Logisim完成单周期处理器开发

# 设计说明

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

# 设计要求

1. 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。
   1. 顶层设计视图的顶层有效驱动信号包括且仅包括：clk、reset。
   2. 提示：图中的其他字符均不是端口信号。

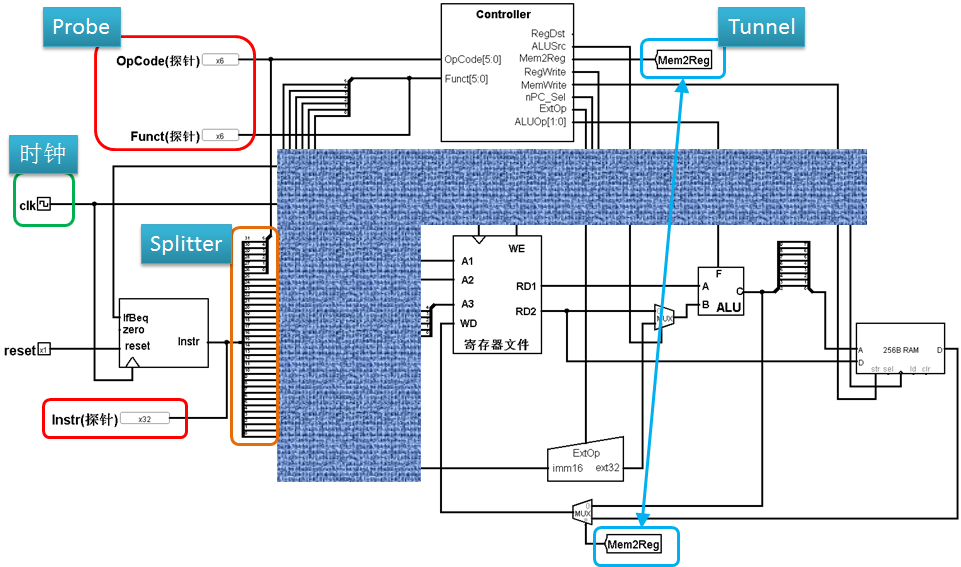


Figure1 顶层设计

* 1. 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。

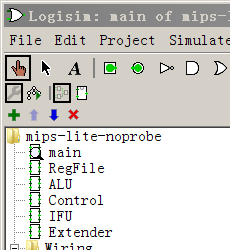


Figure2 设计层次(仅供参考)

1. IFU：内部包括PC、IM(指令存储器)及相关逻辑。
   1. PC：用寄存器实现，宽度为32位。PC应具有复位功能。
   2. IM：容量为1KB，用ROM实现。
2. 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。
3. ALU：32位运算器
4. EXT：可以使用logisim内置的Bit Extender。
5. DM：容量为1KB，用RAM实现。
   1. DM应采用双端口模式，即设置RAM的“Data Interface”属性为“Separate load and store ports”。
6. 必须有时钟源，即如Figure1中绿圈所示。
   1. 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。

# 测试要求

1. 所有指令都应被测试充分。
2. 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。
   1. MIPS-Lite定义的每条指令至少出现1次以上。
   2. 演示时，测试程序必须已经通过IFU中的IM的“Load Image”加载完毕。

# 其他要求

1. 打包文件：Logisim工程文件、测试程序二进制文件
2. 时间要求：实验指导教师指定。

# 开发与调试技巧

1. 对于每条指令，请认真阅读《MIPS32® Architecture For Programmers Volume II: The MIPS32® Instruction Set》！
2. 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。
3. Figure1中Probe的用途是显示被probed信号的值，便于调试。
4. 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。
   1. 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)。
   2. 建议高位永远在上，低位永远在下
5. 如果你对于logisim内置的某个部件的端口不明白，请：
   1. 仔细阅读Help🡪Library Refrence关于该部件的描述。
   2. 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。
6. 建议先在MARS中编写测试程序并调试通过。
   1. 注意MARS中的“Settings🡪Memory Configuration”只能配置指令存储器起始地址为0地址，而不能将指令存储器和数据存储器的起始地址均配置为0地址！
   2. 由于logisim设计中的DM起始地址为0，因此请仔细观察所用到的指令，在把MARS中调试通过的二进制码导出后，你可能需要手工修改指令码中的数据偏移。
   3. 提示：事实上，在现代主流计算机中，数据存储器和指令存储器的起始地址不应该重叠。但在本设计中，由于采用分离存储器设计方案，因此可以暂时忽略这一点。
7. 提示：你可以考虑增加7段数码管等输入输出来让你的测试结果更加直观。