

# 流水线 CPU 设计报告

## 设计说明：

1. 处理器应支持 MIPS-lite2 指令集。

MIPS-lite2={ addu, subu, ori, lw, sw, beq, lui, j, jal, jr, nop }

2. 处理器为流水线设计。

## 数据通路设计

### 1.1 五级流水线设计

流水线处理器分为 F, D, E, M, W 五级流水，其中每级流水所包含的部件如下：

F		
部件	输入	输出
PC	D_nPC[31:0]	PC[31:0]
IM	PC[31:0]	Instr[31:0]

D		
部件	输入	输出
EXT	Instr[16:0]	EXTout
GRF	RS	RD1
	RT	
	W_WA	RD2
	W_WD	
CMP	RD1	judge
	RD2	
nPC	PC	D_nPC
	D_Instr	
	RD1	

E		
部件	输入	输出
ALU	ALUop1	ALUout
	ALUop2	

M		
部件	输入	输出
DM	A	MD
	WD	

  

W		
部件	输入	输出
WK	M_ALUout	MD
	MD	
	PC_8	

1. 2流水寄存器设计

共有 4 级流水线寄存器：FtoD，DtoE，EtoM，MtoW

其中每级流水寄存器所寄存的数据分别如下：

流水寄存器级	数据
FtoD	PC
	Instr
DtoE	RD1
	RD2
	EXTout
	WA
EtoM	ALUout
	ALUop2
MtoW	MD
	ALUout
	PC_4
	WA

2. 控制器设计

2. 1 数据通路控制器（Contro）

采用集中式译码控制器，控制信号及端口如下：

端口	方向	描述
Opcode[5:0]	I	指令操作码
Funct[5:0]	I	指令功能码
branch	0	跳转信号
ExtOp[1:0]	0	位扩展方式
MemtoReg	0	读内存信号
MemWrite	0	内存写使能信号
Branch	0	分支信号
ALUCtrl[3:0]	0	ALU 控制信号，见 ALU 模块
ALUSrc	0	ALU 操作数 2 的来源 0: 寄存器 1: 立即数
RegDst	0	寄存器写地址选择 0: Instr[20:16] 1: Instr[15:11]
RegWrite	0	寄存器写使能信号

各指令具体信号控制如下：

端口	addu	subu	nop	sw	lw	ori	beq	lui	j	jal	jr
ExtOp[1:0]	00	00	00	01	01	00	11	10	xx	xx	xx
MemtoReg	00	00	00	xx	01	00	xx	00	00	10	00
MemRead	0	0	0	0	1	0	0	0	0	0	0
MemWrite	0	0	0	1	0	0	0	0	0	0	0
bCtrl	xxx	xxx	xxx	xxx	xxx	xxx	010	xxx	xxx	xxx	xxx
ALUCtrl[2:0]	000	001	000	000	000	011	000	000	xxx	xxx	000
ALUSrc	0	0	0	1	1	1	0	1	0	0	0
RegDst	00	00	00	xx	01	01	xx	01	xx	10	xx
RegWrite	1	1	0	0	1	1	0	1	0	1	0
PC_Src	00	00	00	00	00	00	01	00	10	10	11
branch	0	0	0	0	0	0	1	0	0	0	0

## 2.2 暂停器设计

当流水线处于 D 级时就需要判断现在是否需要暂停流水线

暂停控制信号 stop 执行动作；

1. 冻结 D 级流水寄存器
2. 清除 E 级流水寄存器：指令全为 0，等价于插入 nop
3. 禁止 PC：防止 PC 继续计数，PC 保持当前值

D 当前指令			E 当前指令			M 当前指令
指令类型	源寄存器	Tuse	cal_r/1/rt	cal_i/1/rd	load/2/rt	load/1/rt
b	rs	0	stall	stall	stall	stall
	rt		stall	stall	stall	stall
cal_r	rs	1			stall	
	rt				stall	
cal_i	rs	1			stall	
load	rs	1			stall	
store	rs	1			stall	
	rt	2			stall	
jr	rs		stall	stall	stall	stall

