

一条指令的硬件实现

大纲

- ch5 中央处理器 (CPU)
 - (一) CPU的功能和基本结构
 - CPU = 控制器 + 数据通路
 - 控制器, 即CU, 是指令的控制部件
 - 数据通路, 是指令的执行部件, 包括运算器、寄存器、数据传输的线路等
 - (二) 指令执行过程
 - 取指、译码
 - 取指令送入IR、PC+1
 - CU根据IR译码, 确定指令的执行时序
 - 执行
 - 取操作数、运算、存结果
 - 异常/中断的检测与处理 (如果有的话)
 - (三) 数据通路的功能和基本结构
 - (四) 控制器的功能和工作原理
 - 1. 硬布线控制器
 - 2. 微程序控制器
 - (五) 异常和中断机制
 - 1. 异常和中断的基本概念
 - 2. 异常和中断的分类
 - 3. 异常和中断的检测与响应
 - (六) 指令流水线
 - 1. 指令流水线的概念
 - 2. 指令流水线的实现
 - 3. 结构冒险、数据冒险、控制冒险的处理
 - 4. 超标量和动态流水线的概念
 - (七) 多处理器基本概念
 - 1. SISD、SIMD、MIMD、向量处理器的基本概念
 - 2. 硬件多线程的基本概念
 - 3. 多核处理器 (multi-core) 的基本概念
 - 4. 共享内存多处理器 (SMP) 的基本概念

2022大纲更新后考察权重被弱化

思考: 一条指令的执行过程

- 取指阶段 (所有指令在取指阶段做的事情都一样)
 - 1. 根据PC从主存中取指令到IR
 - 2. PC+1
 - 可以考虑用硬件 ①、② 实现 PC+1
 - 也可以考虑用硬件 ② 实现 PC自增
- 执行阶段 (根据指令类别来思考)
 - 数据传送类指令 (如: mov, load, store) — 关注数据从哪里流动到哪里?
 - 主存
 - 寄存器
 - 立即数
 - 运算类指令
 - 加、减
 - 可以考虑用硬件 ①②
 - 自增++, 自减- 可以考虑用硬件 ①②③
 - 乘
 - 可以考虑用硬件 ①③
 - 特殊乘法, 乘以 2^n , 可用 ① 左移功能代替, 也可用 ①ALU的左移功能代替
 - 除
 - 可以考虑用硬件 ①④
 - 特殊除法, 除以 2^n , 可用 ① 右移功能代替, 也可用 ①ALU的右移功能代替
 - 移位运算 — 可以考虑用硬件 ①⑤ 实现
 - 与、或、异或等双操作数逻辑运算 — 可考虑用硬件 ① 实现
 - 非运算 — 可考虑用硬件 ①、⑤ 实现
 - 短数—>长数
 - 带符号数扩展可使用硬件 ③
 - 无符号数扩展可使用硬件 ④
 - 转移类指令
 - 条件转移
 - 做条件判断时, 经常用到减法生成标志位, 因此可能会用到硬件 ① 或 ②
 - 减法生成的CF、ZF、SF、OF标志位需要用“标志寄存器”存储起来
 - 无条件转移

无论是条件转移, 还是无条件转移, 在计算转移目标地址时, 需要将“PC+偏移量”, 因此可以考虑使用硬件 ①或②实现这个加法运算

Key: 指令各阶段的功能, 要用什么硬件部件实现? 数据应该如何流动? ——硬件的控制信号、连线只是实现逻辑的一种外化体现。

分析框架

如何看“注释”

- 控制硬件的信号
 - XXXin — 允许数据进入
 - XXXout — 允许数据出去
 - XXXop — 具体的控制信号
- 操作元件 (组合逻辑元件)
 - ①算术逻辑单元 (ALU) — 实现加减乘除等算术运算、与或非等逻辑运算
 - 注意: ALU可以支持“直送”, 即不对输入数据做任何处理
 - 控制信号: n bit, 取决于ALU支持多少种运算, n bit控制信号, 对应 2^n 种运算
 - ②加法器 (Adder) — 结合第二章复习, 相当于弱鸡版ALU, 只能实现加法、减法
 - 控制信号: 1bit — 0加1减, 控制信号的位会影响 CF 标志位
 - ③乘法器 — 实现乘法运算
 - ④除法器 — 实现除法运算
 - ⑤三态门 — 就是一道门, 可开可关
 - 控制信号: 1bit, 用于控制开关
 - ⑥多路选择器 (MUX) — 有 2^n 路输入, 通过 n bit 控制信号选择让其中的某一路输入通过
 - 控制信号: n bit, 取决于有多少路输入, n bit控制信号, 对应 2^n 路
 - ⑦译码器 (Decoder) — 根据 n bit 控制信号选择将 2^n 条输出线路中的某一条信号选通为1
 - 控制信号: n bit, 由译码器的 n bit输入即可决定选通哪根线
- 存储元件 (时序逻辑元件)
 - ⑧普通寄存器 — 具有特定用途的寄存器: MAR、MDR、IR、PC、FR (即标志寄存器, 有的系统中称为PSW)
 - 控制信号: 允许读、允许写
 - ⑨暂存寄存器 — 通常会在单总线结构和ALU配合使用
 - 控制信号: 允许读、允许写
 - ⑩通用寄存器组
 - 控制信号1: 允许读、允许写
 - 控制信号2: 若通用寄存器组共有 2^n 个寄存器, 则需要 n bit 控制信号选择其中一个寄存器

