

《计算机组成与结构》作业答案

第一章

$$\begin{aligned} 1.10 \quad \text{程序执行时间} &= 50000 \times 1 + 80000 \times 2 + 10000 \times 4 + 5000 \times 2 \\ &= 260000 \text{ (时钟周期)} \\ &= 260000 \times 2\text{nS} = 520\mu\text{S} \\ \text{CPI} &= [520000\text{nS} \div (50000 + 80000 + 10000 + 5000)] \div 2 = 1.79 \\ \text{速度} &= 278.8 \text{ MIPS} \end{aligned}$$

1.13 解：

因为总加速比为：

$$S_p = \frac{1}{(1 - \sum f_e) + (\sum f_e / r_e)} \quad (1-12)$$

$$1 / [(1 - 0.6 - X) + (0.01 + 0.01 + 0.05X)] = 10$$

$$1 = 4.2 - 9.5X$$

$$9.5X = 3.2$$

$$X = 33.7$$

计算结果要求 C 部件的执行时间为 33.7%

第二章

2.5

(1) $x_6=0$, $x_5 x_4 x_3 x_2 x_1 x_0$ 任意; $x_6 x_5 x_4 x_3 x_2 x_1 x_0 < 1000000$

(2) $x_6=1$ $x_5=0$, $x_4 x_3 x_2 x_1 x_0$ 任意; $1100000 > x_6 x_5 x_4 x_3 x_2 x_1 x_0 \geq 1000000$

2.9

(1) $[W]_{\text{补}} = [X]_{\text{原}} = [Y]_{\text{反}} = [Z]_{\text{移}} = 00\text{H}$
 $W = 0$, $X = +0$, $Y = +0$, $Z = -128$

(2) $[W]_{\text{补}} = [X]_{\text{原}} = [Y]_{\text{反}} = [Z]_{\text{移}} = 80\text{H}$
 $W = -128$, $X = -0$, $Y = -127$, $Z = 0$

(3) $[W]_{\text{补}} = [X]_{\text{原}} = [Y]_{\text{反}} = [Z]_{\text{移}} = \text{FFH}$
 $W = -1$, $X = -127$, $Y = -0$, $Z = +127$

2.21

设浮点数字长 16 位，基值为 2（以 2 为底）。其中阶码 6 位（含一位阶符），用移码表示；尾数 10 位（含一位数符），用补码表示。

(1) 求能表示的规格化浮点数的范围，填写下表，并与 16 位定点补码整数和定点补码小数的表示范围进行比较。

习题 2.21 附表

	阶码（十六进制）	尾数（十六进制）	真值（十进制）
最大正数	3F	7FC	$(1-2^{-9}) \cdot 2^{31}$

最小正数	00	400	2^{-33}
最大负数	00	BFC	$-(0.5+2^{-9}) \cdot 2^{-32}$
最小负数	3F	800	$-1 \cdot 2^{31}$

(2) 判断下列十进制数能否表示成此格式的规格化浮点数，若可以，请写出对应的码值。

① 3.14

$$3.14 = 11.001000111 \cdot 2^0 = 0.110010001 \cdot 2^2 = 100010 \quad 0110010001 \quad \text{损失精度}$$

② $-1917 = -11101111101.0 \cdot 2^0 = -111011111 \cdot 2^2 = 1000100001 \cdot 2^{11} = 1010111000100001$

由于尾数而损失了精度

③ $105/512 = 0.001101001 \cdot 2^0 = 0.110100100 \cdot 2^{-2} = 011110 \quad 0110100100 \quad \text{能准确表示}$

④ -10^{-6} 可以表示，会损失精度

⑤ 10^{10} 超出规定浮点数的范围，无法表示。

2.22.

$$(1) 5.3125 = 101.0101 \cdot 2^0 = 1.010101 \cdot 2^2$$

$$S=0 \quad 10000001 \quad 0101010000000000000000$$

$$(2) -365.59375 = -101101101.10011 \cdot 2^0 = -1.0110110110011 \cdot 2^8$$

$$S=1 \quad 10000111 \quad 011011011001100000000000$$

第 3 章

3.17.(1)

$$X=0.01001 \quad Y=-0.10111$$

$$[X]_{\text{补}} = 0.0100100 \quad [Y]_{\text{补}} = 1.0100100$$

$$[X]_{\text{补}} + [Y]_{\text{补}} = 1.1001000 \quad \text{无溢出}$$

(2)

