

第 7-10 章作业参考答案 15 级 1-2 班百家版

7.7

贡献者： 陈鹏辉 陈典



山东大学

201500130066 陈鹏辉

201500130019 陈典

订组 7.7 答案整理

解：二地址指令格式：

14 6 6

13 条 $\begin{cases} 0000 & XXXXXX & XXXXXX \\ op & \text{地址} & \text{地址} \\ 1100 & XXXXXX & XXXXXX \end{cases}$ 1101, 1110, 1111 用作扩展

一地址指令格式：

10 6

2^6 条 $\begin{cases} 1101 & 000000 & XXXXXX \\ & \vdots & \\ 1101 & 111111 & XXXXXX \\ op & \text{地址} \end{cases}$

2^6 条 $\begin{cases} 1110 & 000000 & XXXXXX \\ & \vdots & \\ 1110 & 111111 & XXXXXX \end{cases}$

2^6 条 $\begin{cases} 1111 & 000000 & XXXXXX \\ & \vdots & \\ 1111 & 111111 & XXXXXX \end{cases}$

一地址指令可以有 $3 \times 2^6 = 192$ 条。

7. 8

贡献者: 李振宇 吴小明

(1) 二地址指令^码操作数^码位数为 $16-6-6=4$, 有 $2^4=16$ 种操作
 \therefore 二地址指令最多有 $16-M-N$ 种

(2) 二地址指令每减少一种, 可构成 2^6 种一地址指令, 一地址指令每减少一种, 可构成 2^6 种零地址指令操作码.

\therefore 二地址指令操作码位数为 4 位, 设有 X 种二地址指令
 \therefore 至多有 $(2^4-X) \times 2^6$ 种一地址指令
 \therefore 至多有 $[(2^4-X) \times 2^6 - N] \times 2^6$ 种零地址指令
 $\therefore M = [(2^4-X) \times 2^6 - N] \times 2^6$

$$X = 2^4 - M \times 2^{-12} - N \times 2^{-6}$$

7. 16

贡献者: 史曾源 徐昊洋

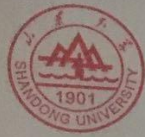
7. 16. (1)

OP (7位)	M (3位)	A (6位)
---------	--------	--------

OP 是操作码, M 是寻址方式, A 是地址码

(2) 直接寻址的最大范围是 $2^6=64$

贡献者：周聪 王银桥



山东大学

7.16

13) 存储字长为16位, 故一次间接寻址的范围是 2^{16} ,
两次间接寻址需要1位作为符号位, 故寻址范围为 2^{15}

14) 若为无符号数, 则范围为 $-32 \sim 31$

若为有符号数, 则范围为 $0 \sim 63$

贡献者：郝振云 冯惠妍



山东大学

7.16(5) 采用补码表示时, 相对寻址的位移量为 $-32 \sim 31$

16) 立即寻址执行时间最短, 因为此时不需寻址.

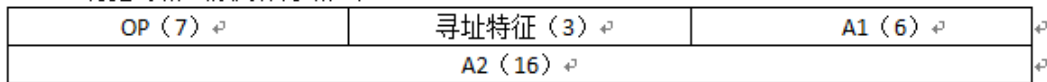
间接寻址执行时间最长, 因为此时需进行一次到多次访存.

相对寻址便于程序浮动, 因为此时操作数位置可随程序存储区的变动而改变, 总是相对于程序一段距离.