## 2.3 转发控制器设计

分析各个指令可能出现的冲突情况，构建下表

冲突			M 当前指令			W 当前指令(Tnew)			
流水级	指令类型	冲突位置	cal_r/rd	cal_i/rt	jal/rt	cal_r/rd	cal_i/rt	load/rt	jal/rt
D	cal_r	rs	AO	AO	M_PC+8	W_RD	W_RD	W_RD	W_RD
		rt	AO	AO	M_PC+8	W_RD	W_RD	W_RD	W_RD
	cal_i	rs				W_RD	W_RD	W_RD	W_RD
	load	rs				W_RD	W_RD	W_RD	W_RD
	store	rs				W_RD	W_RD	W_RD	W_RD
		rt				W_RD	W_RD	W_RD	W_RD
	jr	rs	AO	AO	M_PC+8	W_RD	W_RD	W_RD	W_RD
	b	rs	AO	AO	M_PC+8	W_RD	W_RD	W_RD	W_RD
		rt	AO	AO	M_PC+8	W_RD	W_RD	W_RD	W_RD
E	cal_r	rs	AO	AO	M_PC+8	W_RD	W_RD	W_RD	W_RD
		rt	AO	AO	M_PC+8	W_RD	W_RD	W_RD	W_RD
	cal_i	rs	AO	AO	M_PC+8	W_RD	W_RD	W_RD	W_RD
	store	rs	AO	AO	M_PC+8	W_RD	W_RD	W_RD	W_RD
		rt				W_RD	W_RD	W_RD	W_RD
	load	rs	AO	AO	M_PC+8	W_RD	W_RD	W_RD	W_RD
M	store	rs				W_RD	W_RD	W_RD	W_RD

根据表格构建转发控制器，控制 D 级中 RD1，RD2 的来源，E 级中 ALU 两个操作数的来源以及 M 级 WD 的来源，冲突情况不同来选择转发数据源

## 思考题

在本实验中你遇到了哪些不同指令组合产生的冲突？你又是如何解决的？相应的测试样例是什么样的？请有条理的罗列出来。

类型	测试类型	前序指令	冲突位置	冲突寄存器	测试序列
R 型 (以 addu 为例)	R-W-RS	addu	D	RS	addu \$4, \$4, \$5 nop nop addu \$4, \$4, \$5
	I-W-RS	ori	D	RS	ori \$4, \$5, 0xffff nop nop addu \$4, \$4, \$5
	LD-W-RS	lw	D	RS	lw \$4, 0(\$5) nop nop addu \$4, \$4, \$5
	JAL-W-RS	jal	D	RS	jal loop nop nop addu \$1, \$31, \$1
	R-W-RT	addu	D	RT	addu \$4, \$4, \$5 nop nop addu \$4, \$5, \$4
	I-W-RT	ori	D	RT	ori \$4, \$5, 0xffff nop nop addu \$4, \$5, \$4
	LD-W-RT	lw	D	RT	lw \$4, 0(\$5) nop nop addu \$4, \$5, \$4
	JAL-W-RT	jal	D	RT	jal loop nop nop addu \$1, \$1, \$31
	R-W-RS	addu	E	RS	addu \$4, \$4, \$5 nop addu \$4, \$4, \$5 nop
	I-W-RS	ori	E	RS	ori \$4, \$5, 0xffff nop addu \$4, \$4, \$5 nop
	LD-W-RS	lw	E	RS	lw \$4, 0(\$5)

					nop addu \$4, \$4, \$5 nop
	JAL-W-RS	jal	E	RS	jal loop nop addu \$1, \$31, \$1 nop
	R-M-RS	addu	E	RS	addu \$4, \$4, \$5 addu \$4, \$4, \$5 nop
	I-M-RS	ori	E	RS	ori \$4, \$5, 0xffff addu \$4, \$4, \$5 nop
	JAL-M-RS	jal	E	RS	jal loop addu \$1, \$31, \$1 nop
	R-W-RT	addu	E	RT	addu \$4, \$4, \$5 nop addu \$4, \$5, \$4 nop
	I-W-RT	ori	E	RT	ori \$4, \$5, 0xffff nop addu \$4, \$5, \$4 nop
	LD-W-RT	lw	E	RT	lw \$4, 0(\$5) nop addu \$4, \$5, \$4 nop
	JAL-W-RT	jal	E	RT	jal loop nop addu \$1, \$1, \$31 nop
	R-M-RT	addu	E	RT	addu \$4, \$4, \$5 addu \$4, \$5, \$4 nop
	I-M-RT	ori	E	RT	ori \$4, \$5, 0xffff addu \$4, \$5, \$4 nop
	JAL-M-RT	jal	E	RT	jal loop addu \$1, \$1, \$31 nop
I 型 (以 ori 为	R-W-RS	addu	D	RS	addu \$4, \$4, \$5 nop nop

