ThinPad设计概要

计05 许欣然 邹林希

# 概述

本次大实验计划设计支持指令流水的CPU。预计扩展旁路回路解决数据冲突，增加冒险控制单元解决控制冲突，扩展VGA、PS2。

# 数据通路

详见附件中final.xps文件。

# 指令执行流程

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | IF | ID | EXE | MEM | WB |
| ADDIU rx imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨 sign\_extend(imm) | C🡨A+B |  | rx🡨C |
| ADDIU3 rx ry imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨 sign\_extend(imm) | C🡨A+B |  | ry🡨C |
| ADDSP imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨 sign\_extend(imm)  B🡨SP | C🡨A+B |  | SP🡨C |
| ADDU rx ry rz | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨ry | C🡨A+B |  | rz🡨C |
| AND rx ry | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨ry | C🡨A&B |  | rx🡨C |
| B imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨sign\_extend(imm)  B🡨PC+A  PC🡨B |  |  |  |
| BEQZ rx imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨sign\_extend(imm)  C🡨PC+B  PC🡨C(依情况) |  |  |  |
| BNEZ rx imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨sign\_extend(imm)  C🡨PC+B  PC🡨C(依情况) |  |  |  |
| BTEQZ imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨sign\_extend(imm)  B🡨PC+A  PC🡨B(依情况) |  |  |  |
| CMP rx ry | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨ry | C🡨A-B  依情况定T |  |  |
| JR rx | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  PC🡨A |  |  |  |
| LI rx imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨sign\_extend(imm)  B🡨0 | C🡨A+B |  | rx🡨C |
| LW rx ry imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨sign\_extend(imm) | C🡨A+B | D🡨MEM[C] | ry🡨D |
| LW\_SP rx imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨sign\_extend(imm)  B🡨SP | C🡨A+B | D🡨MEM[C] | rx🡨D |
| MFIH rx | IR🡨Mem[PC]  PC🡨PC+1 | A🡨IH  B🡨0 | C🡨A+B |  | rx🡨C |
| MFPC rx | IR🡨Mem[PC]  PC🡨PC+1 | A🡨PC  B🡨0 | C🡨A+B |  | rx🡨C |
| MTIH rx | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨0 | C🡨A+B |  | IH🡨C |
| MTSP rx | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨0 | C🡨A+B |  | SP🡨C |
| NOP | IR🡨Mem[PC]  PC🡨PC+1 |  |  |  |  |
| OR rx ry | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨ry | C🡨A|B |  | rx🡨C |
| SLL rx ry imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨ry  B🡨sign\_extend(imm) | C🡨A<<B  (ALU判断B=0时C🡨A<<8) |  | rx🡨C |
| SRA rx ry imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨ry  B🡨sign\_extend(imm) | C🡨A>>B  (arith)  (ALU判断B=0时C🡨A>>8 (arith)) |  | rx🡨C |
| SUBU rx ry rz | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨ry | C🡨A-B |  | rz🡨C |
| SW rx ry imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨ry  C🡨sign\_extend(imm) | D🡨A+C | MEM[D]🡨B |  |
| SW\_SP rx imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨sign\_extend(imm)  C🡨SP | D🡨C+B | MEM[D]🡨A |  |
| ADDSP3 rx imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨sign\_extend(imm)  B🡨SP | C🡨A+B |  | rx🡨C |
| BTNEZ imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨sign\_extend(imm)  B🡨PC+A  PC🡨B(依情况) |  |  |  |
| SLTI rx imm | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨sign\_extend(imm) | 依情况定T |  |  |
| SLT rx ry | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨ry | 依情况定T |  |  |
| SRAV rx ry | IR🡨Mem[PC]  PC🡨PC+1 | A🡨rx  B🡨ry | C🡨B>>A(arith) |  | ry🡨C |

