

# 华北电力大学

## 计算机组成与结构设计报告

(2021 -- 2022 年度 第二学期)

题    目\_\_\_\_\_16 位模型机的设计\_\_\_\_\_

班    级\_\_\_\_\_软件工程 2001\_\_\_\_\_

学    号\_\_\_\_\_220201090123\_\_\_\_\_

姓    名\_\_\_\_\_杨照\_\_\_\_\_

课 序 号\_\_\_\_\_2\_\_\_\_\_

成    绩\_\_\_\_\_

2022 年 5 月

## 目录

1.引言.....	3
1.1 设计目的.....	3
1.2 设计任务.....	3
2.详细设计及实现.....	3
2.1 拟定指令系统.....	3
2.2 确定总体结构.....	5
2.3 安排时序.....	7
2.4 拟定指令流程和微命令序列.....	8
3.总结.....	14
附录.....	15

# 1.引言

通过本学期对计算机组成与结构课程的学习，我了解并掌握了计算机系统的基本组成、运算方法与运算器、存储系统、指令系统、中央控制器、总线系统和输入输出系统，通过学习，我对计算机组成结构有了一个基础的认识，也希望通过实践操作来夯实自己计算机组成与结构课程的学习成果，现按照试验任务书的相关要求进行实验，设计一个基于MIPS指令集的16位模型机并实现相关功能。

## 1.1 设计目的

- (1) 掌握计算机的基本组成和工作原理；
- (2) 掌握计算机主要功能部件的功能与设计方法；
- (3) 掌握MIPS指令集各类指令的功能和执行流程；
- (4) 综合运用所学知识，在掌握部件单元电路设计的基础上，进一步设计一台16位的CPU，建立计算机整机概念；
- (5) 通过本次设计，理论联系实际，进一步加强学生分析问题、解决问题的能力，提高自身硬件设计水平。

## 1.2 设计任务

设计一个基于MIPS指令集的16位模型机，完成功能：对内存中存放的一个数组，取出每个元素加2后，再存回原位置；

# 2.详细设计及实现

## 2.1 拟定指令系统

### 2.1.1 确定指令格式

R型指令

5	3	3	3	1	1
op	Rx	Ry	Rz	sa	funct
操作码	源操作数1	源操作数2	目的操作数	移位	功能

I型指令

5	3	3	5
op	Rs	Rt	immediate
操作码	源操作数	目的操作数	立即数

J型指令

5	11
op	immediate
操作码	立即数

## 2.1.2 通用寄存器

R0	R1	R2	R3	R4	R5	R6	R7
000	001	010	011	100	101	110	111

## 2.1.3 MIPS指令代码

假设数组存储在内存中地址2000H~2010H内，根据设计任务可写出相关代码如下

```

LI R0,2000H      # 保存起始地址
LI R1,10H        # 保存数组长度
LW R0,R2,0       # 加载元素
ADDIU R2,2       # 元素值+2
SW R0,R2,0       # 存储元素
ADDIU R0,1       # 访存地址+1
ADDIU R1,0FFH    # R1减一计数
BNEZ R1,0FFAH    # 若R1不为0则跳转到第三条指令

