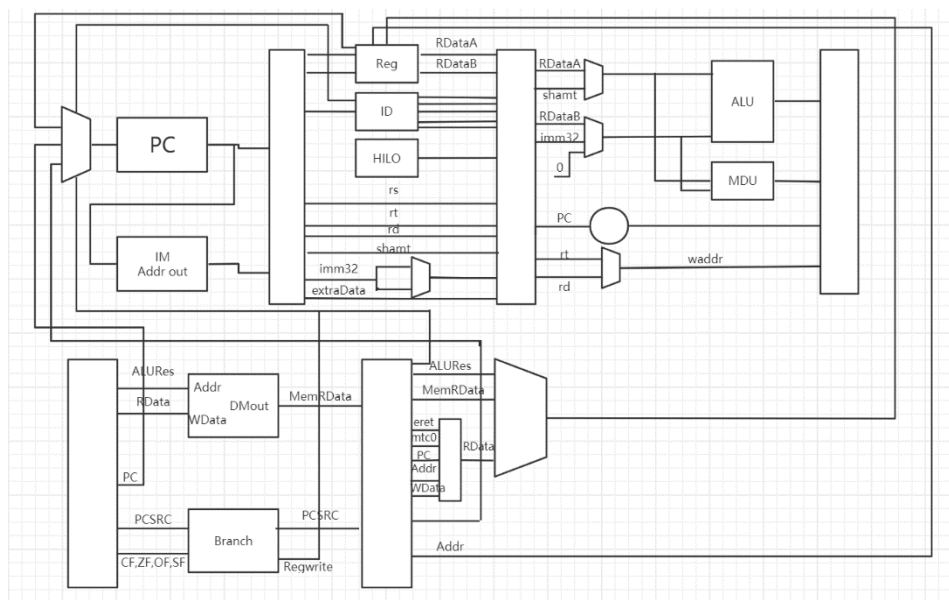


杭州电子科技大学 2 队  
陈嘉毅、范子杭、龚家伟

在本次“龙芯杯”全国大学生计算机系统能力培养大赛团队赛初赛中，我们队初步使用了 Verilog HDL 实现了一个基于 MIPS 基准指令集的单发射五级流水线 CPU。

整个系统主要分为四大部分，即执行部件、暂存器、控制器、接口转换器。其中按照五级流水线的标准，又将控制部件按 IF、ID、EX、MEM、WB 五个不同的运行阶段分为五个模块，并用四个暂存器连接这五个模块。控制器与所有受控执行部件直接相连，而接口转换器只与 IF 与 MEM 两个需要访问外部存储器的模块相连。其中大致结构如下图所示。



IF 模块主要功能为控制程序计数器实现顺序执行或跳转功能, 并访问指令存储器获得

指令以传递到下一模块，输入主要有程序计数器的使能、选择信号、新的地址信息，而输出则是 PC 值、指令字，除此之外还有一些暂停信号以及 SRAM 访存信号。模块内部主要由 PC 寄存器、若干选择器、加法器构成。

### （三）ID 模块设计

ID 模块主要功能为实现 HILO 寄存器与寄存器堆、执行指令所需的各信号产生。输入主要为寄存器堆写数据写地址写使能、指令字等，输出主要为指令执行所需各信号、指令各字段、寄存器堆读数据等。模块内部主要由 4 大部分构成，分别为寄存器堆、HILO 寄存器、一级译码器、二级译码器。其中一级译码器将指令划分为不同字段，如 opcode、rs、rt、rd 等；二级译码器根据指令某些字段判断当前是哪条指令，并通过选择器或查找表的方式输出所需信号。

### （四）EX 模块设计

EX 模块主要功能为数据运算与相对跳转的新 PC 值产生。输入有运算器两个端口的选择信号，代运算数据、运算类型信号、PC 值等，输出有运算结果、标志寄存器值、新 PC 值、HILO 值、写地址等。其中主要由 ALU、MDU（乘除运算），PC 专用加法器构成。ALU 两输入端口均有选择器选择输入数据，MDU 会输出暂停信号以便在计算除法时暂停流水线。

### （五）MEM 模块设计

MEM 模块主要功能为除了输入输出数据与各信号实现对数据存储器的按不同尺寸（字、半字、字节）的访问、条件跳转指令的判断。输入主要有访问存储器所需的地址、写数据、使能、写使能与判断跳转的标志寄存器值、跳转信号等，输出主要为存储器读数据，PC 跳转信号以及暂停信号，除此之外输入输出信号还有 SRAM 访存信号。内部主要为由逻辑门构成的组合电路与选择器、比较器，实现信号与数据的判断和处理。

### （六）WB 模块设计

WB 模块主要功能为实现中断异常的 CP0 寄存器以及选择写回的数据，输入包含 CP0 模块所需的 PC、地址、写数据等信号和各类写回数据（运算器运算结果、存储器读数据、CP0 读数据等）以及相应的选择信号，输出为 CP0 的跳转新 PC 值和经过选择后的寄存器堆写数据。内部由若干选择器以及 CP0 模块（包含寄存器以及实现相应功能所需的逻辑组合电路）构成。