例)					ori \$4, \$4, 0xffff
	I-W-RS	ori	D	RS	ori \$4, \$5, 0xffff nop nop ori \$4, \$4, 0x0000
	LD-W-RS	lw	D	RS	lw \$4, 0(\$5) nop nop ori \$4, \$4, 0xffff
	JAL-W-RS	jal	D	RS	jal loop nop nop ori \$1, \$31, 0xffff
	R-W-RS	addu	E	RS	addu \$4, \$4, \$5 nop ori \$4, \$4, 0xffff nop
	I-W-RS	ori	E	RS	ori \$4, \$5, 0xffff nop ori \$4, \$4, 0xf0f0 nop
	LD-W-RS	lw	E	RS	lw \$4, 0(\$5) nop ori \$4, \$4, 0xffff nop
	JAL-W-RS	jal	E	RS	jal loop nop ori \$1, \$31, 0xffff nop
	R-M-RS	addu	E	RS	addu \$4, \$4, \$5 ori \$4, \$4, 0xffff nop
	I-M-RS	ori	E	RS	ori \$4, \$5, 0xffff ori \$4, \$4, 0xf0f0 nop
	JAL-M-RS	jal	E	RS	jal loop ori \$1, \$31, 0xffff0 nop
LD 型	R-W-RS	addu	D	RS	addu \$4, \$4, \$5 nop nop lw \$5, 0(\$4)
	I-W-RS	ori	D	RS	ori \$4, \$5, 0xffff nop



					nop lw \$5, 0 (\$4)
	LD-W-RS	lw	D	RS	lw \$4, 0 (\$5) nop nop lw \$5, 0 (\$4)
	JAL-W-RS	jal	D	RS	jal loop nop nop lw \$5, 0 (\$31)
	R-W-RS	addu	E	RS	addu \$4, \$4, \$5 nop lw \$5, 0 (\$4) nop
	I-W-RS	ori	E	RS	ori \$4, \$5, 0xffff nop lw \$5, 0 (\$4) nop
	LD-W-RS	lw	E	RS	lw \$4, 0 (\$5) nop lw \$5, 0 (\$4) nop
	JAL-W-RS	jal	E	RS	jal loop nop lw \$5, 0 (\$31) nop
	R-M-RS	addu	E	RS	addu \$4, \$4, \$5 lw \$5, 0 (\$4) nop
	I-M-RS	ori	E	RS	ori \$4, \$5, 0xffff lw \$5, 0 (\$4) nop
	JAL-M-RS	jal	E	RS	jal loop lw \$5, 0 (\$31) nop
ST 型	R-W-RS	addu	D	RS	addu \$4, \$4, \$5 nop nop sw \$5, 0 (\$4)
	I-W-RS	ori	D	RS	ori \$4, \$5, 0xffff nop nop sw \$5, 0 (\$4)
	LD-W-RS	lw	D	RS	lw \$4, 0 (\$5)

					nop nop sw \$5, 0 (\$4)
	JAL-W-RS	jal	D	RS	jal loop nop nop sw \$5, 0 (\$31)
	R-W-RT	addu	D	RT	addu \$4, \$4, \$5 nop nop sw \$4, 0 (\$5)
	I-W-RT	ori	D	RT	ori \$4, \$5, 0xffff nop nop sw \$4, 0 (\$5)
	LD-W-RT	lw	D	RT	lw \$4, 0 (\$5) nop nop sw \$4, 0 (\$6)
	JAL-W-RT	jal	D	RT	jal loop nop nop sw \$31, 0 (\$5)
	R-M-RS	addu	E	RS	addu \$4, \$4, \$5 sw \$5, 0 (\$4) nop
	I-M-RS	ori	E	RS	ori \$4, \$4, \$5 sw \$5, 0 (\$4) nop
	JAL-M-RS	jal	E	RS	jal loop sw \$5, 0 (\$31) nop
	R-W-RS	addu	E	RS	addu \$4, \$4, \$5 nop sw \$5, 0 (\$4) nop
	I-W-RS	ori	E	RS	ori \$4, \$5, 0xffff nop sw \$5, 0 (\$4) nop
	LD-W-RS	lw	E	RS	lw \$4, 0 (\$5) nop sw \$5, 0 (\$4) nop