# 控制信号

控制信号一共有11个，不同值对应的含义如下表，其中RegDst、ALUSrcA、ALUSrcB和Branch信号都有2位，ALUOP信号有3位。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 控制信号 |  | | | | | |
| RegDst(2) | 00:2~4位 | 01:5~7位 | 10:8~10位 | |  |  |
| ALUSrcA(2) | 00:RegA | 01：扩展 | 10：全0 | 11：PC |  |  |
| ALUSrcB(2) | 00：RegB | 01：扩展 | 10：全0 |  |  |  |
| MemRead | 0：X | 1：读内存 | |  |  |  |
| MemWrite | 0：X | 1：写内存 | |  |  |  |
| RegWrite | 0：X | 1：写入寄存器 | |  |  |  |
| MemtoReg | 0：ALU输出 | 1：内存输出 | |  |  |  |
| ALUOP(3) | 000:加 | 001:减 | 010:与 | 011:或 | 100:左移 | 101:右移 |
| TType | 大于等于 🡪1 | 等于 🡪 1 | |  |  |  |
| RegTWrite | 0:X | 1:写入 |  |  |  |  |
| Branch(2) | 00:PC+4 | 01:PC+4+IMM | 10:RegA |  |  |  |

各指令对应的控制信号如下表所示：

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ­­­ | RegDst（2） | ALUSrcA（） | ALUSrcB | MemRead | MemWrite | RegWrite | MemtoReg | ALUOP(3) | TType | RegTWrite | Branch(2) |
| ADDIU rx imm | 10 | 00 | 01 | 0 | 0 | 1 | 0 | 000 | X | 0 | 00 |
| ADDIU3 rx ry imm | 01 | 00 | 01 | 0 | 0 | 1 | 0 | 000 | X | 0 | 00 |
| ADDSP imm | 10 | 00 | 01 | 0 | 0 | 1 | 0 | 000 | X | 0 | 00 |
| ADDU rx ry rz | 00 | 00 | 00 | 0 | 0 | 1 | 0 | 000 | X | 0 | 00 |
| AND rx ry | 10 | 00 | 00 | 0 | 0 | 1 | 0 | 010 | X | 0 | 00 |
| B imm | XX | 00 | X | 0 | 0 | 0 | 0 | XXX | X | 0 | 01 |
| BEQZ rx imm | XX | 00 | X | 0 | 0 | 0 | 0 | XXX | X | 0 | 01 |
| BNEZ rx imm | XX | 00 | X | 0 | 0 | 0 | 0 | XXX | X | 0 | 01 |
| BTEQZ imm | XX | 00 | X | 0 | 0 | 0 | 0 | XXX | X | 0 | 01 |
| CMP rx ry | XX | 00 | 00 | 0 | 0 | 0 | 0 | 001 | 1 | 1 | 00 |
| JR rx | XX | 0X | X | 0 | 0 | 0 | 0 | XXX | X | 0 | 10 |
| LI rx imm | 10 | 00 | 01 | 0 | 0 | 1 | 1 | 000 | X | 0 | 00 |
| LW rx ry imm | 01 | 00 | 01 | 1 | 0 | 1 | 1 | 000 | X | 0 | 00 |
| LW\_SP rx imm | 10 | 00 | 01 | 1 | 0 | 1 | 1 | 000 | X | 0 | 00 |
| MFIH rx | 10 | 00 | 10 | 0 | 0 | 1 | 0 | 000 | X | 0 | 00 |
| MFPC rx | 10 | 11 | 10 | 0 | 0 | 1 | 0 | 000 | X | 0 | 00 |
| MTIH rx | 10 | 00 | 10 | 0 | 0 | 1 | 0 | 000 | X | 0 | 00 |
| MTSP rx | 10 | 00 | 10 | 0 | 0 | 1 | 0 | 000 | X | 0 | 00 |
| NOP | X | 00 | X | 0 | 0 | 0 | 0 | X | X | 0 | 00 |
| OR rx ry | 10 | 00 | 00 | 0 | 0 | 1 | 0 | 011 | X | 0 | 00 |
| SLL rx ry imm | 10 | 01 | 00 | 0 | 0 | 1 | 0 | 100 | X | 0 | 00 |
| SRA rx ry imm | 10 | 01 | 00 | 0 | 0 | 1 | 0 | 101 | X | 0 | 00 |
| SUBU rx ry rz | 00 | 00 | 00 | 0 | 0 | 1 | 0 | 001 | X | 0 | 00 |
| SW rx ry imm | X | 00 | 01 | 0 | 1 | 0 | 0 | 000 | X | 0 | 00 |
| SW\_SP rx imm | X | 00 | 01 | 0 | 1 | 0 | 0 | 000 | X | 0 | 00 |
| ADDSP3 rx imm | 10 | 00 | 01 | 0 | 0 | 1 | 0 | 000 | X | 0 | 00 |
| BTNEZ imm | X | 00 | 01 | 0 | 0 | 0 | 0 | X | X | 0 | 01 |
| SLTI rx imm | X | 00 | 01 | 0 | 0 | 0 | 0 | 001 | 0 | 1 | 00 |
| SLT rx ry | X | 00 | 00 | 0 | 0 | 0 | 0 | 001 | 0 | 1 | 00 |
| SRAV rx ry | 01 | 00 | 00 | 0 | 0 | 1 | 0 | 101 | X | 0 | 00 |