```

## 2.1.4 指令说明

根据汇编程序，共有5种指令，各指令详细说明如下：

LI指令（该指令为双字指令）

5	3	3	5
01101		Rt	

16
imm

指令格式：LI Rt, imm（16位）

指令功能：将立即数 imm加载到寄存器Rt中

LW指令

5	3	3	5
10011	Rs	Rt	imm

指令格式：LW Rs, Rt, imm

指令功能：将Rs的内容加 imm，根据结果访问内存，读取数据加载到Rt中

ADDIU指令

<b>5</b>	<b>3</b>	<b>8</b>
01001	Rs	imm

指令格式：ADDIU Rs, imm

指令功能：将Rs的值与立即数imm求和，结果存入Rs

SW指令

<b>5</b>	<b>3</b>	<b>3</b>	<b>5</b>
11011	Rs	Rt	imm

指令格式：SW Rs, Rt, imm

指令功能：将Rs的内容加 imm，根据结果访问内存，将Rt内容写入主存

BNEZ指令（该指令为双字指令）

<b>5</b>	<b>3</b>	<b>3</b>	<b>5</b>
00101	Rs		

<b>16</b>
imm

指令格式：BNEZ Rs, imm

指令功能：若Rs不为0，则PC与立即数imm求和后，结果加载到PC中

## 2.1.5 程序结构

指令地址	指令	指令机器码	说明
0000H	LI R0,2000H	68002000H	数组起始地址送R0
0002H	LI R1,10H	68200010H	数组长度送R1
0004H	LW R0,R2,0	9A00H	取数，存放在R2中
0005H	ADDIU R2,2	4A02H	加二
0006H	SW R0,R2,0	DA00H	将R2中的数存回原处
0007H	ADDIU R0,1	4802H	R0指向数组中的下一个数
0008H	ADDIU R1,0FFH	49FFH	R1减一计数
0009H	BNEZ R1, 0FFAH	2900FFF9H	若R1不为零则跳转执行第三条指令

## 2.2 确定总体结构

该模型机存储字长、指令字长、机器字长均为16位

## 2.2.1 寄存器

### a) 通用寄存器

R0 (000)、R1 (001)、R2 (010)、R3 (011)、R4 (100)、R5 (101)、R6 (110)、R7 (111)；

### b) 专用寄存器

暂存器C：暂存来自主存的源地址或源数据

暂存器D：暂存来自主存的目的地址或目的数据

指令寄存器PC：存放下条指令地址

指令寄存器IR：存放现行指令

地址寄存器MAR、数据寄存器MBR：实现CPU与主存的接口

标准寄存器PSW：当数据全零时寄存器数据为全1，否则全0

## 2.2.2 运算部件

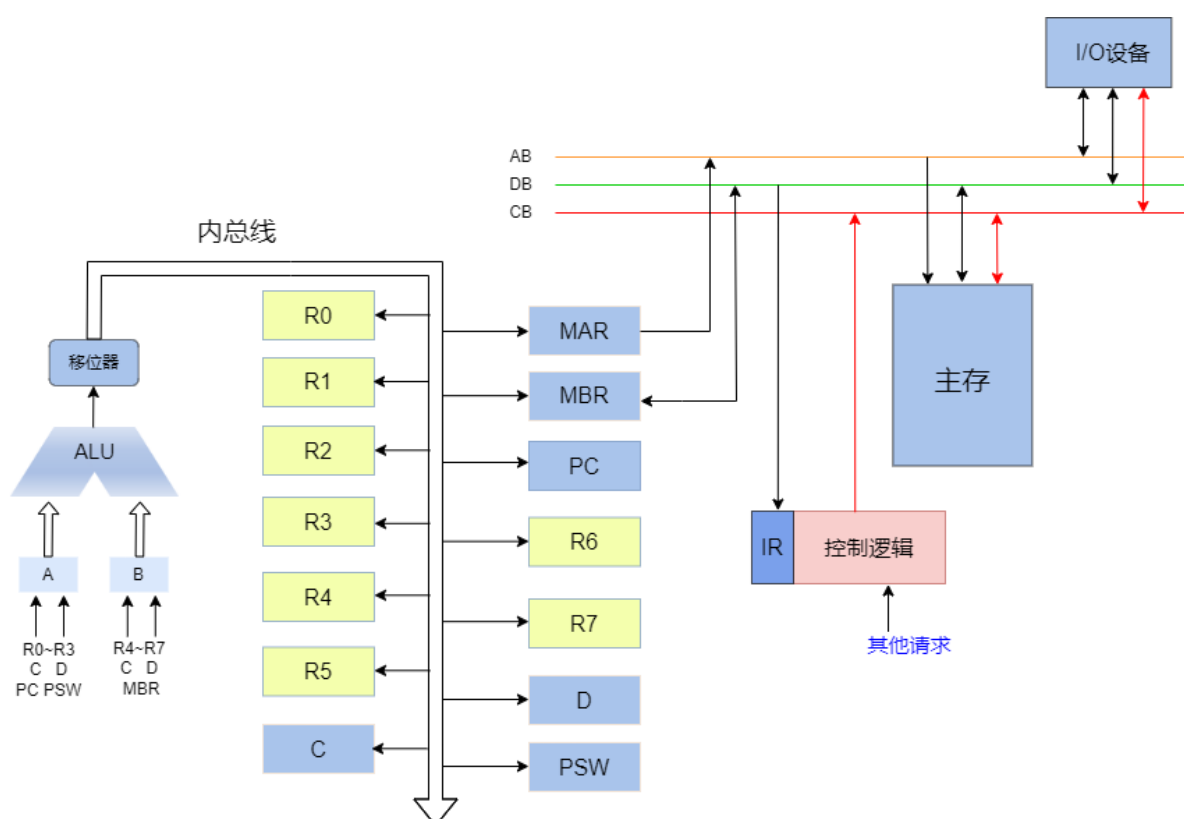
- 移位器：实现直送、左移、右移等功能
- ALU：SN74181 4片、SN74182 1片
- 选择器A、选择器B：选择数据来源