变址寻址最适合处理数组问题, 此时变址值可自动修改而不需修改程序

贡献者： 马付敏 黄韵萍

(7) 将指令格式改为双字格式



(8) 可采用上述双字长一地址指令，通过合适的存执方式完成。

8.2 什么是指令周期？指令周期是否有一个固定值？为什么？

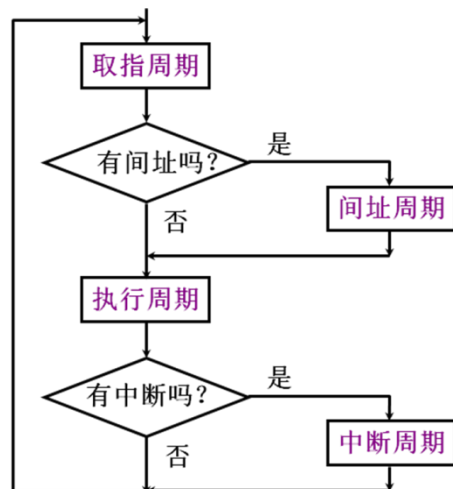
1) 指令周期是指 CPU 每取出并执行一条指令所需的全部时间。

2) 由于计算机中各种指令执行所需的时间差异很大，因此为了提高 CPU 运行效率，即使在同步控制的机器中，不同指令的指令周期长度都是不一致的，也就是说指令周期对于不同的指令来说不是一个固定值。

3) 指令周期长度不一致的根本原因在于设计人员，为了提高 CPU 运行效率而这样安排的，指令功能不同，需完成的微操作复杂程度亦不同，因此，不同指令的指令周期也不同。