- ⑪. 带“移位功能”的寄存器
 - 可以实现移位运算 (算数左移/右移、逻辑左移/右移)
 - 隐藏技能: 用移位运算等价实现“乘除 2^n ”
- ⑫. 带“自增、自减功能”的寄存器
 - 可实现寄存器的值++、--
 - 常见用途: 有的PC寄存器带有自增功能
- 带其他特殊功能的寄存器
 - ⑬. 带“符号扩展”功能的寄存器 — 可以将位数短的带符号数 (补码) 扩展为位数更长的带符号数 (补码)
 - ⑭. 带“零扩展”功能的寄存器 — 可以将位数短的无符号数扩展为位数更长的无符号数
 - ⑮. 带“取反功能”的寄存器 — 可以实现全部位按位取反

- 数据在通用寄存器组
 - 特点: 用R[i]表示, 中括号内为寄存器编号
 - eg: $R[6] \times R[3] \rightarrow R[2]$. 将寄存器R6 * R3的内容写入R2
- 数据在某个特殊名字的寄存器
 - 特点: 直接用寄存器名字表示
 - eg: $R[6] + 10 \rightarrow PC$. 把寄存器R6+10的内容写入PC
 - eg: $MDR \rightarrow PC$. 把MDR的内容写入PC
- 数据在主存
 - 特点: 用M[addr]表示, addr为主存地址
 - eg: $R[6] \rightarrow M[10086]$ — 将R6的内容存入主存地址10086
 - eg: $R[3] \ll 2 \rightarrow M[R[6]]$ — 将R3进行“左移两位”运算, 并将结果存入R6所指的主存地址 (寄存器间接寻址)
 - eg: $M[PC+10] \rightarrow MDR$ — 将PC+10指向的主存内容取到MDR
- 特点总结
 - 喜欢用“中括号”, 中括号里边表示寄存器编号、或主存地址
 - 要使用寄存器里的内容, 不用加小括号, 直接给寄存器名即可

- 数据在通用寄存器组
 - 特点: 箭头左边, 用寄存器名加小括号, 表示寄存器里的值; 箭头右边表示存入某个寄存器不加小括号
 - eg: $(R6) \times (R3) \rightarrow R2$. 将寄存器R6 * R3的内容写入R2
- 数据在某个特殊名字的寄存器
 - 特点: 直接用寄存器名字表示, 要使用寄存器里的值需要加小括号
 - eg: $(R6) + 10 \rightarrow PC$. 把寄存器R6+10的内容写入PC
 - eg: $(MDR) \rightarrow PC$. 把MDR的内容写入PC
- 数据在主存
 - 特点: 用M(addr)表示, addr为主存地址
 - eg: $(R6) \rightarrow M(10086)$ — 将R6的内容存入主存地址10086 (寄存器间接寻址)
 - eg: $(R3) \ll 2 \rightarrow M(R6)$ — 将R3进行“左移两位”运算, 并将结果存入R6所指的主存地址 (寄存器间接寻址)
 - eg: $((PC) + 10) \rightarrow MDR$ — 将PC+10指向的主存内容取到MDR
 - 在箭头的左边表示“访问主存”时, 可以不加M, 用两层括号表示寄存器间接寻址
- 特点总结
 - 喜欢用“小括号”
 - 箭头左边若要使用某个寄存器里的内容, 必须在寄存器名字外面加小括号
 - 箭头右边表示存入某个寄存器, 不用加小括号

取指阶段：把PC所指的指令取出来
取到IR

PC \rightarrow MAR

M(MAR) \rightarrow MDR \rightarrow IR



此“1”为下一条指令(注意指令字长)

PC + “1”

实现方式
(硬件)

\rightarrow ALU (加法)

\rightarrow 加法器

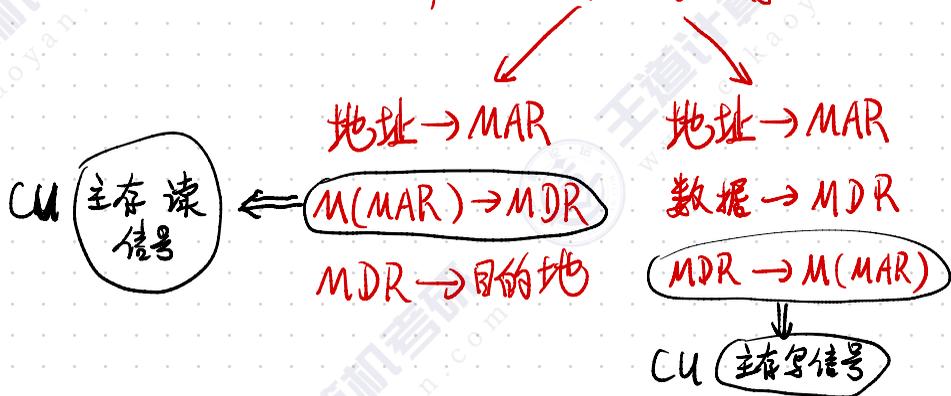
\rightarrow 有“自增功能”的寄存器 (PC寄存器)

执行阶段：数据传送类

- ① 寄 → 寄
- ② 寄 → 主存
- ③ 主存 → 寄
- ④ 立 → 寄
- ⑤ 立 → 主

⑥ 寄 → 暂存寄存器
↑
某条指令的子步骤

注意1：如果有主存，关注读/写主存



注意2：关注总线占用（总线是临界资源）
(安排控制信号)

执行阶段：转移类指令 → 改变PC值 (重点题, 13年43题, 4.2.3大8)
(可能)

条件转移 → Cmp 前置 (本质是减法 $A-B$, 生成 CF, ZF, OF, SF)

↓
回顾如何生成?