## （七）暂存器模块设计

暂存器模块主要功能是为流水线提供支撑，使各阶段的执行部件相互独立，不互相影响。其由大量触发器构成，输入输出取决于其之前或之后执行部件的输入输出。

## （八）控制器模块设计

控制器主要功能为控制整条流水线各部件的运行。输入有各模块发送的暂停信号、判断是否存在数据冲突的所需信号（写使能、写地址、读地址等）、判断是否存在跳转指令的信号，输出有模块的时钟信号、PC 使能信号、以及用于控制暂存器的暂停信号。内部有主要由判断是否数据冲突的组合逻辑电路，以及一些寄存器加组合电路构成的时序电路组成，以达到按正确的时序发送各模块所需的时钟信号以及部分暂停信号的效果。

## （九）接口转换器模块设计

该模块我们直接使用了比赛资料包中提供的“`cpu_axi_interface.v`”，但最终提交的 SRAM 版本未使用该文件。

# 三、设计结果

## （一）设计交付物说明

- 1、Top.v: myCPU 顶层模块；
- 2、CU.v: CPU 冲突与流水线暂停综合控制器；
- 3、WRConflict.v: 数据冲突判别模块；
- 4、IF.v ID.v EX.v MEM.v WB.v: 五个执行阶段对应的部件；
- 5、IF\_ID.v ID\_EX.v EX\_MEM.v MEM\_WB.v:；连接五个执行阶段的暂存器；
- 6、PC.v: 程序计数器模块；
- 7、ID1.v: 一级指令译码器；
- 8、ID2.v: 二级指令译码器；
- 9、RegHILO.v: HILO 寄存器；
- 10、RegFile.v: 通用寄存器堆；
- 11、ALU.v: 运算器；
- 12、MDU.v: 乘除法运算器；

13、 CP0.v: CP0 寄存器以及异常中断管理模块;

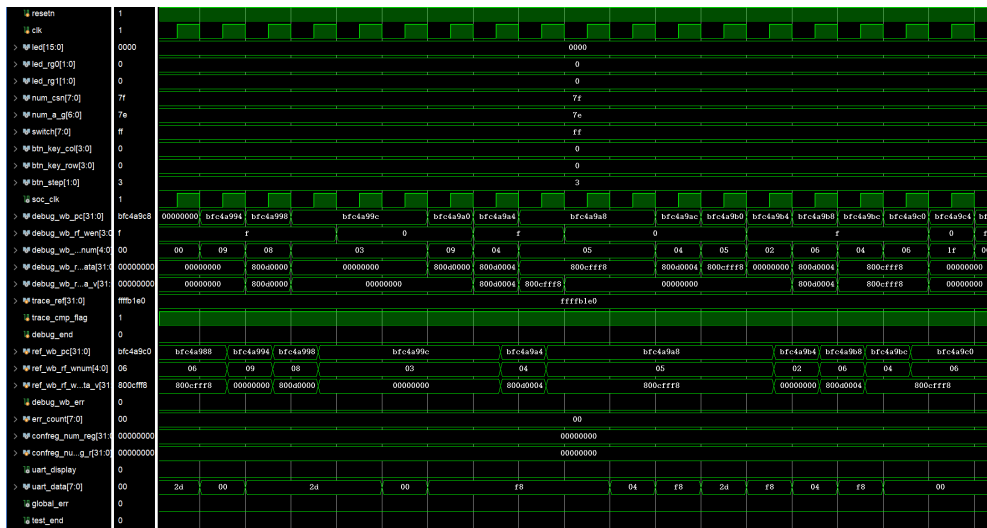
14、 divu.xci: 无符号除法器 IP;

15、 div.xci: 有符号除法器 IP;

只需要将上述文件导入 soc\_sram\_func 工程即可仿真、综合并上板。

## （二）设计演示结果

因时间紧张，AXI 接口版本无法对随机延迟的存储器做出适配没有采用，SRAM 接口版本在功能上也有一定缺陷，在仿真中可以对几乎所有指令正确执行，但对于中断异常的支持有所缺失。板级测试因时间原因无法调试至成功运行状态，因此分数文件 `score.xls` 无法填写。仿真中某时刻图像如下：



#### 四、参考设计说明

1、除法器模块 div.xci 和 divu.xci 使用了 Xilinx IP – Divider;

2、类 SRAM 与 AXI 转接器 `cpu_axi_interface.v` 采用在“`nscsc2021_group_v0.01.rar`”比赛资料包中的参考代码（最终未使用）；

3、ALU.v 部分参考了下列文章中的 ALU 设计，例如信号的名称及含义，部分 ALU 功能的实现。参考链接：[https://blog.csdn.net/weixin\\_43074474/article/details/95872934](https://blog.csdn.net/weixin_43074474/article/details/95872934)