8.3 画出指令周期的流程图，分别说明图中每个子周期的作用。

流程图如下：



取指周期：完成取指令和分析指令的操作。

间址周期：取操作数的有效地址。

执行周期：执行指令的操作。

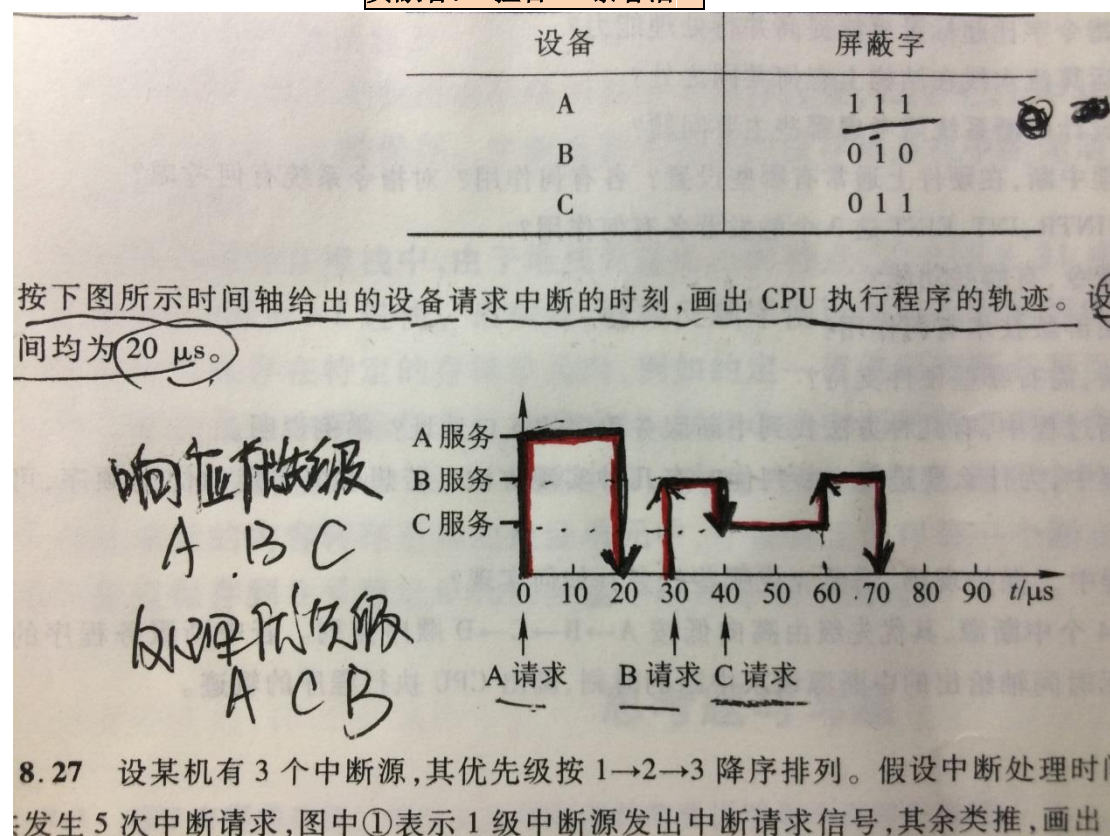
中断周期：将程序断点保存到存储器。

8.5 中断周期前是什么阶段？中断周期后又是什么阶段？在中断周期 CPU 应完成什么操作？

中断周期前是指令的执行阶段（处于执行周期）。中断周期后是取指令阶段（处于取指周期）。在中断周期中，CPU 应完成关中断、保存断点和转中断服务程序入口三个操作。

8. 26

贡献者：拉吉 黎睿洁



9. 7

贡献者： 叶欢 黄嘉华

时钟周期 = $0.1 \times 10^{-6} \text{ s}$
 机器周期 = $0.4 \times 10^{-6} \text{ s}$

① 平均指令周期 = 10^{-6} s
 每个指令周期包含 $\frac{10^{-6}}{0.4 \times 10^{-6}} = 2.5$ 个机器周期

② 改变芯片后, 机器周期 = 1.6 us
 则改变后平均指令周期 = $1.6 \text{ us} \times 2.5 = 4 \text{ us}$
 平均指令执行速度为 $\frac{1}{4 \text{ us}} = 0.25 \text{ MIPS}$

③ 若要得到 0.8 MIPS 的指令执行速度,
 平均指令周期为 $\frac{1}{0.8 \text{ MIPS}} = 1.25 \text{ us}$
 机器周期 = $\frac{1.25}{2.5} = 0.5 \text{ us}$
 时钟周期 = $\frac{0.5}{4} = 0.125 \text{ us}$
 主频 = $\frac{1}{0.125 \text{ us}} = 8 \text{ MHz}$

9. 13 (1)

贡献者： 李昕宜 侯林林

取指周期

(PC) → Bus → MAR	PC ₀ , MAR _i
1 → R	M(R) = 1 (对内存的读命令有效)
M (MAR) → MDR	
(MDR) → Bus → IR	MDR ₀ , IR _i
OP(IR) → CU	
(PC) + 1 → PC	PC ₊₁

执行周期

(R ₄) → Bus → MAR → 地址线	R ₄₀ , MAR _i
1 → R	M(R) = 1 (对内存的读命令有效)
M (MAR) → MDR	

MDR \rightarrow Bus \rightarrow Y

MDR₀, Y_i

(R₂) \rightarrow Bus \rightarrow ALU

R2₀, ALU_i

(R₂) + (Y) \rightarrow Z

"+"

(Z) \rightarrow Bus \rightarrow R₂

Z₀, R2_i

9. 13 (2)