$$X=0.10010 \quad Y=0.11000$$

$$[X]_{\text{补}} = 0.1001000 \quad [Y]_{\text{补}} = 0.1100000$$

$$[X]_{\text{补}} + [Y]_{\text{补}} = 1.0101000 \quad \text{溢出}$$

3.20.(1) $X = -0.1101$ $Y = +0.0110$

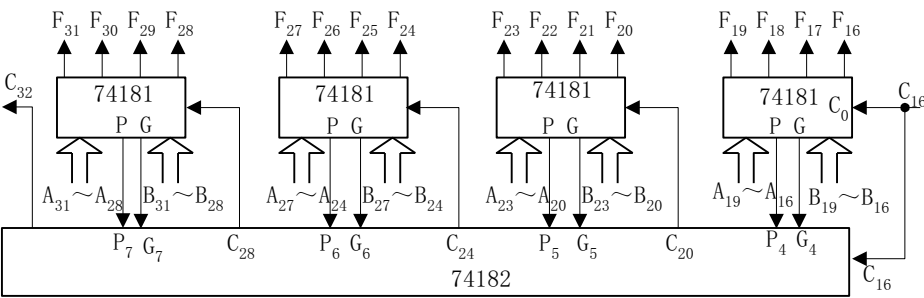
$[X]_{\text{补}} = 11.0011$ $[-X]_{\text{补}} = 00.1101$

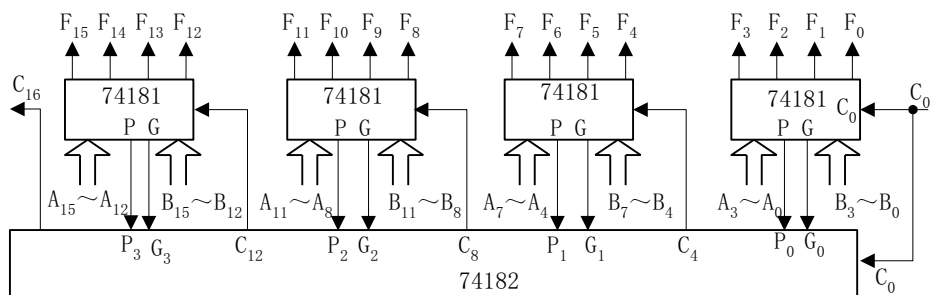
符号	D	A	A ₋₁	操 作
0 0	0 0 0 0	0 0 1 1 0	0	+0
0 0	0 0 0 0			
0 0	0 0 0 0	0 0 0 1 1	0	右移一位 + $[-X]_{\text{补}}$
0 0	0 0 0 0			
0 0	1 1 0 1			
0 0	1 1 0 1	1 0 0 0 1	1	右移一位 +0
0 0	0 1 1 0			
0 0	0 0 0 0			
0 0	0 1 1 0	0 1 0 0 0	0	右移一位 + $[X]_{\text{补}}$
0 0	0 0 1 1			
1 1	0 0 1 1			
1 1	0 1 1 0	0 0 1 0 0	0	右移一位 0
1 1	1 0 1 1			
0 0	0 0 0 0			
1 1	1 0 1 1	1 0 0 1 0		右移一位
1 1	1 1 0 1			

$[X \cdot Y]_{\text{补}} = 1.10110010$

$[Y]_{\text{补}} = 00.0110$

3.21.





3.22.(1) $X = -0.10101$ $Y = +0.11011$

$[X]_{\text{补}} = 11.01011$ $[-Y]_{\text{补}} = 11.00101$

$[Y]_{\text{补}} = 00.11011$

符号	被除数 (余数)	商	操作
0 0	1 0 1 0 1 0 0 0 0 0	0	左移一位
0 1	0 1 0 1 0 0 0 0 0 0		$- Y $
1 1	0 0 1 0 1		
0 0	0 1 1 1 1 0 0 0 0 0	1	$R \geq 0$, 商为1
0 0	1 1 1 1 0 0 0 0 0 1		左移一位
1 1	0 0 1 0 1		$- Y $
0 0	0 0 0 1 1 0 0 0 0 1	1	$R \geq 0$, 商为1
0 0	0 0 1 1 0 0 0 1 1		左移一位
1 1	0 0 1 0 1		$- Y $
1 1	0 1 0 1 1 0 0 1 1	0	$R < 0$, 商为0
1 0	1 0 1 1 0 0 1 1 0		左移一位
0 0	1 1 0 1 1		$+ Y $
1 1	1 0 0 0 1 0 1 1 0 0	0	$R < 0$, 商为0

$Z = -0.110001$; $Z = -0.1100$, 余 $= -0.10001 \times 2^{-4}$

3.25.