# 模块接口

## 寄存器堆RegisterFile

Port ( SelectA : in STD\_LOGIC\_VECTOR (2 downto 0);

SelectB : in STD\_LOGIC\_VECTOR (2 downto 0);

RegA : out STD\_LOGIC\_VECTOR (15 downto 0);

RegB : out STD\_LOGIC\_VECTOR (15 downto 0);

SelectC : in STD\_LOGIC\_VECTOR (3 downto 0);

InputData : in STD\_LOGIC\_VECTOR (15 downto 0);

RegWrite : in STD\_LOGIC;

EqualFlag : out STD\_LOGIC);

## 控制器Controller

Port ( InstructionInput : in STD\_LOGIC\_VECTOR (15 downto 0);

RegDst : out STD\_LOGIC;

ALUOp : out STD\_LOGIC\_VECTOR (1 downto 0);

ALUSrc : out STD\_LOGIC;

RegTWrite : out STD\_LOGIC;

TType : out STD\_LOGIC;

MemRead : out STD\_LOGIC;

MemWrite : out STD\_LOGIC;

RegWrite : out STD\_LOGIC;

MemtoReg : out STD\_LOGIC);

## 状态寄存器

### IFID

Port ( InstructionInput : in STD\_LOGIC\_VECTOR (15 downto 0);

PCInput : in STD\_LOGIC\_VECTOR (15 downto 0);

InstructionOutput : out STD\_LOGIC\_VECTOR (15 downto 0);

PCOutput : out STD\_LOGIC\_VECTOR (15 downto 0);

clock : in STD\_LOGIC);

### IDEX

Port ( RegTWriteInput : in STD\_LOGIC;

RegDstInput : in STD\_LOGIC;

ALUOPInput : in STD\_LOGIC\_VECTOR (1 downto 0);

ALUSrcInput : in STD\_LOGIC;

TTypeInput : in STD\_LOGIC;

MemReadInput : in STD\_LOGIC;

MemWriteInput : in STD\_LOGIC;

RegWriteInput : in STD\_LOGIC;

MemtoRegInput : in STD\_LOGIC;

RegAInput : in STD\_LOGIC\_VECTOR (15 downto 0);

RegBInput : in STD\_LOGIC\_VECTOR (15 downto 0);

Instruction24Input : in STD\_LOGIC\_VECTOR (2 downto 0);

Instruction57Input : in STD\_LOGIC\_VECTOR (2 downto 0);

Instruction810Input : in STD\_LOGIC\_VECTOR (2 downto 0);

ExtendedAddressInput : in STD\_LOGIC\_VECTOR (15 downto 0);

RegTWriteOutput: out STD\_LOGIC;

RegDstOutput : out STD\_LOGIC;

ALUOPOutput : out STD\_LOGIC\_VECTOR (1 downto 0);

ALUSrcOutput : out STD\_LOGIC;

TTypeOutput : out STD\_LOGIC;

MemReadOutput : out STD\_LOGIC;

MemWriteOutput : out STD\_LOGIC;

RegWriteOutput : out STD\_LOGIC;

MemtoRegOutput : out STD\_LOGIC;

RegAOutput : out STD\_LOGIC\_VECTOR (15 downto 0);

RegBOutput : out STD\_LOGIC\_VECTOR (15 downto 0);

Instruction24Output : out STD\_LOGIC\_VECTOR (2 downto 0);

Instruction57Output : out STD\_LOGIC\_VECTOR (2 downto 0);

Instruction810Output : out STD\_LOGIC\_VECTOR (2 downto 0);

ExtendedAddressOutput : out STD\_LOGIC\_VECTOR (15 downto 0);

clock : in STD\_LOGIC );

### EX\_MEM

Port ( dataA : in STD\_LOGIC\_VECTOR (15 downto 0);

dataB : in STD\_LOGIC\_VECTOR (15 downto 0);

operation : in STD\_LOGIC\_VECTOR (3 downto 0);

result : out STD\_LOGIC\_VECTOR (15 downto 0);

zeroFlag : out STD\_LOGIC);

