

# 单周期CPU设计与实现

---

薛睿、仇洁婷、郑海刚



# 实验目的



理解单周期CPU工作过程；

理解指令存储器和数据存储器的哈佛结构存储；

熟悉MIPS指令集；

掌握单周期CPU设计与实现方法。

# 实验题目

实现MIPS指令集的单周期CPU设计。如下表24条必做、7条选做。

指令类型	指令	分数等级	指令类型	指令	分数等级
R型指令	add	必做	I型指令	addi	必做
	addu	必做		addiu	必做
	sub	必做		andi	必做
	subu	必做		ori	必做
	and	必做		xori	必做
	or	必做		sltiu	必做
	xor	必做		lui	必做
	nor	必做		lw	必做
	slt	选做		sw	必做
	sltu	选做		beq	必做
	sll	选做		bne	选做
	srl	选做		bgtz	选做
	sra	选做	J型指令	j	必做
	sllv	必做		jal	必做
	srlv	必做			必做
	srav	必做			
	jr	必做			

# 实验原理-CPU开发基本过程

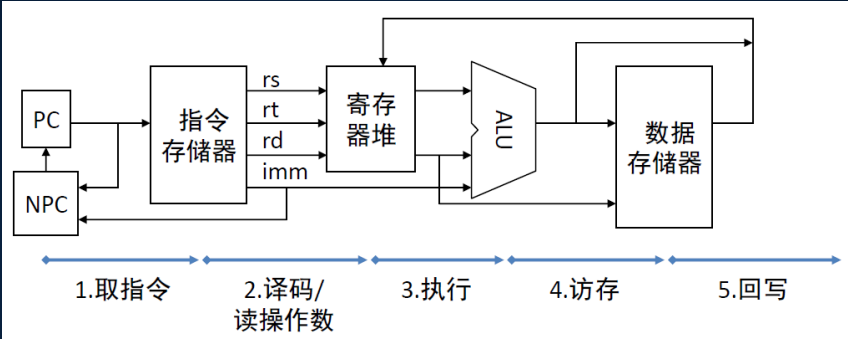
- 1) 构造功能部件。构造一组必要的通路功能部件。对于每个功能部件，都需要定义其功能及接口信号，并用HDL描述其内部具体的电路行为或结构。
- 2) 构造指令级别的通路与控制信号取值。根据指令操作语义构造指令级别的通路与控制信号取值。
- 3) 综合通路。当所有指令级别的通路构造完毕后，将其综合为完整通路。
- 4) 综合控制器。将全部指令对应的控制信号取值均建立后，合并所有的控制信号取值形成控制信号矩阵，并生成相应的控制信号表达式。
- 5) 生成工程项目。将功能第2步和第3步的结果翻译为HDL语言，并在顶层工程文件中组装通路与控制器。