贡献者: 马俊博 李星宇

取指令 PC \rightarrow BUS \rightarrow MAR PC₀, MAR_i
取地址 out of range 溢出

② M(MAR) \rightarrow MDR MAR₀, 1 \rightarrow R, MDR_i

③ MDR \rightarrow BUS \rightarrow IR MDR₀, IR_i
送指令寄存器 +1 取下一条指令

④ PC + 1 \rightarrow PC PC₀, MAR_i

⑤ Ad(IR) \rightarrow BUS \rightarrow MAR IR₀, MAR_i

⑥ M(MAR) \rightarrow MDR MAR₀, 1 \rightarrow R, MDR_i

⑦ MDR \rightarrow Ad(IR) MDR₀, IR_i 找到数据所在地址

⑧ Ad(IR) \rightarrow MAR IR₀, MAR_i

⑨ M(MAR) \rightarrow MDR MAR₀, 1 \rightarrow R, MDR_i

⑩ MDR \rightarrow BUS \rightarrow Y MDR₀, Y_i 取数据 mem

⑪ R₁ \rightarrow BUS \rightarrow ALU R₁₀, ALU_i

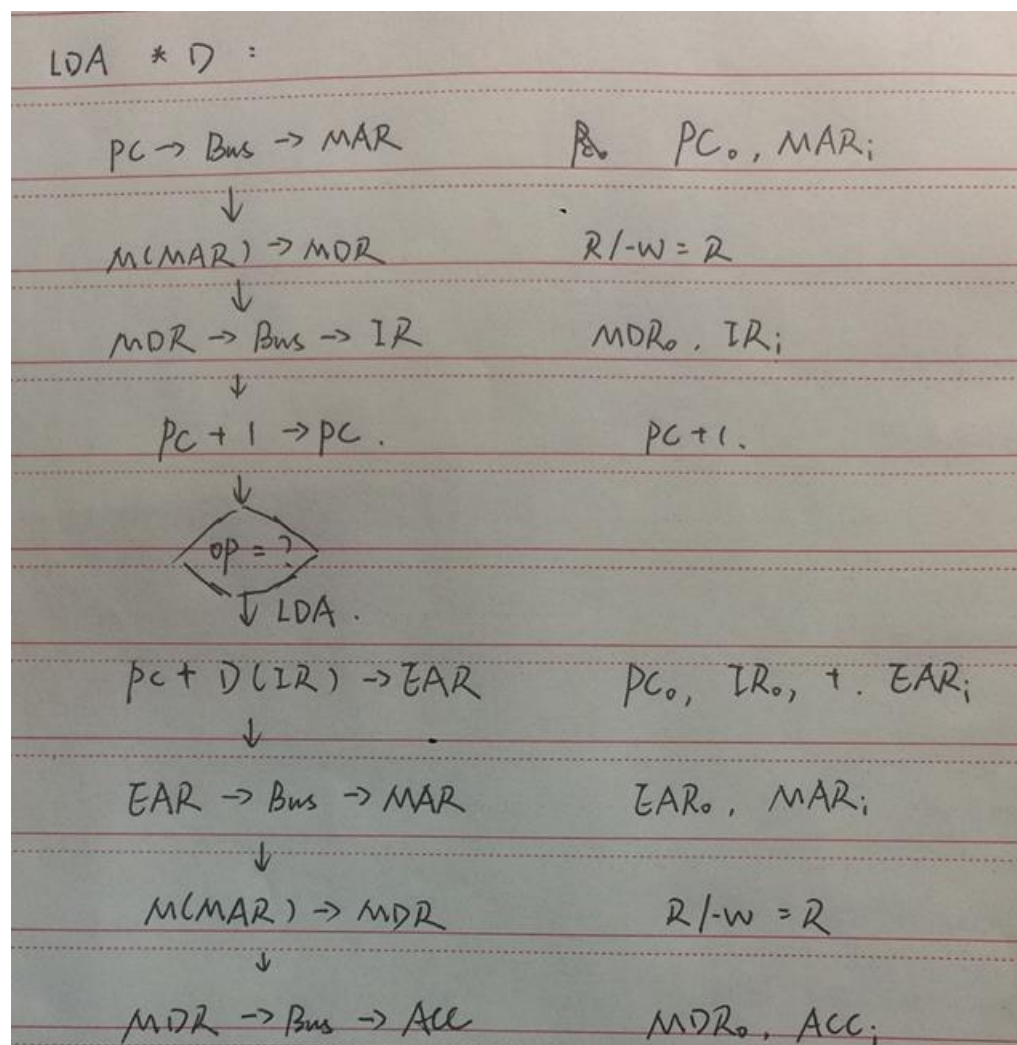
⑫ BUS - Y \rightarrow Z Y₀, Z_i

⑬ Z \rightarrow BUS \rightarrow R₁ Z₀, R_{1i}

ALU 运算
减法, 2 为结果
Z 的结果送到 R₁
更新寄存器

9. 14 (1)

贡献者: 谢德龙 孔连杰



张天姿
张珊
9.14(2)

