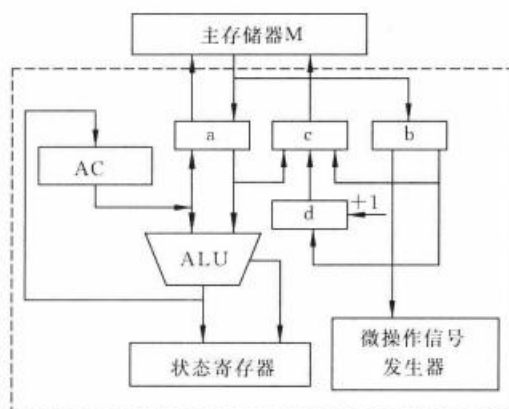


第六章 中央处理器

6.1 CPU 结构如下图所示，其中有一个累加寄存器 AC、一个状态条件寄存器和其他 4 个寄存器，各部分之间的连线表示数据通路，箭头表示信息传送方向。
要求：

- (1) 标明图中 a, b, c, d 这 4 个寄存器的名称。
- (2) 简述指令从主存取出到产生控制信号的数据通路。
- (3) 简述数据在运算器和主存之间进行存/取访问的数据通路。



答：(1) a 是数据寄存器 MDR，c 是地址寄存器 MAR，b 是指令寄存器 IR，d 是程序计数器 PC。

因为主存 M 和 CPU 之间的连线有地址线 and 数据线，其中地址线的信息仅从 CPU 传送到主存，数据则允许双方相互传送，由此可见 a 是 MDR，c 是 MAR。

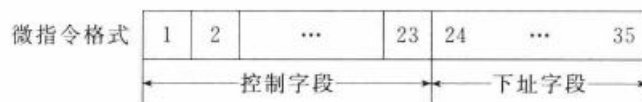
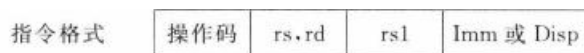
PC 送地址到地址寄存器，并有自动加一的功能，因此 d 是 PC。

IR 接受从主存来的命令，并控制微型操作信号发生器，因此 b 是 IR。

(2) 指令地址从 d (程序计数器 PC) 送到 c (地址寄存器 MAR)，数据从主存中送到 b (指令寄存器 IR)。

(3) 读写地址由 b (IR 的地址段) → c (地址寄存器 MAR)，读时数据由主存储器 M → a (数据寄存器 MDR) → ALU → AC，写时数据由 AC → a (数据寄存器 MDR) → 主存储器 M。

6.2 设某计算机运算控制器逻辑图如图 6.6 所示，控制信号意义如表 6.1 所示，指令格式和微指令格式如下：



其中 1~23 位代表的 1~23 号控制信号如表 6.1 所示。

试着写出下述 3 条指令的微程序编码：

- (1) JMP (无条件转移到 (rs1) +Disp)
- (2) Load (从 (rs1) +Disp 指示的内存单元取数，送 rs 保存)

(3) Store (把 rs 内容送到 (rs1) +Disp 指示的内存单元)

提示：先列出各指令的执行步骤和每步所需要的控制信号，最后再写出编码。

答：每条指令都要执行取指令到指令寄存器的微指令，这是一条公用的微指令。

所需的控制信号如下：

PC → AB (由上一条指令结束前发出)

指令 → IR₀:ADS, M/IO, 读 (W/R=0) , 主存自动将读出数据送数据总线 DB,

DB → IR。以上信号控制完成取指令操作。

PC + 1: 为顺序执行的下一条指令地址做准备。

所以第一条取指微指令的编码如下：

<1>0010, 1000, 0000, 0000, 0000, 110

然后由指令译码器 (ROM) 根据操作码产生下一条微指令的地址。

(1) JMP 指令所需控制信号如下。

计算转移地址 ((rs1) +disp) 送入 AR:

rs1 → GR, (rs1) → ALU, disp → ALU, +, ALU → AR

下一条指令地址送入 PC:

ALU → PC

所以 JMP 的微指令为

<2>0101, 0001, 0100, 1000, 0010, 0XX

(2) LOAD 指令。

计算主存地址 ((rs1) + disp) 并送入 AR:

rs1 → GR, (rs1) → ALU, disp → ALU, +, ALU → AR

所以第 2 条微指令为

<2>0101, 0001, 0100, 1000, 0010, 0XX

从主存取数: AR → AB (地址送地址总线), ADS, M/IO 读 (W/R = 0), DB → DR

(读出数据送数据寄存器)