原理：根据 标志位 判断是否转移 (指令中会给 地址码)

如果要转移：①. 注意指令寻址方式

- 相对寻址 → $PC + \text{偏移}$
 - 以字为单位
 - 以字节为单位
- 直接寻址 (少见)

RISC (指令长度相同)

②. 注意PC的值

- 以字节为单位
- 以指令字为单位

CISC

王道考研——计算机组成原理

WWW.CSKAOYAN.COM

强化P4：一堆指令的执行

认准一手课程完整更新公众号【研途小时】！

2009	2010	2011	2012	2013	2014	2015	2016
中断控制方式的处理过程; DMA控制方式的处理过程。	指令格式; 寻址范围; 指令执行的微操作过程	C语言中常见变量的表示; 强制类型转换; 补码加法的应用; 溢出判断。	CPU性能指标的计算; 引入Cache后的访存原理; 虚拟内存的工作原理; DMA控制方式的工作原理; 低位交叉存储的流水线	CPU性能指标、总线性能指标的计算; 低位交叉存储方式; 总线的突发传送过程; 引入Cache后的访存原理;	结合C语言, 读懂各条指令的作用; 条件转移指令的工作原理; 五段式指令流水线	指令执行的数据通路; 数据通路上各种常见的硬件部件及作用; 控制信号连线;	中断控制方式的处理过程;
指令执行的详细过程; 说明每一步的微操作、微命令。并安排合理的时序	Cache和主存的映射; C语言汇总二维数组的存储原理; Cache命中率的计算	虚拟存储系统的地址结构; Cache的工作原理; TLB的工作原理;	数据的移位运算; 五段式指令流水线; 流水线的“冲突”原因;	条件转移指令的工作原理; CPU内部常见的硬件部件(根据处理逻辑推测)	Cache的工作原理; 指令的溢出判断; 虚拟存储, 缺页异常的产生原因; TLB的工作原理;	指令格式; 各步微操作对应的微命令; 微操作的时序安排;	TLB的工作原理; Cache的工作原理; 虚拟存储, 缺页异常; Cache淘汰策略、页面淘汰处理;
2017	2018	2019	2020	2021	2022	2023	2024
C语言强制类型转换; 各种数的精度问题、溢出问题	程序定时查询方式的工作过程; 中断查询方式的工作过程; DMA方式的工作过程;	C语言对应的指令序列; 条件转移指令、无条件转移指令、函数调用call指令的原理; 数据的精度、溢出问题	数据的运算: 二进制乘法; 溢出问题	指令格式; 数据的运算、溢出问题;	一条指令的执行过程; 指令执行的电路数据通路原理	Cache、虚拟页式存储	一条指令的执行过程; 指令执行的电路数据通路原理
C语言对应的指令序列; 比较指令cmp、条件转移指令的工作原理; 数据的的运算, 算数左移	虚拟存储, 地址结构; TLB的工作原理; Cache的工作原理; 有TLB、Cache的地址变换过程	虚拟分页存储; Cache的工作原理;	Cache的工作原理; 结合C语言分析Cache命中情况;	虚拟存储, 地址结构; TLB的工作原理;	磁盘+IO控制方式	结合C语言, 分析指令序列的工作原理	结合C语言, 分析指令序列的工作原理

• 图示说明:

主考第二章

数据的运算; 强制类型转换; 精度、溢出问题

3+n

主考第三章

Cache、TLB、虚拟分页

9

主考第五章

一条指令的执行过程

7

主考第四章

指令序列的工作过程

6

主考第七章

三种IO控制方式

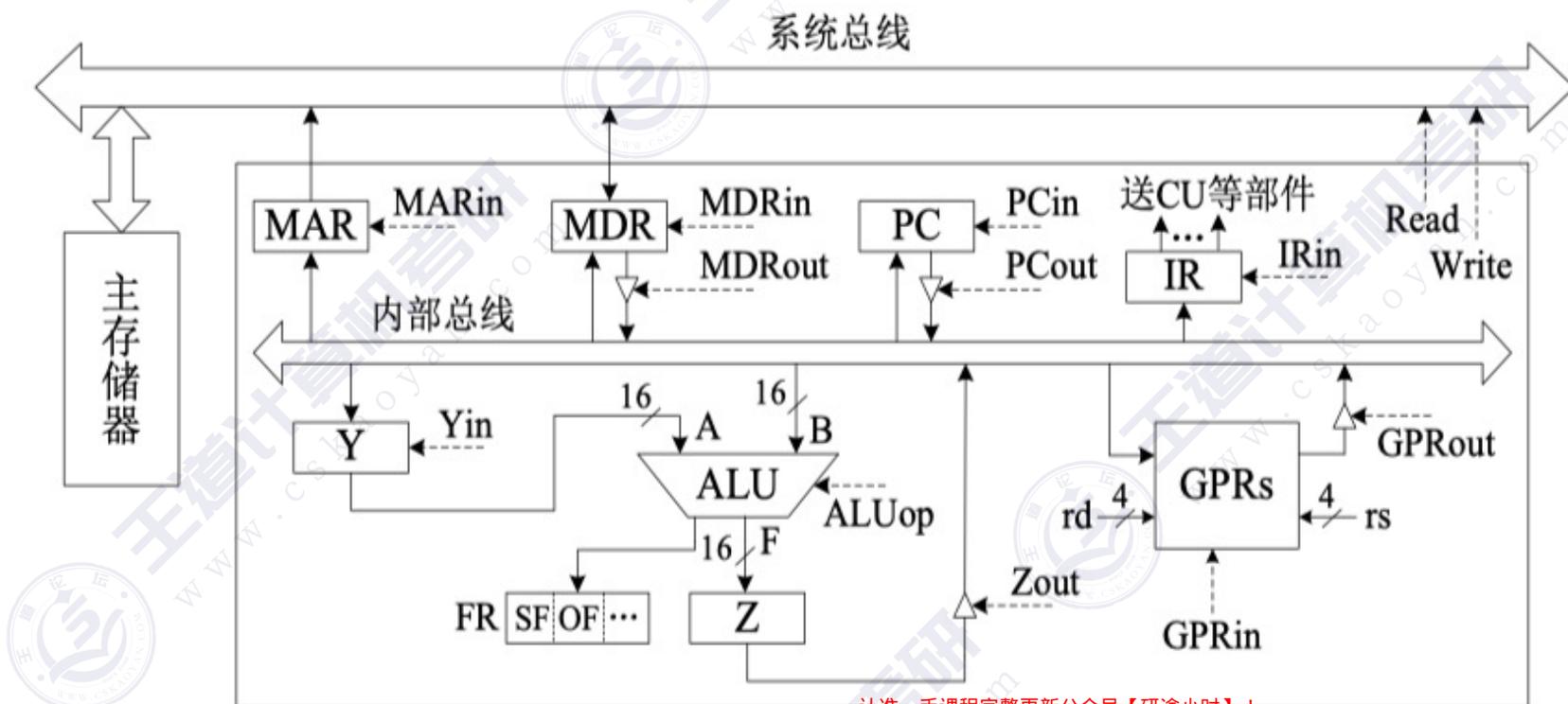
4

杂七杂八

3

2022年真题

43. (15分) 某CPU中部分数据通路如题43图所示, 其中, GPRs为通用寄存器组; FR为标志寄存器, 用于存放ALU产生的标志信息; 带箭头虚线表示控制信号, 如控制信号Read、Write分别表示主存读、主存写, MDRin表示内部总线上数据写入MDR, MDRout表示MDR的内容送内部总线。



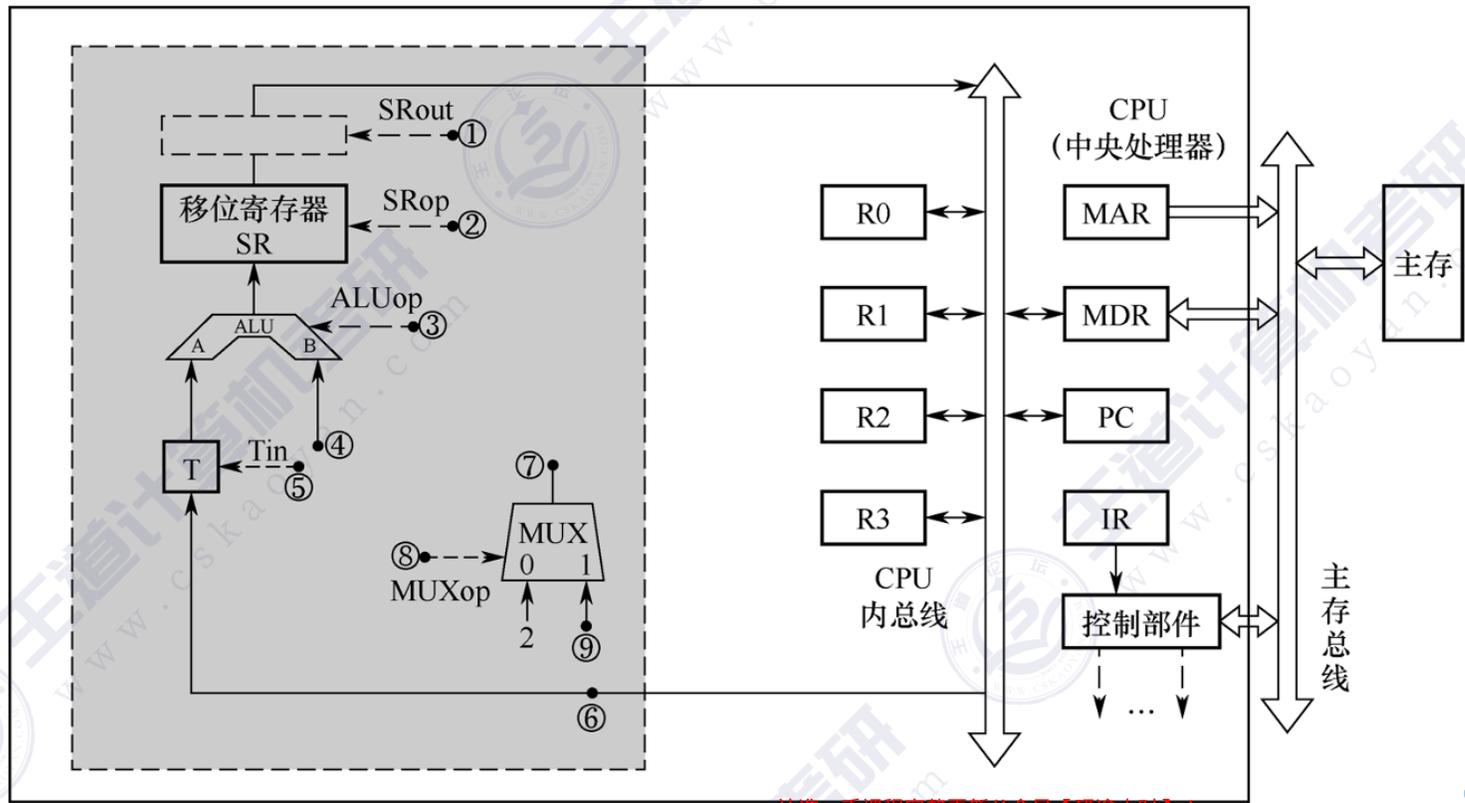
认准一手课程完整更新公众号【研途小时】!