### MEM\_WB

Port ( RegWriteInput : in STD\_LOGIC;

MemtoRegInput : in STD\_LOGIC;

ALUDataInput : in STD\_LOGIC\_VECTOR (15 downto 0);

ReadDataInput : in STD\_LOGIC\_VECTOR (15 downto 0);

RegDestInput : in STD\_LOGIC\_VECTOR (2 downto 0);

RegWriteOutput : out STD\_LOGIC;

MemtoRegOutput : out STD\_LOGIC;

ALUDataOutput : out STD\_LOGIC\_VECTOR (15 downto 0);

ReadDataOutput : out STD\_LOGIC\_VECTOR (15 downto 0);

RegDestOutput : out STD\_LOGIC\_VECTOR (2 downto 0);

clock : in STD\_LOGIC);

### 符号扩展SignedExtend

Port ( Instruction : in STD\_LOGIC\_VECTOR (15 downto 0);

ExtendedAddress : out STD\_LOGIC\_VECTOR (15 downto 0);

Selector : in STD\_LOGIC\_VECTOR (1 downto 0));

### 加法器Adder16

Port ( InputA : in STD\_LOGIC\_VECTOR (15 downto 0);

InputB : in STD\_LOGIC\_VECTOR (15 downto 0);

Output : out STD\_LOGIC\_VECTOR (15 downto 0));

## 存储器

### 指令存储器InstructionMemory

Port ( Address : in STD\_LOGIC\_VECTOR (15 downto 0);

DataOutput : out STD\_LOGIC\_VECTOR (15 downto 0));

### 数据存储器

Port ( address : in STD\_LOGIC\_VECTOR (15 downto 0);

dataInput : in STD\_LOGIC\_VECTOR (15 downto 0);

output : out STD\_LOGIC\_VECTOR (15 downto 0);

MemWrite : in STD\_LOGIC;

MemRead : in STD\_LOGIC );

### 选择器Mux16

Port ( InputA : in STD\_LOGIC\_VECTOR (15 downto 0);

InputB : in STD\_LOGIC\_VECTOR (15 downto 0);

InputC : in STD\_LOGIC\_VECTOR (15 downto 0);

InputD : in STD\_LOGIC\_VECTOR (15 downto 0);

control1 : in STD\_LOGIC\_VECTOR(1 downto 0);

Output : out STD\_LOGIC\_VECTOR (15 downto 0));

### 16位寄存器Reg16

Port ( input : in STD\_LOGIC\_VECTOR (15 downto 0);

output : out STD\_LOGIC\_VECTOR (15 downto 0);

clk : in STD\_LOGIC);

## ALU

Port ( dataA : in STD\_LOGIC\_VECTOR (15 downto 0);

dataB : in STD\_LOGIC\_VECTOR (15 downto 0);

operation : in STD\_LOGIC\_VECTOR (3 downto 0);

result : out STD\_LOGIC\_VECTOR (15 downto 0);

zeroFlag : out STD\_LOGIC);

## 旁路控制器bypassControl

Port ( rsInput : in STD\_LOGIC\_VECTOR (2 downto 0);

rtInput : in STD\_LOGIC\_VECTOR (2 downto 0);

EX\_MEMRdInput : in STD\_LOGIC\_VECTOR (2 downto 0);

Mem\_WBInput : in STD\_LOGIC\_VECTOR (2 downto 0);

EX\_MEMRegWriteInput : in STD\_LOGIC;

MEM\_WBRegWriteInput : in STD\_LOGIC;

rsSelectOutput : out STD\_LOGIC\_VECTOR (1 downto 0);

rtSelectOutput : out STD\_LOGIC\_VECTOR (1 downto 0));

## T寄存器控制器Tcontrol

Port ( RegTWrite : in STD\_LOGIC;

TType : in STD\_LOGIC;

ALUResult : in STD\_LOGIC\_VECTOR (15 downto 0);

clock : in STD\_LOGIC;

T : out STD\_LOGIC);

## 分支选择器BranchSelector

Port ( TInput : in STD\_LOGIC;

RegAInput : in STD\_LOGIC\_VECTOR (15 downto 0);

InstInput : in STD\_LOGIC\_VECTOR (15 downto 0);

output : out STD\_LOGIC\_VECTOR (1 downto 0));

# 调用关系

详见数据通路。