山东大学
1901
SHANGHAI UNIVERSITY

张珊/张天姿

取指周期: $PC \rightarrow Bus \rightarrow MAR$	PC_0 MAR_i
$M(MAR) \rightarrow MDR$	MAR_0 $R/\bar{W}=R$, MDR_i
$MDR \rightarrow Bus \rightarrow IR$	MDR_0 IR_i
$PC+1 \rightarrow PC$	$+1$
执行周期: $(XR) + Ad(IR) \rightarrow EAR$	XR_0 $Ad(IR)_0$, $+ EAR_i$
$EAR \rightarrow Bus \rightarrow MAR$	EAR_0 MAR_i
$M(MAR) \rightarrow MDR$	MAR_0 , $R/\bar{W}=R$, MDR_i
$MDR \rightarrow Bus \rightarrow X$	MDR_0 Acc X_i
$(ACC) - X \rightarrow LATCH$	ACC_0 , X_0 , $K_i = "-"$, $LATCH_i$
$LATCH \rightarrow Bus \rightarrow ACC$	$LATCH_0$ ACC_i

10. 2 (1)

贡献者: 牛顿 王昊

(1) 取指周期: $T_0: PC \rightarrow MAR, 1 \rightarrow R$

$T_1: M(MAR) \rightarrow MDR, (PC) + 1 \rightarrow PC$

$T_2: MDR \rightarrow IR, OP(IR) \rightarrow ID$

执行周期: $T_0: Ad(IR) \rightarrow MAR, 1 \rightarrow R$

$T_1: M(MAR) \rightarrow MDR$

$T_2: MDR \rightarrow Y$

$T_3: (R_1) * Y \rightarrow Z$

$T_4: Z \rightarrow R_1$

其中 ALU 输入端 分别连接 总线 和 寄存器 Y 的 输出端
ALU 的输出端 与 寄存器 Z 的 输入端 相连, Y 的 输入端 与
总线 相连, Z 的 输出端 与 总线 相连.

10. 2 (2)

贡献者: 阮荣坤 周志鹏

(2)↵

取指周期: $T_0: PC \rightarrow MAR, 1 \rightarrow R$ ↵

$T_1: M(MAR) \rightarrow MDR, PC + 1 \rightarrow PC$ ↵

$T_2: MDR \rightarrow IR, OP(IR) \rightarrow ID$ ↵

执行周期: $T_0: Ad(IR) \rightarrow MAR, 1 \rightarrow R$ ↵

$T_1: M(MAR) \rightarrow MDR$ ↵

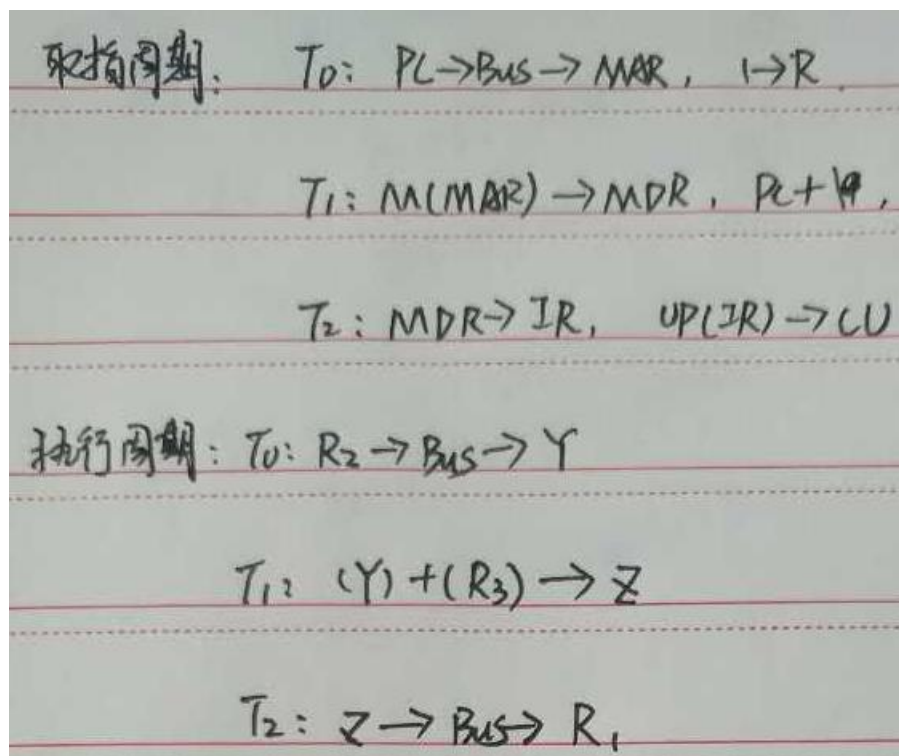
$T_2: (MDR) + 1 \rightarrow ACC \rightarrow R_1$ ↵