## 2.2.3 总线与数据通路

(1) 组成：ALU部件、寄存器组、内总线、地址总线、数据总线、控制总线

(2) 特点：IR寄存器直接与控制逻辑结合；ALU为内部数据传送通路中心；寄存器采用分立结构；内总线采用单向数据总线；MAR、MBR实现CPU与系统总线的连接

(3) 结构图



## 2.2.4 各类信息的传送路径

### (1) 指令信息

主存  $\xrightarrow{\text{根据 MAR, 读出}}$  DB  $\xrightarrow{\text{置于}}$  IR

### (2) 地址信息

- 取指地址

PC  $\rightarrow$  选择器 A  $\rightarrow$  ALU  $\rightarrow$  移位器  $\rightarrow$  内总线  $\xrightarrow{\text{打入}}$  MAR

- 后继指令信息

PC  $\rightarrow$  选择器 A  $\rightarrow$  ALU  $\xrightarrow{+X}$  移位器  $\rightarrow$  内总线  $\xrightarrow{\text{打入}}$  PC

- 操作数地址

寄存器  $\rightarrow$  选择器 A/B  $\rightarrow$  ALU  $\rightarrow$  移位器  $\rightarrow$  内总线  $\xrightarrow{\text{打入}}$  MAR

### (3) 数据信息

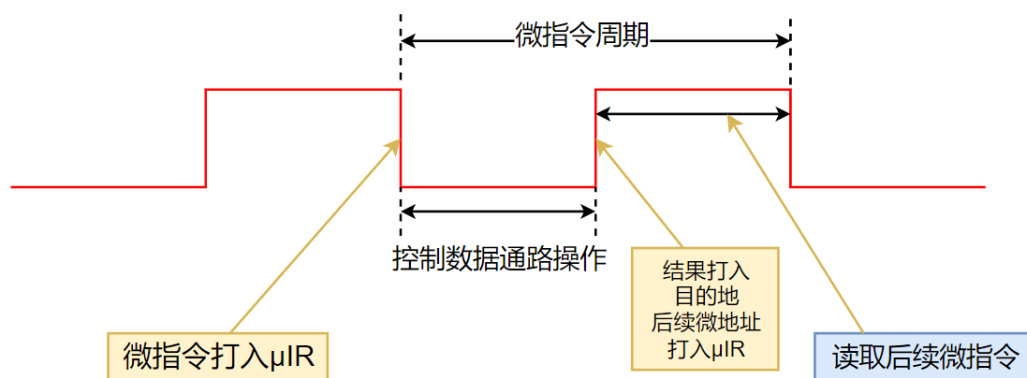
$R_i \rightarrow R_j$ :  $R_i \rightarrow$  A/B  $\rightarrow$  ALU  $\rightarrow$  移位器  $\rightarrow$  内总线  $\xrightarrow{\text{打入}}$   $R_j$

$R_i \rightarrow M$ :  $R_i \rightarrow$  A/B  $\rightarrow$  ALU  $\rightarrow$  移位器  $\rightarrow$  内总线  $\xrightarrow{\text{打入}}$  MBR  $\rightarrow$  DB  $\rightarrow$  M