2022年真题

- (1) 设 ALU 的输入端 A、B 及输出端 F 的最高位分别为 A_{15} 、 B_{15} 及 F_{15} ，FR 中的符号标志和溢出标志分别为 SF 和 OF，则 SF 的逻辑表达式是什么？A 加 B、A 减 B 时 OF 的逻辑表达式分别是什么？要求逻辑表达式的输入变量为 A_{15} 、 B_{15} 及 F_{15} 。
- (2) 为什么要设置暂存器 Y 和 Z？
- (3) 若 GPRs 的输入端 rs、rd 分别为所读、写的通用寄存器的编号，则 GPRs 中最多有多少个通用寄存器？rs 和 rd 来自图中的哪个寄存器？已知 GPRs 内部有一个地址译码器和一个多路选择器，rd 应连接地址译码器还是多路选择器？
- (4) 取指令阶段（不考虑 PC 增量操作）的控制信号序列是什么？若从发出主存读命令到主存读出数据并传送到 MDR 共需 5 个时钟周期，则取指令阶段至少需要几个时钟周期？
- (5) 图中控制信号由什么部件产生？图中哪些寄存器的输出信号会连到该部件的输入端？

2015年真题

43. (13分) 某16位计算机的主存按字节编码, 存取单位为16位; 采用16位定长指令字格式; CPU采用单总线结构, 主要部分如下图所示。图中R0~R3为通用寄存器; T为暂存器; SR为移位寄存器, 可实现直送(mov)、左移一位(left)和右移一位(right)3种操作, 控制信号为SRop, SR的输出由信号SRout控制; ALU可实现直送A(mova)、A加B(add)、A减B(sub)、A与B(and)、A或B(or)、非A(not)和A加1(inc)7种操作, 控制信号为ALUop。



认准一手课程完整更新公众号【研速小时】!

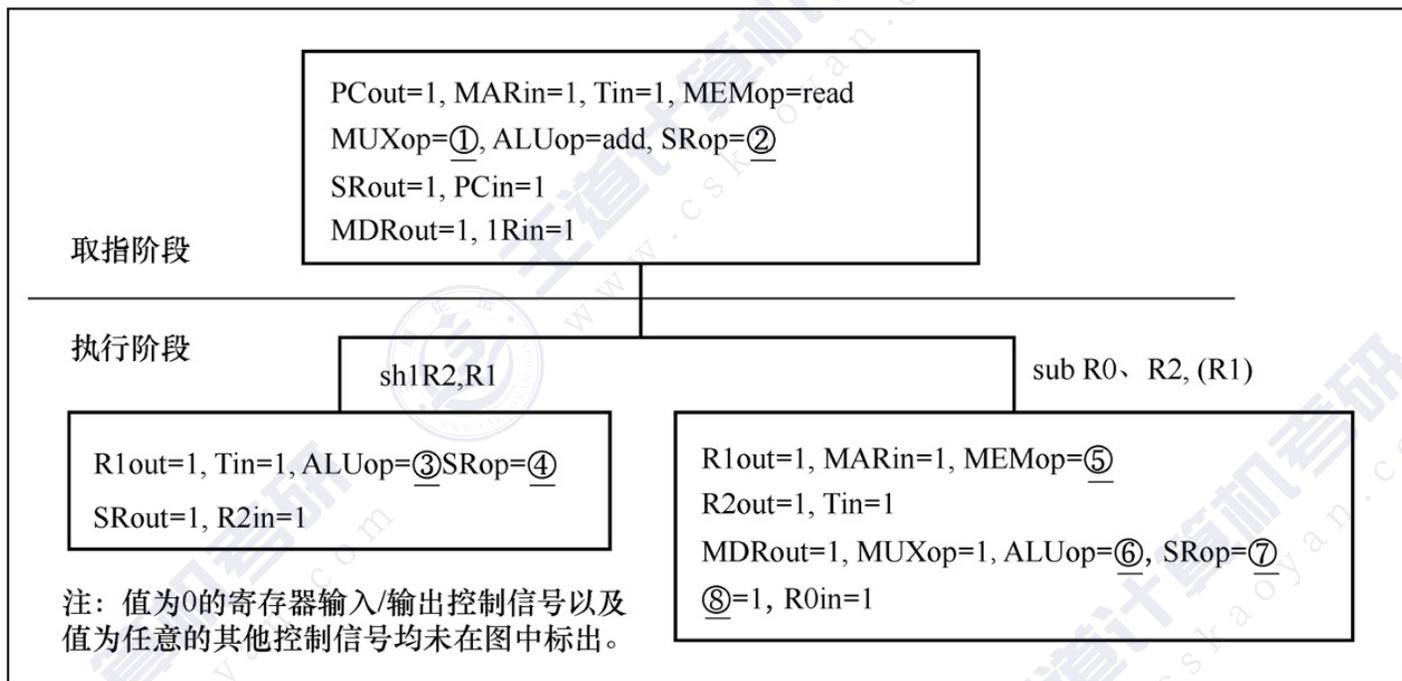
2015年真题

请回答下列问题。←

- 1) 图中哪些寄存器是程序员可见的? 为何要设置暂寄存器 T? ←
- 2) 控制信号 ALUop 和 SRop 的位数至少各是多少? ←
- 3) 控制信号 SRout 所控制部件的名称或作用是什么? ←
- 4) 端点①~⑨中, 哪些端点须连接到控制部件的输出端? ←
- 5) 为完善单总线数据通路, 需要在端点①~⑨中相应的端点之间添加必要的连线。写出连线的起点和终点, 以正确表示数据的流动方向。←