所以第三条微指令为

<3>0000, 0100, 0000, 0000, 0001, 110

存储器取出的数据经过 ALU 送入 r_d

DR → ALU, r_d → GR (地址), +, ALU → GR (数据)

下一条指令地址送入地址总线:

PC → AB

所以第 4 条微指令为

<4>1000, 0000, 1001, 1000, 1000, 0XX

(3) STORE 指令

计算主存地址 ((rs1) + disp) 并送入 AR:

rs1 → GR, (rs1) → ALU, disp → ALU, +, ALU → AR

所以第 2 条微指令为

<2>0001, 0001, 0100, 1000, 0010, 0XX

从 GR 取数到 DR:

rs → GR (地址), (rs) → ALU, +, (ALU 另一个输入端为 0), ALU → DR

(读出数据送数据寄存器)

所以第三条微指令为

<3>0000, 0000, 1010, 1000, 0100, 0XX

将数据存入主存:

AR \rightarrow AB, DR \rightarrow DB, ADS, M/IO, W/R

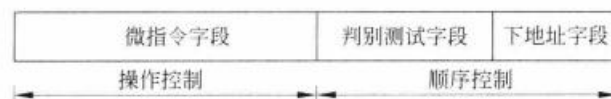
PC \rightarrow AB

所以第 4 条微指令为

$\langle 4 \rangle 1000, 0010, 0000, 0000, 0001, 111$

(X 可以表示任意值, 当 ADS = 0 时, 微指令最后两位不起作用。)

6.6 已知某机采用微程序控制方式, 其控制存储器容量为 521X48 位。微指令字长为 48 位, 微程序可在整个控制存储器中实现转移, 可控制微程序转移的条件共 4 个 (直接控制), 微指令采用水平型格式, 如下图所示。请问微指令中的 3 个字段分别应为多少位?