通路

控制器

# 实验原理-构造数据通路

必要的功能部件

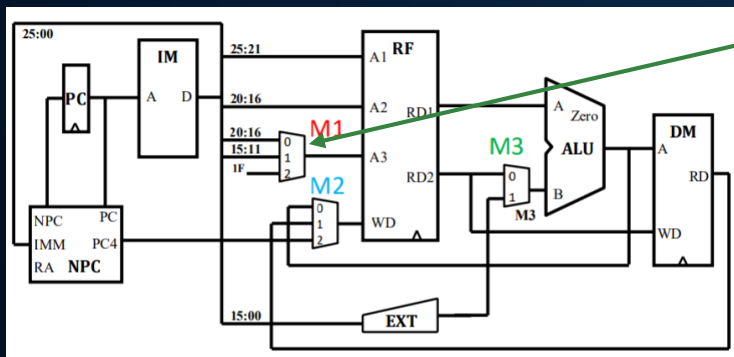


构造数据通路

部件	PC	NPC		IM (指令存储器)		RF (寄存器堆)				S_EXT	ALU	
输入信号	DI	PC	Imm	RA (寄存器)	A	A1 (读出1)	A2 (读出2)	A3 (写入)	WD (写回值)		A	B
addu	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]	IM.D[20:16]	IM.D[15:11]	ALU.C		RF.RD1	RF.RD2
subu	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]	IM.D[20:16]	IM.D[15:11]	ALU.C		RF.RD1	RF.RD2
jr	NPC.NPC	PC.DO		RF.RD1	PC.DO	IM.D[25:21]						
ori	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]		IM.D[20:16]	ALU.C	IM.D[15.0]	RF.RD1	EXT.Ext
lw	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]		IM.D[20:16]	DM.RD	IM.D[15.0]	RF.RD1	EXT.Ext
sw	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]		IM.D[20:16]		IM.D[15.0]	RF.RD1	EXT.Ext
beq	NPC.NPC	PC.DO	IM.D[15:0]		PC.DO	IM.D[25:21]	IM.D[20:16]				RF.RD1	RF.RD2
jal	NPC.NPC	PC.DO	IM.D[25:0]		PC.DO			0x1F	NPC.PC4			
完整	NPC.NPC	PC.DO	IM.D[25:0]	RF.RD1	PC.DO	IM.D[25:21]	IM.D[20:16]	IM.D[15:11]	ALU.C	IM.D[15.0]	RF.RD1	RF.RD2
								0x1F	NPC.PC4			

# 实验原理-综合数据通路

部件	PC	NPC		IM (指令存储器)		RF (寄存器堆)				S_EXT	ALU	
输入信号	DI	PC	Imm	RA (寄存器)	A	A1 (读出1)	A2 (读出2)	A3 (写入)	WD (写回值)		A	B
addu	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]	IM.D[20:16]	IM.D[15:11]	ALU.C		RF.RD1	RF.RD2
subu	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]	IM.D[20:16]	IM.D[15:11]	ALU.C		RF.RD1	RF.RD2
jr	NPC.NPC	PC.DO		RF.RD1	PC.DO	IM.D[25:21]						
ori	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]		IM.D[20:16]	ALU.C	IM.D[15:0]	RF.RD1	EXT.Ext
lw	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]		IM.D[20:16]	DM.RD	IM.D[15:0]	RF.RD1	EXT.Ext
sw	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]		IM.D[20:16]		IM.D[15:0]	RF.RD1	EXT.Ext
beq	NPC.NPC	PC.DO	IM.D[15:0]		PC.DO	IM.D[25:21]	IM.D[20:16]				RF.RD1	RF.RD2
jal	NPC.NPC	PC.DO	IM.D[25:0]		PC.DO			0x1F	NPC.PC4			
完整	NPC.NPC	PC.DO	IM.D[25:0]	RF.RD1	PC.DO	IM.D[25:21]	IM.D[20:16]	IM.D[15:11]	ALU.C	IM.D[15:0]	RF.RD1	EXT.Ext
								0x1F	NPC.PC4			



部件、输入、输出

# 实验原理-添加控制信号

所属单元	取指单元					译码单元					执行单元		数据存储器
部件	PC	NPC			IM	RF ( 寄存器堆)				S_EXT	ALU		DM
输入信号	DI	PC	Imm	RA ( 寄存器)	A	A1 ( 读出1)	A2 ( 读出2)	A3 ( 写入)	WD ( 写回值)		A	B	A
addu	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]	IM.D[20:16]	IM.D[15:11]	ALU.C		RF.RD1	RF.RD2	
subu	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]	IM.D[20:16]	IM.D[15:11]	ALU.C		RF.RD1	RF.RD2	
jr	NPC.NPC	PC.DO		RF.RD1	PC.DO	IM.D[25:21]							
ori	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]		IM.D[20:16]	ALU.C	IM.D[15:0]	RF.RD1	EXT.Ext	
lw	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]		IM.D[20:16]	DM.RD	IM.D[15:0]	RF.RD1	EXT.Ext	ALU.C
sw	NPC.NPC	PC.DO			PC.DO	IM.D[25:21]		IM.D[20:16]		IM.D[15:0]	RF.RD1	EXT.Ext	ALU.C
beq	NPC.NPC	PC.DO	IM.D[15:0]		PC.DO	IM.D[25:21]	IM.D[20:16]				RF.RD1	RF.RD2	
jal	NPC.NPC	PC.DO	IM.D[25:0]		PC.DO			0x1F	NPC.PC4				
完整	NPC.NPC	PC.DO	IM.D[25:0]	RF.RD1	PC.DO	IM.D[25:21]	IM.D[20:16]	IM.D[15:11]	ALU.C	IM.D[15:0]	RF.RD1	EXT.Ext	ALU.C
								IM.D[20:16]	DM.RD			RF.RD2	
								0x1F	NPC.PC4				
功能选择信号	无选择	NPCOp			无选择	无选择	无选择		RFWr	EXTOp	ALUOp		DMWr
多路选择器信号								WRSel	WDSel			BSel	无选择
返回信号													

