

Lecture 13: 知识 - 指令格式

指令格式

指令的基本格式

概念： 指令格式

- 概念：
 - 一条指令就是机器语言的一个语句，它是一组有意义的二进制代码；
 - 一条指令通常要包括**操作码**字段和**地址码**字段两部分；
 - 根据**地址码数量**的不同，可以分为不同的指令结构；
- 零地址指令：
 - 格式：
 - 只有操作指令；
 - 情况：
 - 1. 不需要操作数：
 - 如 空操作、停机、关中断 等指令
 - 2. 堆栈计算机：
 - 两个操作数隐含存放在栈顶和次栈顶，计算结果压回栈顶；
 - 如数据结构的后缀表达式；
- 一地址指令：
 - 格式：
 - 操作指令 + 地址1
 - 情况：
 - 1. 只需要单操作数：
 - 如 加 1、减 1、取反、求补 等；
 - 2. 需要两个操作数，但其中一个操作数隐含在某个寄存器；
 - 如隐含在 ACC；
 - 含义：
 - $(ACC)OP(A1) \rightarrow ACC$
 - 注意：
 - 注：A1 指某个主存地址， $(A1)$ 表示 A1 所指向的地址中的内容
- 二地址指令：
 - 格式：
 - 操作指令 + 地址1(目的操作数) + 地址2(源操作数)
 - 情况：
 - 常用于需要两个操作数的 算术运算、逻辑运算 相关指令；
 - 含义：
 - $(A1)OP(A2) \rightarrow A1$

- 三地址指令：
 - 格式：
 - 操作指令 + 地址1 + 地址2 = 地址3(结果)
 - 情况：
 - 常用于需要两个操作数的算术运算、逻辑运算相关指令；
 - 含义：
 - $(A1)OP(A2) \rightarrow A3$
- 四地址指令：
 - 格式：
 - 操作指令 + 地址1 + 地址2 = 地址3(结果), 地址4=下一条将要执行指令的地址
 - 情况：
 - 执行指令后, 将 PC 的值修改位 A4 所指地址；
- 地址码与位数：
 - n 位地址码的直接寻址范围 = 2^n
 - 若指令总长度固定不变, 则地址码数量越多, 寻址能力越差；

指令长度

概念：指令字长与存储字长

- 指令字长：
 - 一条指令的总长度；
 - 半字长指令、单字长指令、双字长指令 $\rightarrow$ 指令长度是机器字长的多少倍；
 - 指令字长会影响取指令所需时间；
 - 如：机器字长=存储字长=16bit, 则取一条双字长指令需要两次访存；
- 存储字长：
 - 一个存储单元中的二进制代码位数；
 - 通常和 MDR 位数相同；

概念：机器字长

- **真题：**2020年
- 概念：机器字长
 - 机器字长是指计算机能够直接处理的二进制数据的位数, 是 CPU 内部用于整数运算的数据通路的宽度；
- 总结：位数与机器字长相同
 - (1) CPU 总线的宽度
 - (1) ALU
 - (2) 通用寄存器位数；
- 总结：位数可能与机器字长不同
 - (1) 指令寄存器 IR
 - 与指令字长的长度相同；
 - (2) 浮点数寄存器

- 和浮点数长度相同

概念：定/变长指令字结构

- 概念：
 - 指的是整条指令的长度是否变换；
- 定长指令字结构：
 - 指令系统中所有指令的长度都相等；
- 变长指令字结构：
 - 指令系统中各种指令的长度不等；

概念：定/变长操作码

- 定长操作码：
 - 指令系统中所有指令的操作码长度都相同；
 - 控制器的译码电路设计简单，但灵活性较低；
- 可变长操作码：
 - 指令系统中各指令的操作码长度可变；
- 定长指令字结构 + 可变长操作码 -> 扩展操作码指令格式；

概念：指令分类

- 数据传送：
 - LOAD/STORE
- 算术运算：
- 移位操作：
- 转移操作：
 - JMP：无条件转移
 - JZ：条件转移
- 输入输出操作

扩展操作码的指令格式

基本格式： 定长指令字结构 + 可变长操作码

- 即：不同地址数的指令使用不同长度的操作码

扩展操作码概念

概念：扩展操作码

- 概念：
 - 为了能够让 CPU 可以识别当前的指令是哪种地址指令格式，需要留一定位数作为扩展位，用于识别其格式；
 - 如果开头 4 位全为 0，且后续的位数中出现非全 0，则肯定是三地址指令；
 - 如果开头的 8 位全 0，且后续的位数中出现非全 0，则肯定是二地址指令；
 - 同理一地址指令；

- 图示：

4位操作码	0000	A ₁	A ₂	A ₃	15条三地址指令
	0001	A ₁	A ₂	A ₃	
	⋮	⋮	⋮	⋮	
	1110	A ₁	A ₂	A ₃	
8位操作码	1111	0000	A ₂	A ₃	15条二地址指令
	1111	0001	A ₂	A ₃	
	⋮	⋮	⋮	⋮	
	1111	1110	A ₂	A ₃	
12位操作码	1111	1111	0000	A ₃	15条一地址指令
	1111	1111	0001	A ₃	
	⋮	⋮	⋮	⋮	
	1111	1111	1110	A ₃	
16位操作码	1111	1111	1111	0000	16条零地址指令
	1111	1111	1111	0001	
	⋮	⋮	⋮	⋮	
	1111	1111	1111	1111	