	JAL-W-RS	jal	E	RS	jal loop nop sw \$5, 0 (\$31) nop
	R-W-RT	addu	E	RT	addu \$4, \$4, \$5 nop sw \$4, 0 (\$5) nop
	I-W-RT	ori	E	RT	ori \$4, \$5, 0xffff nop sw \$4, 0 (\$5) nop
	LD-W-RT	lw	E	RT	lw \$4, 0 (\$5) nop sw \$4, 0 (\$6) nop
	JAL-W-RT	jal	E	RT	jal loop nop sw \$31, 0 (\$5) nop
	R-W-RT	addu	M	RT	addu \$4, \$4, \$5 sw \$4, 0 (\$5) nop nop
	I-W-RT	ori	M	RT	ori \$4, \$5, 0xffff sw \$4, 0 (\$5) nop nop
	LD-W-RT	lw	M	RT	lw \$4, 0 (\$5) sw \$4, 0 (\$6) nop nop
	JAL-W-RT	jal	M	RT	jal loop sw \$31, 0 (\$5) nop nop
JR	R-W-RS	addu	D	RS	addu \$4, \$4, \$5 nop nop jr \$4
	I-W-RS	ori	D	RS	ori \$4, \$5, 0xffff nop nop jr \$4

	LD-W-RS	lw	D	RS	lw \$4, 0 (\$5) nop nop jr \$4
	JAL-W-RS	jal	D	RS	jal loop nop nop jr \$31
	R-M-RS	addu	D	RS	addu \$4, \$4, \$5 nop jr \$4
	I-M-RS	ori	D	RS	ori \$4, \$5, 0xffff nop jr \$4
	JAL-M-RS	jal	D	RS	jal loop nop jr \$31
	JAL-E-RS	jal	D	RS	jal loop jr \$31
B 型	R-W-RS	addu	D	RS	addu \$4, \$4, \$5 nop nop beq \$4, \$3, loop
	I-W-RS	ori	D	RS	ori \$4, \$5, 0xffff nop nop beq \$4, \$3, loop
	LD-W-RS	lw	D	RS	lw \$4, 0 (\$5) nop nop beq \$4, \$3, loop
	JAL-W-RS	jal	D	RS	jal loop nop nop beq \$31, \$3, loop
	R-M-RS	addu	D	RS	addu \$4, \$4, \$5 nop beq \$4, \$3, loop
	I-M-RS	ori	D	RS	ori \$4, \$5, 0xffff nop beq \$4, \$3, loop
	JAL-M-RS	jal	D	RS	jal loop nop beq \$31, \$3, loop

	JAL-E-RS	jal	D	RS	jal loop beq \$31,\$3,loop
	R-W-RT	addu	D	RT	addu \$4,\$4,\$5 nop nop beq \$3,\$4,loop
	I-W-RT	ori	D	RT	ori \$4,\$5,0xffff nop nop beq \$3,\$4,loop
	LD-W-RT	lw	D	RT	lw \$4,0(\$5) nop nop beq \$3,\$4,loop
	JAL-W-RT	jal	D	RT	jal loop nop nop beq \$3,\$31,loop
	R-M-RT	addu	D	RT	addu \$4,\$4,\$5 nop beq \$3,\$4,loop
	I-M-RT	ori	D	RT	ori \$4,\$5,0xffff nop beq \$3,\$4,loop
	JAL-M-RT	jal	D	RT	jal loop nop beq \$3,\$31,loop
	JAL-E-RT	jal	D	RT	jal loop beq \$3,\$31,loop