$M \rightarrow R_j$ :  $M \rightarrow$  DB  $\rightarrow$  MBR  $\rightarrow$  B  $\rightarrow$  ALU  $\rightarrow$  移位器  $\rightarrow$  内总线  $\xrightarrow{\text{打入}}$   $R_j$

## 2.3 安排时序

---



## 2.4拟定指令流程和微命令序列

指令按 取指 (FT) →译码 (CT) →执行 (ET) →访存 (VT) →写回 (WT) 等五个周期进行

### 2.4.1 信号定义

- 写入通用寄存器Rx信号 (x=0, 1, 2, ..., 7) : CPR0~CPR7
- MAR使能端信号: EMAR
- 经内总线打入MAR信号: CPMAR
- 经内总线打入MBR信号: CPMBR
- MBR写入信号, 写入来自主存的数据: SMBR
- IR写入信号, 写入来自主存的指令码: SIR
- 移位器直传, 左移, 右移选择信号: DM、TL、TR
- 寄存器C、D、PSW、PC打入信号: CPC、CPD、CPPSW、CPPC

### 2.4.1 指令流程

#### 1、LI R0,2000H

工作周期	指令流程	控制信号
FT0	M→IR, PC+1→PC	EMAR,R,SIR,PC→A,Y=A+1,DM,CPPC
ET0	PC→MAR	PC→A,Y=A,DM,CPMAR
ET1	PC+1→PC	PC→A,Y=A+1,DM,CPPC
ET2	M→MBR	EMAR,R,SMBR
WT0	MBR→R0	MBR→B,Y=B,DM,CPR0
WT1	PC→MAR	PC→A,Y=A,DM,CPMAR

#### 2、LI R1,10H



工作周期	指令流程	控制信号
FT0	$M \rightarrow IR, PC+1 \rightarrow PC$	EMAR,R,SIR, $PC \rightarrow A, Y=A+1, DM, CPPC$
ET0	$PC \rightarrow MAR$	$PC \rightarrow A, Y=A, DM, CPMAR$
ET1	$PC+1 \rightarrow PC$	$PC \rightarrow A, Y=A+1, DM, CPPC$
ET2	$M \rightarrow MBR$	EMAR,R,SMBR
WT0	$MBR \rightarrow R1$	$MBR \rightarrow B, Y=B, DM, CPR1$
WT1	$PC \rightarrow MAR$	$PC \rightarrow A, Y=A, DM, CPMAR$

### 3、LW R0,R2,0

工作周期	指令流程	控制信号
FT0	$M \rightarrow IR, PC+1 \rightarrow PC$	EMAR,R,SIR, $PC \rightarrow A, Y=A+1, DM, CPPC$
ET0	$R0 \rightarrow MAR$	$R0 \rightarrow A, Y=A, DM, CPMAR$
VT0	$M \rightarrow MBR$	EMAR,R,SMBR
WT0	$MBR \rightarrow R2$	$MBR \rightarrow B, Y=B, DM, CPR2$
WT1	$PC \rightarrow MAR$	$PC \rightarrow A, Y=A, DM, CPMAR$

### 4、ADDIU R2,2

工作周期	指令流程	控制信号
FT0	$M \rightarrow IR, PC+1 \rightarrow PC$	EMAR,R,SIR, $PC \rightarrow A, Y=A+1, DM, CPPC$
ET0	$R2+1 \rightarrow C$	$R2 \rightarrow A, Y=A+1, DM, CPC$
WT0	$C+1 \rightarrow R2$	$C \rightarrow A, Y=A+1, DM, CPR2$
WT1	$PC \rightarrow MAR$	$PC \rightarrow A, Y=A, DM, CPMAR$

### 5、SW R0,R2,0

工作周期	指令流程	控制信号
FT0	$M \rightarrow IR, PC+1 \rightarrow PC$	EMAR,R,SIR, $PC \rightarrow A, Y=A+1, DM, CPPC$
ET0	$R0 \rightarrow MAR$	$R0 \rightarrow A, Y=A, DM, CPMAR$
WT0	$R2 \rightarrow MBR$	$R2 \rightarrow A, Y=A, DM, CPMBR$
WT1	$MBR \rightarrow M$	EMAR,W
WT2	$PC \rightarrow MAR$	$PC \rightarrow A, Y=A, DM, CPPC$