分析：三地址指令

- 如果指令字长为 16 位，每个地址码占 4 位，且是三地址指令结构；
- 前 4 位为基本操作码字段 OP，另有 3 个 4 位长的地址字段 A₁、A₂ 和 A₃；
- 4 位基本操作码若全部用于三地址指令，则有 16 条；
 - 指令的个数由操作码的种类决定，因为操作码有四位变化，且 1111 表示二地址码，则此时指令个数为 $2^4 - 1 = 15$
 - 因为至少须将 1111 留作扩展操作码之用，所以需要减 1、即三地址指令为 15 条：

分析：二地址指令

- 和三地址指令类似，需要留 1111 1111 作为扩展操作码使用，因此二地址指令为 15 条；

分析：一地址指令

- 同理，一地址指令为 15 条；

分析：零地址指令

- 零地址指令为 16 条；

概念：扩展操作码注意事项

- (1) 不允许短码是长码的前缀，即短操作码不能与长操作码的前面部分的代码相同；
 - 即：哈夫曼树的前缀编码；
 - 因此需要扩展操作码来识别；
- (2) 各指令的操作码一定不能重复；
- (3) 通常情况下，对使用频率较高的指令，分配较短的操作码，对使用频率较低的指令，分配较长的操作码，从而尽可能减少指令译码和分析的时间；

扩展操作码设计举例

设计：设指令字长固定为 16 位，试设计一套指令系统满足；

- 概念：
 - 已知前缀的前提下，设计某地址指令的个数；
 - 比如情况 C：
 - 当前已经有 1111, 11 六个前缀被使用；
 - 并且当前需要设计一地址指令，因此后背的 XXXX 被作为地址使用；
 - 所以现在还剩下 $16-6-4=6$ 位可以使用；
 - 并且：XXXX, XXXX, XXXX, XXXX 中第一部分和第四部分不可以使用；
 - 第二部分为：11XX，第三部分为 XXXX，当出现 1111, 1111 时不可以使用；
- 图示：

- a) 有15条三地址指令
共 $2^4=16$ 种状态
留出 $16-15=1$ 种
- b) 有12条二地址指令
共 $1 \times 2^4=16$ 种
留出 $16-12=4$ 种
- c) 有62条一地址指令
共 $4 \times 2^4=64$ 种
留出 $64-62=2$ 种
- d) 有32条零地址指令
共 $2 \times 2^4=32$ 种

	0000 - 1110	A ₁	A ₂	A ₃
1111 XXXX XXXX XXXX	1111	0000 - 1011	A ₁	A ₂
1111 11XX XXXX XXXX	1111	1100 - 1110 1111	0000 - 1111 0000 - 1101	A ₁
1111 1111 111X XXXX	1111	1111	1110 - 1111	0000 - 1111