2015年真题

44. (10分) 题 43 中描述的计算机, 其部分指令执行过程的控制信号如题 44 图 a 所示。←

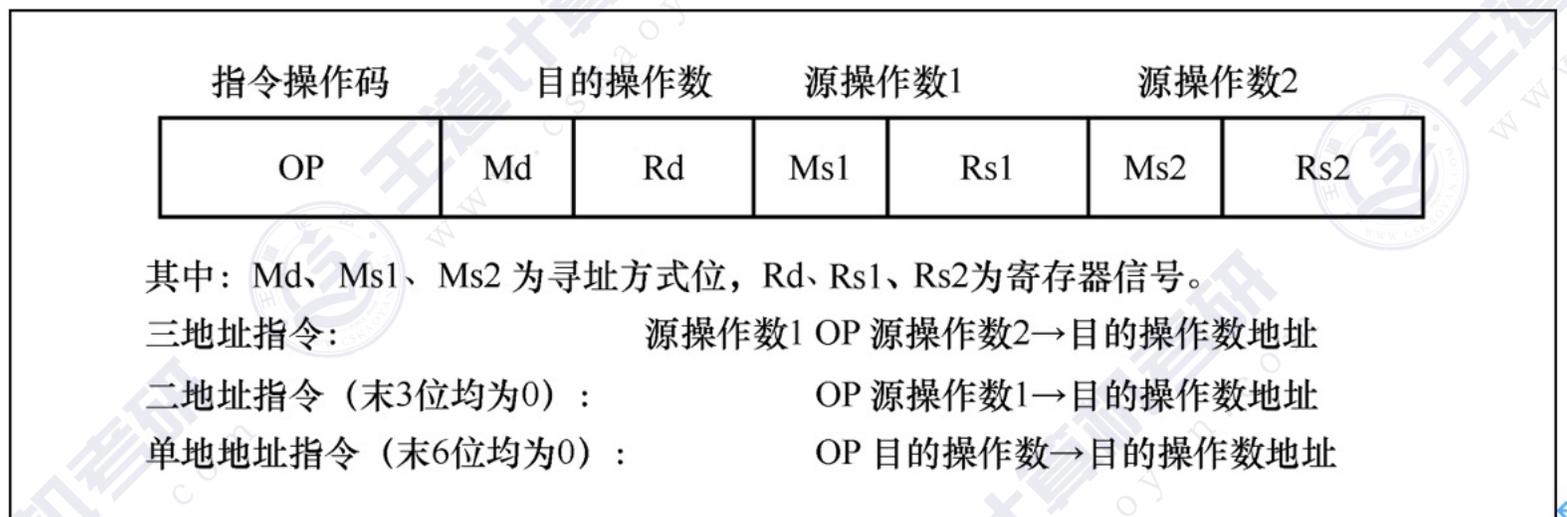


题 44 图 a 部分指令控制信号←

该机指令格式如题 44 图 b 所示, 支持寄存器直接和寄存器间接两种寻址方式, 寻址方式位分别为 0 和 1, 通用寄存器 R0~R3 的编号分别为 0、1、2 和 3。←

2015年真题

该机指令格式如题 44 图 b 所示，支持寄存器直接和寄存器间接两种寻址方式，寻址方式位分别为 0 和 1，通用寄存器 R0~R3 的编号分别为 0、1、2 和 3。



题 44 图 b 指令格式

请回答下列问题。

1) 该机的指令系统最多可定义多少条指令？

2015年真题

2) 若 `inc`、`shl` 和 `sub` 指令的操作码分别为 01H、02H 和 03H，则以下指令对应的机器代码各是什么？ ←

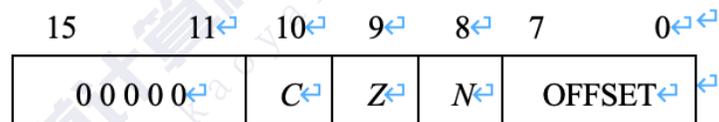
```
inc R1           ; R1 + 1 → R1 ←
shl R2, R1       ; (R1) << 1 → R2 ←
sub R3, (R1), R2 ; ((R1)) - (R2) → R3 ←
```

3) 假设寄存器 X 的输入和输出控制信号分别为 `Xin` 和 `Xout`，其值为 1 表示有效，为 0 表示无效（如 `PCout = 1` 表示 PC 内容送总线）；存储器控制信号为 `MEMop`，用于控制存储器的读（read）和写（write）操作。写出题图 a 中标号①~⑧处的控制信号或控制信号的取值。 ←

4) 指令“`sub R1, R3, (R2)`”和“`inc R1`”的执行阶段至少各需要多少个时钟周期？ ←

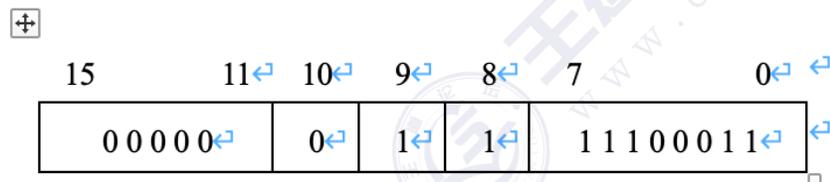
2013年真题

44. (14分) 某计算机采用16位定长指令字格式，其CPU中有一个标志寄存器，其中包含进位/借位标志CF、零标志ZF和符号标志NF。假定为该机设计了条件转移指令，其格式如下：



