算。那么
    a) 请给出采用 Figure6、Figure7 中的方法设计的每个控制信号所对应的 2 输入与门、2 输入或门、非门的数量。
    b) 请与第 17 项对比，你更喜欢哪种设计方法。为什么?

七、 其他要求

19. 打包文件：Logisim 工程文件、测试程序、测试程序二进制文件、项目报告。
20. 截止时间：教师指定。
21. 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。
22. 实验报告请按照《实验报告撰写规则》要求排版。

八、 开发与调试技巧

23. 对于每条指令，请认真阅读《MIPS32® Architecture For Programmers Volume II: The MIPS32® Instruction Set》!
24. Figure1 中 Tunnel 的用途是将具有相同 name 的 tunnel 连接在一起。Tunnel 可以避免将图画的很乱。
25. Figure1 中 Probe 的用途是显示被 probed 信号的值，便于调试。
26. Figure1 中 Splitter 的用途是从某组信号中提取其中部分信号。例如，IFU 输出 32 位指令，需要提取高 6 位(OpCode)和低 6 位(Funct)分别输入 controller。
    a) splitter 是有位序的!但字号太小，需要放大设计图(界面左下有比例设置)。
    b) 建议高位永远在上，低位永远在下
27. 如果你对于 Logisim 内置的某个部件的端口不明白，请：
    a) 仔细阅读 Help→Library Refrence 关于该部件的描述。
    b) 放大 Logisim 显示比例直至能清晰看到代表部件的各个端口的圆点，然后将鼠标停留相应的圆点上，就可以读取端口具体信息。

28. 建议先在 MARS 中编写测试程序并调试通过。
29. 片选信号就是对存储器地址的高位分析。
    a) 假设 DM 有 256MB 容量，并且映射在 0x3000_0000～0x3FFF_FFFF 区间。那么只需要把高 4 位地址与 0x3 进行比较，比较结果就是 DM 的片选信号。
    b) 为了实现片选，你需要用基本逻辑门搭 2 个片选信号生成逻辑，一个输出至 IM，一个输出至 DM。
    c) Logisim 内置的 RAM 有片选信号！
30. 提示：你可以考虑增加 7 段数码管等输入输出设备来让你的测试结果更加直观。
    a) 本条非必做要求。
    b) 7 段数码管也需要类似片选等信号，其工作原理与第 29 项类似。