## 6、ADDIU R0,1

工作周期	指令流程	控制信号
FT0	M→IR, PC+1→PC	EMAR,R,SIR,PC→A,Y=A+1,DM,CPPC
WT0	R0+1→R0	R0→A,Y=A+1,DM,CPR2
WT1	PC→MAR	PC→A,Y=A,DM,CPMAR

## 7、ADDIU R1,0FFH

工作周期	指令流程	控制信号
FT0	M→IR, PC+1→PC	EMAR,R,SIR,PC→A,Y=A+1,DM,CPPC
WT0	R1-1→R1	R1→A,Y=A-1,DM,CPR1
WT1	PC→MAR	PC→A,Y=A,DM,CPMAR

## 8、BNEZ R1,0FFAH

工作周期	指令流程	控制信号
FT0	M→IR, PC+1→PC	EMAR,R,SIR,PC→A,Y=A+1,DM,CPPC
ET0	PC→MAR	PC→A,Y=A,DM,CPMAR
ET1	PC+1→PC	PC→A,Y=A+1,DM,CPPC
VT0	M→MBR	EMAR,R,SMBR
WT0	MBR*PSW→C	MBR→B,PSW→A,Y=A+B,DM,CPC
WT1	PC+C→PC	PC→A,C→B,Y=A+B,CPPC
WT2	PC→MAR	PC→A,Y=A,DM,CPMAR

## 2.4.2 微命令序列

- 微命令格式

AI	BI	MSC	S	ZO	EMAR	SMBR	SIR	R	W	SC
4	4	6	2	4	1	1	1	1	1	3

- 微命令详解

1. AI: ALU的A输入端选择, 4位

0000: 无输入

0001: R0输入

0010: R1输入

0011: R2输入

0100: R3输入

- 0101: C输入
- 0110: D输入
- 0111: PC输入
- 1000: PSW输入

## 2. BI: ALU的B输入端选择, 4位

- 0000: 无输入
- 0001: R4输入
- 0010: R5输入
- 0011: R6输入
- 0100: R7输入
- 0101: C输入
- 0110: D输入
- 0111: MBR输入

## 3. MSC: ALU功能选择

即ALU功能选择信号 S3 S2 S1 S0 M Cn, 共六位, 具体功能表此处省略, 详见附录 (P15)

## 4. S: 移位器控制 (2位)

- 00: DM
- 01: TL
- 10: TR

## 5. ZO: 内总线输出分配 (4位)

- 0000: 无输出
- 0001: CPR0
- 0010: CPR1
- 0011: CPR2
- 0100: CPR3
- 0101: CPR4
- 0110: CPR5
- 0111: CPR6
- 1000: CPR7
- 1001: CPC
- 1010: CPD
- 1011: CPMAR
- 1100: CPMBR
- 1101: CPPC

## 6. EMAR: MAR读出信号 (1位)

- 0: 不传送地址
- 1: 地址 --> 主存

## 7. SMBR: MBR写入信号

- 0: 不写入
- 1: DB数据写入MBR

## 8. SIR: IR写入信号 (1位)

- 0: 不写入
- 1: DB数据写入IR

## 9. R: 主存读出信号 (1位)

- 0: 不读
- 1: 读出

## 10. W: 主存写入信号 (1位)

- 0: 不写
- 1: 写入

## 11. SC: 顺序控制字段 (3位)

- 000: 顺序执行
- 001: 转移 (高8位给出转移指令)
- 010: 按op分支转移
- 100: 返回首地址处执行

### • 各指令微码如下:

所有指令的FT0均相同, 所有设置为控存中的第一条微指令, 且后续各指令的微程序中均不含该项

工作周期	指令流程	微指令
FT0	M→IR, PC+1→PC	0111000011110100110110110000

### 1、LI R0,2000H

工作周期	指令流程	微指令
ET0	PC→MAR	0111000011110000101100000000
ET1	PC+1→PC	0111000011110100110100000000
ET2	M→MBR	0000000011111100000011010000
WT0	MBR→R0	0000011111110000000100000000
WT1	PC→MAR	0111000011110000101100000100