规格化浮点数：

非规格化浮点数：

$(1-2^{-8}) \cdot 2^{63}$
2^{-65}
$-(0.5+2^{-8}) \cdot 2^{-64}$
$-1 \cdot 2^{63}$

$(1-2^{-8}) \cdot 2^{63}$
$2^{-8} \cdot 2^{-64}$
$-2^{-8} \cdot 2^{-64}$
$-1 \cdot 2^{63}$

3.26.(1)

$X = 11/16 \times 2^{-4}$ $Y = 35/64 \times 2^{-3}$

$X = 0.101100 \times 2^{-4} = 01100 \quad 0101100$

$$Y = 0.100011 \times 2^{-3} = 01101$$

X 规格化表示为：01100 0101100

Y 规格化表示为：01101 0100011

对阶 X 表示为：01101 0010110

求 $X+Y=01101\ 0111001$ 此为规格化浮点数。

对阶后求差，对补码可用加法实现 $[X-Y]$

$$[-Y]_{\text{补}} = 1011101$$

差的尾数 = 1110011，可见尾数不是规格化浮点数，需左规两次： $= 1001100$

此时阶码应减 2，为 01011。

$$X-Y = 01011\ 1001100$$

第 4 章

1. 双地址指令最多为 2^4 条。因此，m 必须小于 2^4 。

单地址指令最多为 $(2^4 - m) \times 2^6 - \lceil L/64 \rceil$ 取整

4. $(2^4 - m) \times 2^6$

4.6.

I	X	E	寻址方式
0	00	$E=D$	直接寻址
0	01	$E=(PC)+D$	程序计数器相对寻址
0	10	$E=(R)+D$	寄存器相对(变址)寻址
0	11	$E=(R1)+D$	基址相对(变址)寻址
1	00	$E=(D)$	间接寻址
1	11	$E=((R1)+D)$	基址相对间接寻址

4.10.(1)AX=1200H

(2)AX=0100H

(3)AX=4C2AH

(4)AX=3412H

(5)AX=4C2AH

(6)AX=7856H

(7)AX=65B7H

4.12. 参考程序如下：

```

START : MOV  DI, 4000H
        MOV  DS, DI
        MOV  SI, 0000H
        MOV  CX, 0C000H

NEXT0: MOV  AL, 55H
        MOV  [SI], AL
        INC  SI
        LOOP NEXT0
        MOV  SI, 0000H
        MOV  CX, 0C000H

NEXT3: MOV  AL, [SI]
        CMP  AL, 55H
        JNE  NEXT5
        INC  SI
        DEC  CX
        JNZ  NEXT3
        MOV  AL, 7EH
        HLT

NEXT5: MOV  AL, 81H
        HLT

```

```

START : MOV  DI, 4000H
        MOV  ES, DI
        MOV  DI, 0000H
        MOV  CX, 0C000H

NEXT0: MOV  AL, 55H
        CLD
        REP  SCASB
        JCXZ NEXT7
        MOV  AL, 7EH
        HLT

NEXT7 : MOV  AL, 81H

```

HLT

15. 参考程序如下：

```
START: MOV DI, 0A000H
        MOV DS, DI
        MOV SI, 0100H
        MOV CX, 32
        MOV DX, 02E0H
GOON:   IN AL, DX
        AND AL, 24H
        CMP AL, 24H
        JNE GOON
        MOV DX, 02E7H
        IN AL, DX
        MOV [SI], AL
        INC SI
        LOOP GOON
        HLT
```

18. 参考程序如下：

```
START: MOV DI, 4000H
        MOV DS, DI
        MOV SI, 0000H
        MOV CX, 4000H
        MOV DX, 0000H
GOON:   MOV AL, [SI]
        CMP AL, 'A'
        JNE NEXT
        INC DX
NEXT:   INC SI
        DEC CX
        JNZ GOON
```

HLT

4.21.

人们将具有复杂指令系统的计算机称为复杂指令集计算机（CISC）。

复杂指令集计算机的主要特点是：

- (1) 指令系统复杂，指令数目多达 200 ~ 3000 条。
- (2) 指令长度不固定，有更多的指令格式和更多的寻址方式。
- (3) CPU 内部的通用寄存器比较少。
- (4) 有更多的可以访问主存的指令。
- (5) 指令种类繁多，但各种指令的使用频度差别很大。
- (6) 不同的指令执行时间相差很大，一般都需要多个时钟周期才能完成。
- (7) 控制器大多采用微程序控制器来实现。
- (8) 难以用优化编译的方法获得高效率的目的代码。

精简指令集计算机（RISC）精简指令集计算机的主要特点罗列如下：

- (1) 只设置使用频度高的一些简单指令，复杂指令的功能由多条简单指令的组合来实现。
- (2) 指令长度固定，指令种类少，寻址方式种类少。
- (3) 访存指令很少，有的 RISC 只有 LDA（读内存）和 STA（写内存）两条指令。多数指令的操作在速度快的内部通用寄存器间进行。
- (4) CPU 中设置大量的通用寄存器，一般有几个甚至几百个。
- (5) 控制器用硬件实现，采用组合逻辑控制器。
- (6) 采用流水线技术，大多数指令 1 个时钟周期即可完成。
- (7) 有利用优化编译程序。
- (8) 可简化硬件设计，降低设计成本。

确实是要将 CPU 的指令系统精减，只采用最经常使用的指令系统中 20% 的指令。同时，通过指令的减化可以使 CPU 的结构更加简单、更加合理，从而提高执行速度。

第 5 章

5.8.