输入：opcode/funct

状态信号

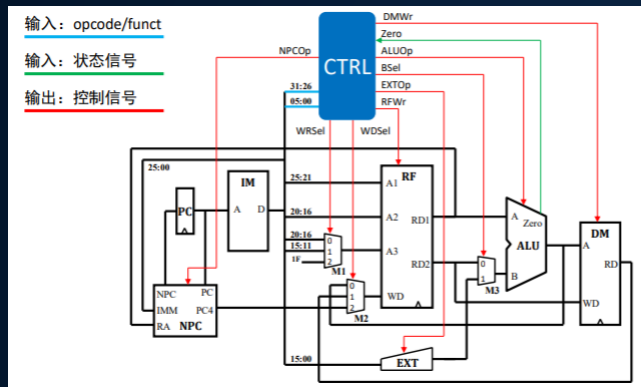
输出：控制信号

功能选择信号

多路选择器信号

返回信号

# 实验原理-综合控制信号



输入: opcode/funct

状态信号

输出: 控制信号

确定控制信号的取值

控制信号取值矩阵 (示例)

指令	NPCOp	RFWr	EXTOp	ALUOp	DMWr	WRSel	WDSel	Bsel	ZeroG
addu	PC+4	1	X	ADD	0	00	00	0	0
subu	PC+4	1	X	SUB	0	00	00	0	0
jr	0b10	0	X	X	0	XX	XX	X	0
ori	PC+4	1	0	OR	0	01	00	1	0
lw	PC+4	1	1	ADD	0	01	01	1	0
sw	PC+4	0	1	ADD	1	01	XX	1	0
beq	0b01	0	X	SUB	0	XX	XX	0	1
jal	0b11	1	X	X	0	XX	10	X	0



# 实验原理-实现

```
module minisys(rst,clk,debug信号);
```

```
    input rst;           //板上的Reset信号，低电平复位
```

```
    input clk;           //板上的100MHz时钟信号
```