### 2、LI R1,10H

工作周期	指令流程	微指令
ET0	PC→MAR	0111000011110000101100000000
ET1	PC+1→PC	0111000011110100110100000000
ET2	M→MBR	0000000011111100000011010000
WT0	MBR→R1	0000011111110000001000000000
WT1	PC→MAR	0111000011110000101100000100

### 3、LW R0,R2,0

工作周期	指令流程	微指令
ET0	R0→MAR	0001000011110000101100000000
VT0	M→MBR	0000000011111100000011010000
WT0	MBR→R2	0000011111110000001100000000
WT1	PC→MAR	0111000011110000101100000100

### 4、ADDIU R2,2

工作周期	指令流程	微指令
ET0	R2+1→C	0011000011110100100100000000
WT0	C+1→R2	0101000011110100001100000000
WT1	PC→MAR	0111000011110000101100000100

### 5、SW R0,R2,0

工作周期	指令流程	微指令
ET0	R0→MAR	0001000011110000101100000000
WT0	R2→MBR	0011000011110000110000000000
WT1	MBR→M	0000000011111100000010001000
WT2	PC→MAR	0111000011110000101100000100

### 6、ADDIU R0,1

工作周期	指令流程	微指令
WT0	R0+1→R0	0001000011110100000100000000
WT1	PC→MAR	0111000011110000101100000100

## 7、ADDIU R1,0FFH

工作周期	指令流程	微指令
WT0	R1-1→R1	00100000000000000000000000000000
WT1	PC→MAR	0111000011110000101100000100

## 8、BNEZ R1,0FFAH

工作周期	指令流程	微指令
ET0	PC→MAR	0111000011110000101100000000
ET1	PC+1→PC	0111000011110100110100000000
VT0	M→MBR	00000000111111000000011010000
WT0	MBR*PSW→C	1000011111101000100100000000
WT1	PC+C→PC	0111010110110000110100000000
WT2	PC→MAR	0111000011110000101100000100

将所有微指令按序写入控存即可

## 3. 总结

本次设计任务综合了计算机组成结构几乎全部的知识点，是一次对综合能力的考察。为了完成本次作业，我做了充分的准备，反复浏览了教材、实验指导书及课程PPT，最终实现了基于MIPS指令系统的16位模型机设计。虽然最终结果可能依然存在瑕疵，但通过本次设计，我巩固了计算机组成与结构的基本知识，基本掌握了MIPS指令系统，也真正实现了学以致用、理论联系实际，增强了我分析问题、解决问题的能力，也让我对计算机有了更深一层的理解和认识，使我受益匪浅，对我今后的学习有了极大的帮助。

附录:

S3 S2 S1 S0	M=1 (逻辑运算)	M=0 (算术运算)、 Cn=0	M=0 (算术运算)、Cn=1
0000	$\overline{A}$	$A - 1$	$A$
0001	$\overline{AB}$	$AB - 1$	$AB$
0010	$\overline{A} + B$	$\overline{AB} - 1$	$A\overline{B}$
0011	1	-1	0
0100	$\overline{A + B}$	$A + (\overline{A + B})$	$A + (A + \overline{B}) + 1$
0101	$\overline{B}$	$AB + (\overline{A + B})$	$AB + (\overline{A + B}) + 1$
0110	$\overline{A \oplus B}$	$A - B - 1$	$A - B$
0111	$A + \overline{B}$	$A + \overline{B}$	$(A + \overline{B}) + 1$
1000	$\overline{A}B$	$A + (A + B)$	$A + (A + B) + 1$
1001	$A \oplus B$	$A + B$	$A + B + 1$
1010	$B$	$\overline{AB} + (A + B)$	$\overline{AB} + (A + B) + 1$
1011	$A + B$	$A + B$	$A + B + 1$
1100	0	$A + A$	$A + A + 1$
1101	$A\overline{B}$	$AB + A$	$AB + A + 1$
1110	$AB$	$\overline{AB} + A$	$\overline{AB} + A + 1$
1111	$A$	$A$	$A + 1$