**1总体设计**

**1.1流水线**

本CPU采用3级流水结构，3个阶段分辨是取指译码，运算访存，回写。在同一时刻，最多有3条指令在流水线上执行。CPU的流水线部分由以下5个部分组成

pc\_reg 生成next取指地址并且输出当前取指地址，收到跳转情况与流水线暂停的控制

id 译码。根据指令，判断指令的类型，决定是否要读寄存器，是否要 读/写内存，生成执行阶段所要的参数。并检测syscall、自陷异常

id\_ex 译码阶段和运算阶段之间的流水线寄存器。

ex 执行。根据译码完成给出信息进行执行其中mem（访存模块），div（除法模块），都归属执行阶段的多指令操作

mem\_wb译码阶段和运算阶段之间的流水线寄存器。并进行精准异常检测

**1.2流水线冲突的处理**

**数据冲突** 由于流线采用单发、顺序发射顺序回写所有流水线只会出现RAW（read after write）冲突，此冲突只会产生在译码读取regfile时，采用旁路相关性检测模块（OITF）解决,本质就是fifo存储正在流水线执行指令回写目的寄存器索引号，通过模块内对比给出冲突指示。此方法针对hi/lo、LLbit和cp0三种寄存器都使用类似方式解决数据冲突

**控制冲突** 控制冲突主要包括跳转指令和异常跳转，主要使用改变PC和流水线冲刷解决

**1.3流水线暂停**

流水线暂停主要由两种情况产生，第一执行多周期指令，例如除法、访存中load 指令；第二种情况就是数据冲突；两种情况都暂停流水线取指译码阶段，后面阶段正常执行。以达到流水线等待多周期指令完成和回写完成后的冲突撤销继续取指，多周期指令需要外部模块ctrl进行控制

**1.4流水冲刷**

流水线冲刷主要由两种情况产生，第一种情况，跳转/分支指令，但是MIPS32指令集采用的是指令槽的模式，又由于流水线采用三级流水所以不用冲刷指令只需要冲刷取指PC。

第二种情况，异常跳转，由于需要精准异常，所以异常判断在回写阶段之前统一进行异常判断。所以流水线执行阶段之前都需要冲刷。所以即需要冲刷取指PC又需要冲刷译码指令。异常的流水线冲刷也需要ctrl模块进行跳转的地址生成。

**1.5外设**

由于只是验证流水线，除了ram和rom外没有其他外设，并且两块存储空间为独立空间，并没有使用任何总线合并内存空间

**2模块设计**

整个工程分两大模块，第一存储外设ram、rom，第二就是内核模块

内核模块主要分为几大类

第一流水线类，1.1已经有详细的介绍了

第二类寄存器类，regfile通用32个寄存器模块，reghilo高低位寄存器模块、regLLbit链接状态位寄存器、cp0协处理器。以上几个寄存器都是MIPS32指令集需要的寄存器

第三类数据冲突类，oitf通用32寄存器冲突模块、oitfhi\oitflo高低位寄存器冲突模块、oitfLLbit链接状态位冲突模块、oitfcp0协处理器冲突模块.以上几个模块都是伴随寄存器的旁路相关性检测模块。

第四类控制类，ctrl主要职责是多周期指令暂停控制和异常跳转控制功能

**3指令实现**

空指令：pref、sync、nop

逻辑指令：ori、andi、xori、lui、and、or、xor、nor

移位指令：sll、srl、sra、sllv、srlv、srav

移动指令：movn、movz、mfhi、mflo、mthi、mtlo

算术指令：add、addu、sub、subu、slt、sltu、addi、addiu、slti、sltiu、clz、clo、mul、mult、multu、madd、maddu、msub、msubu、div、divu

跳转/分支指令：jr、jalr、j、jal、beq、bgtz、blez、bne、bltz、bltzal、bgez、bgezal

访存指令：lb、lbu、lh、lhu、lw、lwl、lwr、ll、sb、sh、sw、swl、swr、sc

协处理器指令：mtc0、mfc0

异常指令：syscall、自陷指令、eret