$T_3: R_1 \rightarrow MDR, 1 \rightarrow W$ ↵

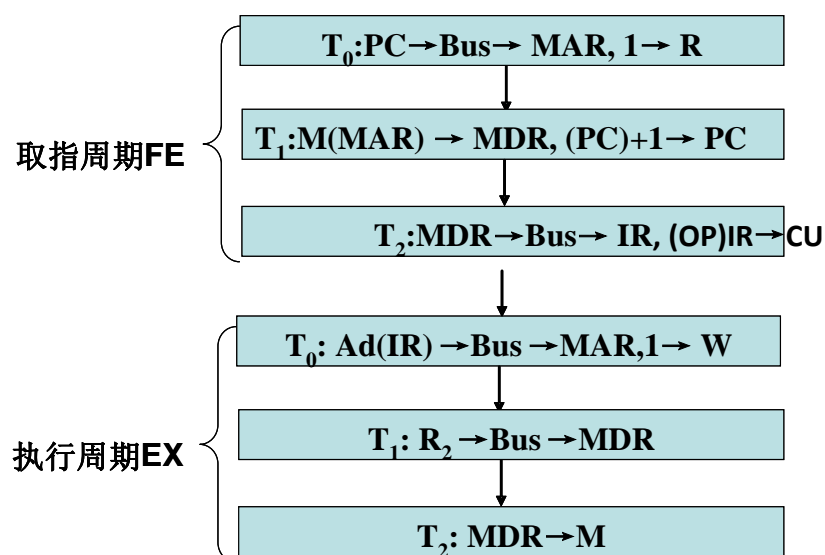
$T_4: MDR \rightarrow M(MAR), (PC + 1) * R_1 + PC * \overline{R_1} \rightarrow PC$ ↵

10. 4 (1)

贡献者： 田智君 罗布顿珠



10. 4 (4)



10.9 试比较组合逻辑设计和微程序设计的设计步骤和硬件组成，说明哪一种控制速度更快，为什么？

一) 设计步骤

组合逻辑控制器的设计步骤：1) 拟定机器的指令系统；2) 确定 CPU 总体结构；3) 确定时序系统，拟定指令流程；4) 安排每条指令中微操作的节拍；5) 列出微操作命令的操作时间表；6) 写出每一个微操作命令的逻辑表达式并化简；7) 画出相应的组合逻辑电路图。

微程序控制器的设计步骤：前三个步骤和组合逻辑控制器相同，后边的步骤如下：

- 1) 写出对应机器指令的微操作及节拍安排；
- 2) 确定微指令格式（确定微指令的编码方式和后继微地址的形成方式）；
- 3) 编写微指令码点。

二) **硬件组成**：组合逻辑控制器由组合逻辑电路提供微命令，其核心器件是各种门电路构成的复杂树形网络；微程序控制器由存储逻辑(微指令)提供微命令，其核心器件是控制存储器。

三) 组合逻辑控制器速度更快，因为其微命令全部由硬件（组合逻辑门电路）产生。