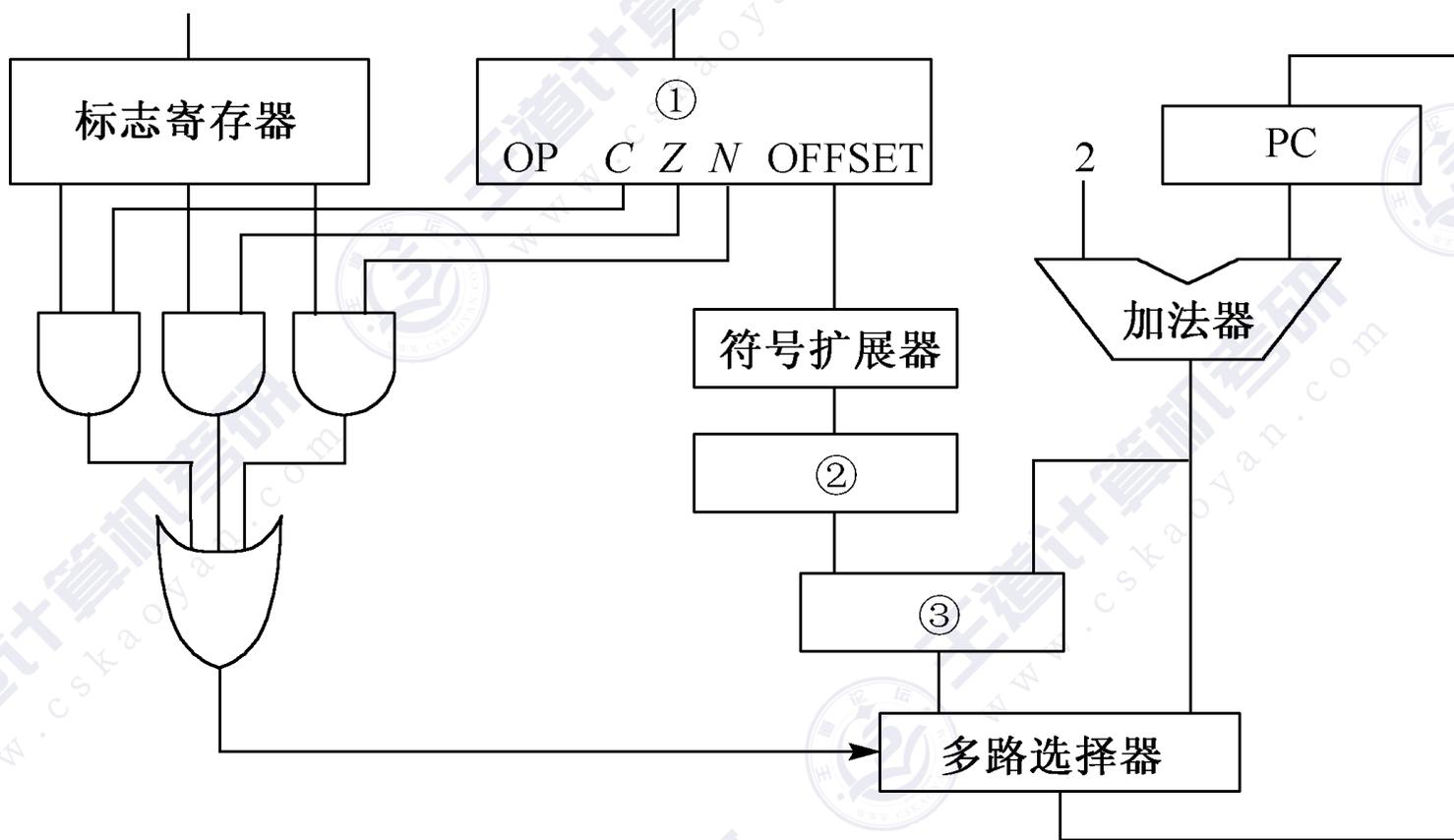
其中，00000为操作码OP；C、Z和N分别为CF、ZF和NF的对应检测位，某检测位为1时表示需检测对应标志位，需检测的标志位中只要有一个为1就转移，否则不转移。例如，若 $C=1$ ， $Z=0$ ， $N=1$ ，则需检测CF和NF的值，当 $CF=1$ 或 $NF=1$ 时发生转移；OFFSET是相对偏移量，用补码表示。转移执行时，转移目标地址为 $(PC)+2+2\times OFFSET$ ；顺序执行时，下条指令地址为 $(PC)+2$ 。请回答下列问题。

- (1) 该计算机存储器按字节编址还是按字编址？该条件转移指令向后（反向）最多可跳转多少条指令？
- (2) 某条件转移指令的地址为200CH，指令内容如下图所示，若该指令执行时 $CF=0$ ， $ZF=0$ ， $NF=1$ ，则该指令执行后PC的值是多少？若该指令执行时 $CF=1$ ， $ZF=0$ ， $NF=0$ ，则该指令执行后PC的值又是多少？请给出计算过程。



- (3) 实现“无符号数比较小于等于时转移”功能的指令中，C、Z和N应各是什么？
- (4) 以下是该指令对应的数据通路示意图，要求给出图中部件①~③的名称或功能说明。

2013年真题



扩展（了解一哈）

➤ 条件码:

OF (Overflow Flag) 溢出标志。溢出时为1, 否则置0。

SF (Sign Flag) 符号标志。结果为负时置1, 否则置0。

ZF (Zero Flag) 零标志, 运算结果为0时ZF位置1, 否则置0。

CF (Carry Flag) 进位标志, 进位时置1, 否则置0。

AF (Auxiliary carry Flag) 辅助进位标志, 记录运算时第3位 (半个字节) 产生的进位置。有进位时1, 否则置0。

PF (Parity Flag) 奇偶标志。结果操作数中1的个数为偶数时置1, 否则置0。

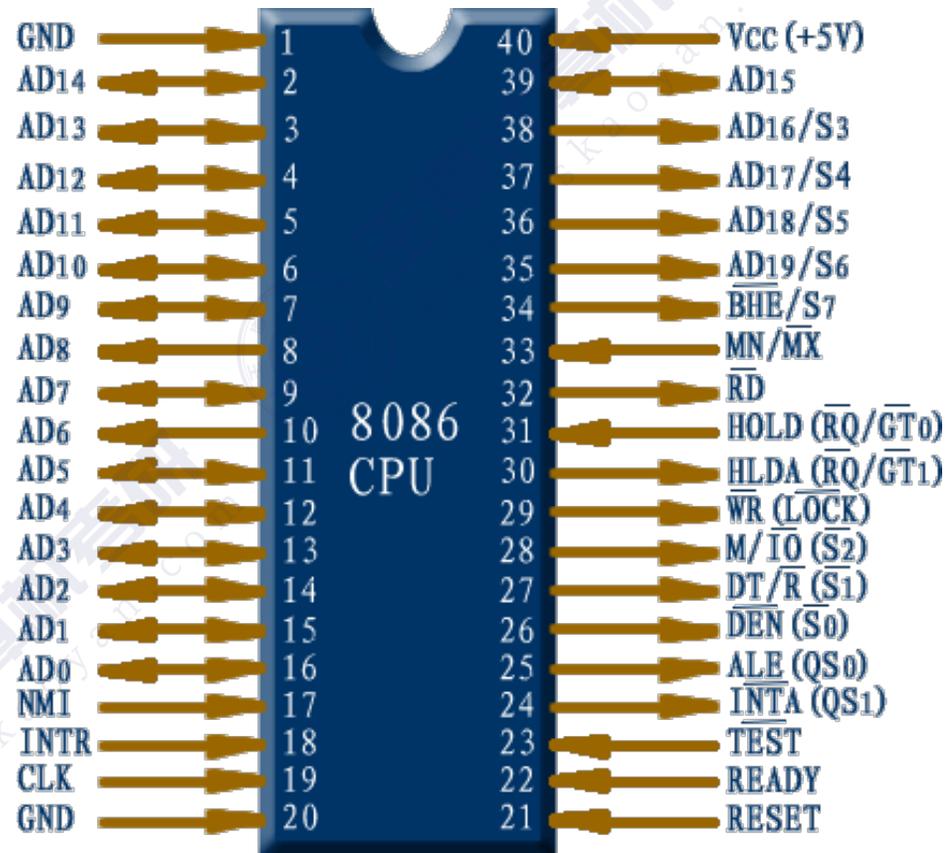
➤ 控制标志位:

DF (Direction Flag) 方向标志, 在串处理指令中控制信息的方向。

IF (Interrupt Flag) 中断标志。

TF (Trap Flag) 陷阱标志。

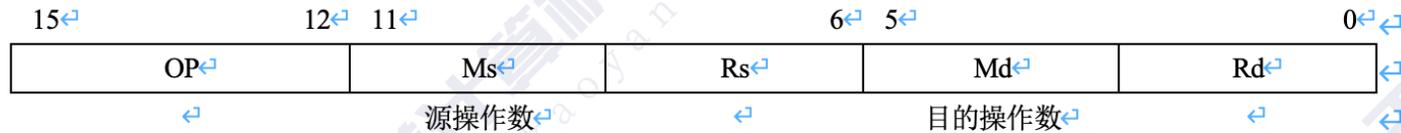
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF



- NMI:不可屏蔽中断请求信号。常用于处理电源掉电紧急情况。
- INTR:可屏蔽中断请求信号。

2010年真题

43. (11分) 某计算机字长为16位，主存地址空间大小为128KB，按字编址。采用单字长指令格式，指令各字段定义如下图所示。



转移指令采用相对寻址方式，相对偏移量用补码表示，寻址方式定义见下表。

Ms/Md	寻址方式	助记符	含义
000B	寄存器直接	Rn	操作数 = (Rn)
001B	寄存器间接	(Rn)	操作数 = ((Rn))
010B	寄存器间接、自增	(Rn)+	操作数 = ((Rn)), (Rn) + 1 → Rn
011B	相对	D(Rn)	转移目标地址 = (PC) + (Rn)

注：(X)表示存储器地址 X 或寄存器 X 的内容。

请回答下列问题：

1) 该指令系统最多可有多少条指令？该计算机最多有多少个通用寄存器？存储器地址寄存器 (MAR) 和存储器数据寄存器 (MDR) 至少各需要多少位？

2) 转移指令的目标地址范围是多少？

3) 若操作码 0010B 表示加法操作 (助记符为 add)，寄存器 R4 和 R5 的编号分别为 100B 和 101B，R4 的内容为 1234H，R5 的内容为 5678H，地址 1234H 中的内容为 5678H，地址 5678H 中的内容为 1234H，则汇编语言为 “add(R4), (R5)+” (逗号前为源操作数，逗号后为目的操作数) 对应的机器码是什么 (用十六进制表示)？该指令执行后，哪些寄存器和存储单元中的内容会改变？改变后的内容是什么？

2009年真题

44. (13分) 某计算机字长为16位, 采用16位定长指令字结构, 部分数据通路结构如下图所示, 图中所有控制信号为1时表示有效、为0时表示无效。例如, 控制信号MDRinE为1表示允许数据从DB打入MDR, MDRin为1表示允许数据从内总线打入MDR。假设MAR的输出一直处于使能状态。加法指令“ADD(R1), R0”的功能为 $(R0) + ((R1)) \rightarrow (R1)$, 即将R0中的数据与R1的内容所指主存单元的数据相加, 并将结果送入R1的内容所指主存单元中保存。←

下表给出了上述指令取指和译码阶段每个节拍(时钟周期)的功能和有效控制信号, 请按表中描述方式用表格列出指令执行阶段每个节拍的功能和有效控制信号。←

时钟←	功能←	有效控制信号←
C1←	$MAR \leftarrow (PC)$ ←	PCout, MARin←
C2←	$MDR \leftarrow M(MDR)$ ← $PC \leftarrow (PC) + 1$ ←	MemR, MDRinE, PC+1←
C3←	$IR \leftarrow (MDR)$ ←	MDRout, IRin←
C4←	指令译码←	无←

2009年真题