分析：真题：2022年

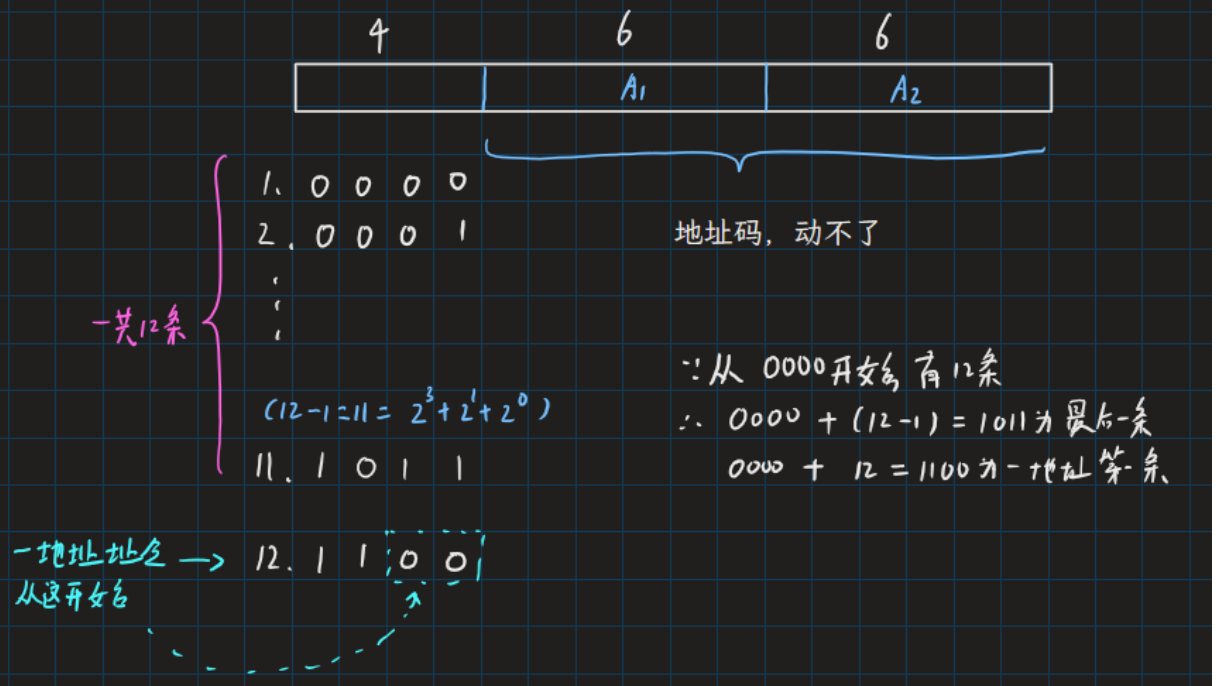
- 题目：

【2022 统考真题】
设计某指令系统时，假设采用 16 位定长指令字格式，操作码使用扩展编码方式，地址码为 6 位，包含零地址、一地址和二地址 3 种格式的指令。若二地址指令有 12 条，一地址指令有 254 条，则零地址指令的条数最多为____

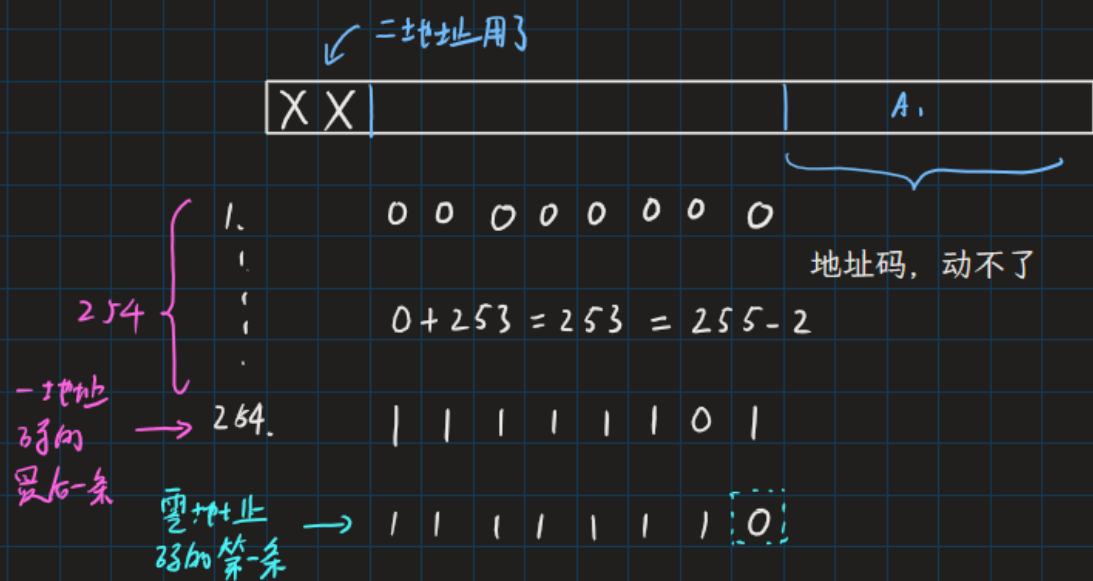
$2^9=16, 1位$

- 解析：

二地址指令:



一地址指令: 个数 = 254, $2^8 = 256$



零地址指令:

X X X X X X X X X X

1, 0 0 0 0 0 0 0
:
:
:
1

$$n_1 = 2^6 + 2^5 + \dots + 2^0 = 2^7 - 1 = 127$$

1 1 1 1 1 1 1

$$n_2: 127 + 1 = 128$$

- 总结: 分析方法
 - 从 n 地址指令, 开始分析地址格式;
 - 将 n 地址指令从第 1 条指令一直排列至第 m 条指令 (若题目中给出 n 地址指令为 m 条时), 得到第 m 条指令的二进制排列形式;
 - n 地址指令的第 m+1 条指令格式即为 n-1 地址指令格式的第一条指令, 重复以上分析过程;

扩展操作码分析

分析: 不同操作码比较

- 定长操作码:
 - 概念:
 - 在指令字的最高位部分分配固定的若干位 (定长) 表示操作码;
 - 一般 n 位操作码字段的指令系统最大能够表示 2^n 条指令。
 - 特点:
 - 优点:
 - 定长操作码对于简化计算机硬件设计, 提高指令译码和识别速度很有利;
 - 缺点:
 - 指令数量增加时会占用更多固定位, 留给表示操作数地址的位数受限;
- 扩展操作码 (不定长操作码):
 - 概念:
 - 全部指令的操作码字段的位数不固定, 且分散地放在指令字的不同位置上;
 - 最常见的变长操作码方法是扩展操作码, 使操作码的长度随地址码的减少而增加, 不同地址数的指令可以具有不同长度的操作码, 从而在满足需要的前提下, 有效地缩短指令字长;
 - 特点:

- 优点：
 - 在指令字长有限的前提下仍保持比较丰富的指令种类；
- 缺点：
 - 增加了指令译码和分析的难度，使控制器的设计复杂化；

指令注释风格