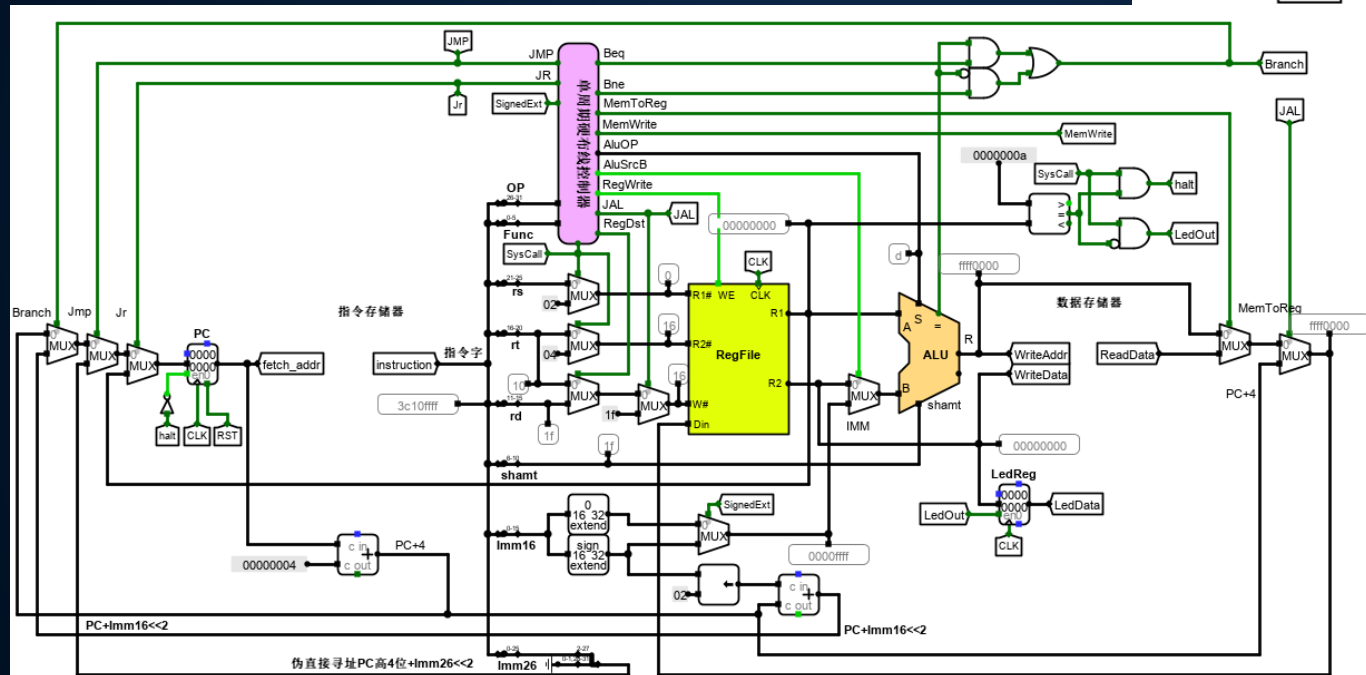
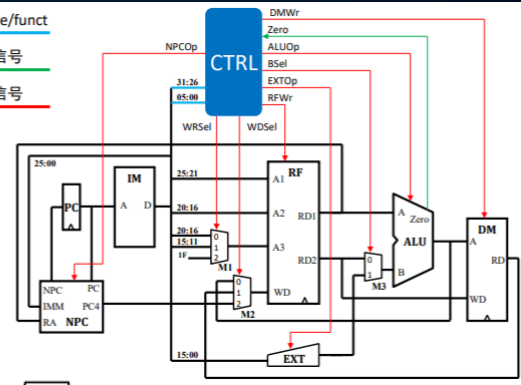
模块	功能	文件
顶层模块	实例化、连接各部件	minisys.v
时钟	23MHz	cpuclk.v
指令存储器模块	64KB	programrom.v
数据存储器模块	64KB	dmemory32.v
执行模块		executs32.v
取指模块		ifetc32.v
译码模块（含寄存器组）		idecode32.v
控制器模块		control32.v

# 实验原理-设计思考

输入: opcode/funct

输入: 状态信号

输出: 控制信号



# 实验原理-debug信号

信号	位宽	输入/输出	描述
debug_wb_pc	<b>32</b>	<b>output</b>	查看PC的值，连接PC寄存器
debug_wb_rf_wen	<b>1</b>	<b>output</b>	查看寄存器堆的写使能，连接RFWr
debug_wb_rf_wnum	<b>5</b>	<b>output</b>	查看寄存器堆的目的寄存器号，连接目的寄存器A3
debug_wb_rf_wdata	<b>32</b>	<b>output</b>	查看寄存器堆的写数据，连接WD

# 实验步骤

- 1) 根据给出的数据通路表示例，完成数据通路表。画出数据通路图。
- 2) 根据给出的控制信号表示例，完成控制信号表。在数据通路图上添加控制信号，完成单周期CPU设计结构图。
- 3) 划分模块，使用Verilog语言描述实现各模块。
- 4) 编写仿真程序，测试设计正确性，至少完成2个模块的仿真程序，代码和截图加入实验报告。
- 5) 设计优化。

# 提交作业



单周期CPU设计表（6月5日24点前）

程序