一．水平型微指令

(1) 直接表示法

在微指令的控制域字段中，直接表示法就是直接用一个二进制位表示一种微命令。

(2) 字段译码法 (字段编码)

一种性能良好而普遍使用的控制域编码方法是字段译码法, 它是不译码法和译码法的结合。它将控制域分为若干字段。若各字段的编码相互独立, 则通过各字段独立译码就可以获得计算机系统的全部控制信号, 这被称作直接译码方式。若某些字段的编码相互关联, 则关联字段要通过两级译码才能获得相关的控制信号, 这被称作间接译码方式。

二. 垂直型微指令

垂直型微指令的控制域变得非常紧凑、短小, 是减小微指令长度的有效设计方法。垂直型微指令的控制字段一种极端情况是: 如果 CPU 执行机器指令只需 50 余种微命令, 则可用 6 位二进制编码构成微指令的控制字段。

垂直型微指令的控制域变得非常紧凑、短小, 是减小微指令长度的有效设计方法。垂直型微指令的控制字段一种极端情况是: 如果 CPU 执行机器指令只需 50 余种微命令, 则可用 6 位二进制编码构成微指令的控制字段。

5.13.

四个字段加上空操作分别需要 3、4、4、2 位表示而条件地址需 2 位, 共 15 位。因而, 次地址最多为 $2^4 - 15 = 9$ 位。叫控制存储器的容量为 $2^9 = 512$ 个单元, 每个单元 24 位

5.14. (1) 实现加法指令 ADD AX, (BX) 的微流程:

- ① PC_{out}, AR_{in}
- ② $AR \rightarrow AB, RD, PC + 1$
- ③ $MD \rightarrow DB, DR_{in}$
- ④ DR_{out}, IR_{in}
- ⑤ BX_{out}, AR_{in}
- ⑥ $AR \rightarrow AB, RD$
- ⑦ $MD \rightarrow DB, DR_{in}$
- ⑧ DR_{out}, S_{in}
- ⑨ $S \rightarrow ALU$
- ⑩ $AX_{out} \rightarrow ALU$
- ⑪ $ADD \rightarrow T, T_{in}$
- ⑫ T_{out}, AX_{in}

(2) 加法指令 ADD AX, DISP(SI) 的微流程:

- ① PC_{out}, AR_{in}

② $AR \rightarrow AB, RD, PC + 1$

③ $MD \rightarrow DB, DR_{in}$

④ DR_{out}, IR_{in}

⑤ PC_{out}, AR_{in}

⑥ $AR \rightarrow AB, RD, PC + 1$

⑦ $MD \rightarrow DB, DR_{in}$

⑧ DR_{out}, S_{in}

⑨ $SI_{out} \rightarrow ALU$

⑩ $ADD \rightarrow T, T_{in}$

⑪ T_{out}, AR_{in}

12 $AR \rightarrow AB, RD,$

13 $MD \rightarrow DB, DR_{in}$

14 DR_{out}, S_{in}

15 $S \rightarrow ALU$

16 $AX_{out} \rightarrow ALU$

⑬ $ADD \rightarrow T, T_{in}$

⑭ T_{out}, AX_{in}

(3) 减法指令 SUB BX,100(DI)的指令微流程：

① PC_{out}, AR_{in}

② $AR \rightarrow AB, RD, PC + 1$

③ $MD \rightarrow DB, DR_{in}$

④ DR_{out}, IR_{in}

⑤ PC_{out}, AR_{in}

⑥ $AR \rightarrow AB, RD, PC + 1$

⑦ $MD \rightarrow DB, DR_{in}$

⑧ DR_{out}, S_{in}

⑨ $DI_{out} \rightarrow ALU$

⑩ $ADD \rightarrow T, T_{in}$

⑪ T_{out}, AR_{in}

12 $AR \rightarrow AB, RD,$

13 $MD \rightarrow DB, DR_{in}$

14 DR_{out}, S_{in}

15 $S \rightarrow ALU$

16 $BX_{out} \rightarrow ALU$

17 $SUB \rightarrow T, T_{in}$

18 T_{out}, BX_{in}

5.17.

(1) A 为数据寄存器 ; B 为指令寄存器 ; C 为地址寄存器 ; D 为程序计数器

(2) LDA X 指令执行阶段的微操作流程 :

$X \rightarrow C$

$C \rightarrow MM, RD$

$MM \rightarrow A$

$A \rightarrow ALU \rightarrow AC$

(3) STA Y 指令执行阶段的微操作流程

$Y \rightarrow C$

$AC \rightarrow A$

$A \rightarrow MM, WR$