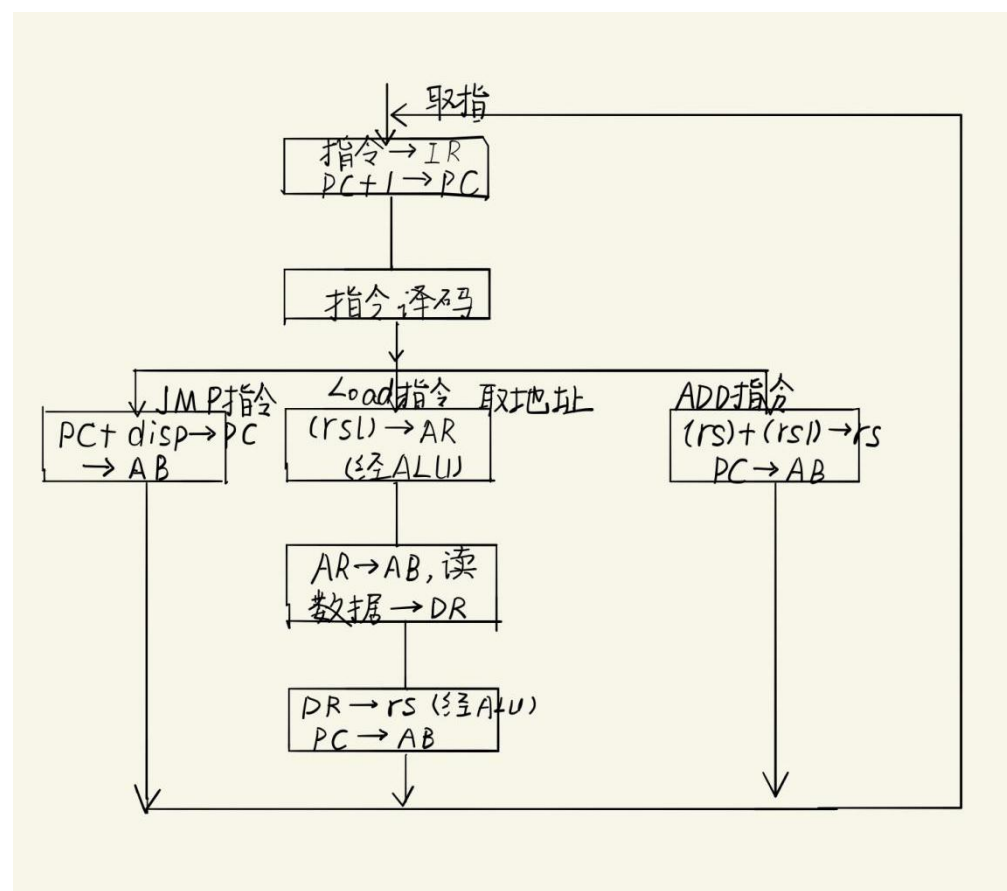


答: 控制存储器容量为 512 字, 所以下地址字段为 9 位; 判别测试字段控制微程序转移, 为 4 位, 还有 $48-9-4=35$ 位为微指令字段长度。

6.7 参照教材中的图 6.6 和表 6.1 画出下述 3 条指令的微程序流程图。

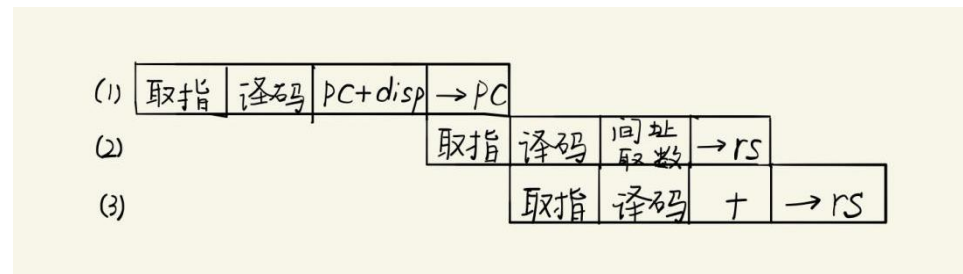
- (1) JMP disp (相对寻址)
- (2) Load rs, @rs1 (间接寻址)
- (3) ADD rs, rs1 (寄存器寻址)

答:



6.8 假设某计算机采用 4 级流水线（取指、译码、执行、送结果），其中译码级可同时完成从寄存器取数的操作，并假设存储器的读写操作（允许同时取指和取数）可在一个机器周期内完成，问顺序执行上题的 3 条指令总共需要多少周期？

答：3 条指令的流水线如图：



假设 PC + disp 的结果直接作为下一条指令的地址去取指，总共需要 8 个周期完成 3 条指令。假如下一条指令地址要从 PC 中取，则还需要加上 1 个周期，即取第 (2) 条指令要在第 (1) 条指令的“-> PC”操作完成后才能进行。

6.13 微机 A 和微机 B 是采用不同主频的 CPU 芯片，片内逻辑电路完全相同。若 A 机的 CPU 主频为 8MHz，B 机为 12MHz，则 A 机的 CPU 主振周期为 A us, 如 A 即的平均指令执行速度为 0.4MIPS，则 A 机的平均指令周期为 B us, B 机的平均指令执行速度为 C MIPS。

答：

A 为 0.125us。因为 A 机主振周期 = $1/8 \text{ MHz} = 0.125\text{us}$ 。

B 为 2.5us。因为 A 机的平均指令周期 = $1/0.4\text{MIPS} = 2.5\text{us}$ 。

C 为 0.6MIPS。因为 B 机的平均指令执行速度 = $0.4\text{MIPS} \times 12 / 8 = 0.6\text{MIPS}$ 。

6.14 某机采用两级流水线组织，第一级为取指、译码，需要 200ns 完成操作‘第二级为执行周期，大部分指令能在 180ns 内完成，但有两类指令要 360ns 才能完成，在程序运行时，这类指令所占比例为 5%~10%。

根据以上情况，机器周期（即一级流水线操作时间应选为 A）。两条执行周期长的指令采用 B 的方法解决。

A: 1. 180ns, 2. 190ns, 3. 200ns, 4. 360ns

B: 1. 机器周期选为 360ns; 2. 两个机器周期完成

答：A---3. B---2.

因为在流水线的各个机器周期中，相邻的指令处于不同的流水线，所以机器周期应该选择两级中的较大的时间。虽然在指令集中有两条指令要 360ns 要完成，但所占比例小，因此可以采用特殊方法处理，所以机器周期选为 200ns 比较合适。两条执行周期长的指令，采用延长机器周期的方法，较为简单的方法是采用两个机器周期完成，这显然要比将整个机器周期选为 360ns 要快的多。