扩展操作码指令格式

操作码指出指令中该指令应该执行什么性质的操作和具有何种功能。

操作码是识别指令、了解指令功能与区分操作数地址内容的组成和使用方法等的关键信息。例如，指出是算术加运算，还是减运算；是程序转移，还是返回操作。

操作码分类：

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

指令由操作码和若干个地址码组成。

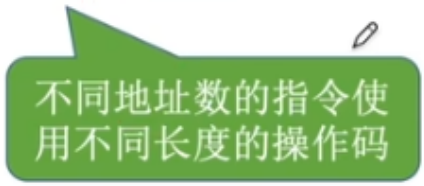
定长指令字结构：指令系统中所有指令的长度都相等

变长指令字结构：指令系统中各种指令的长度不等

定长操作码：指令系统中所有指令的操作码长度都相同

可变长操作码：指令系统中各指令的操作码长度可变

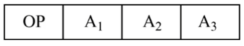
定长指令字结构+可变长操作码 →扩展操作码指令格式



扩展操作码

扩展操作码举例

指令字长为16位，每个地址码占4位：
前4位为基本操作码字段OP，另有3个4位长的地址字段A_1、A_2和A_3。

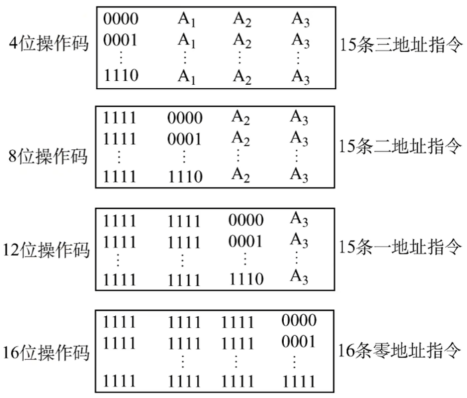


4位基本操作码若全部用于三地址指令，则有16条。
但至少须将1111留作扩展操作码之用，即三地址指令为15条；

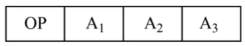
1111 1111留作扩展操作码之用，二地址指令为15条；

1111 1111 1111留作扩展操作码之用，一地址指令为15条；

零地址指令为16条。



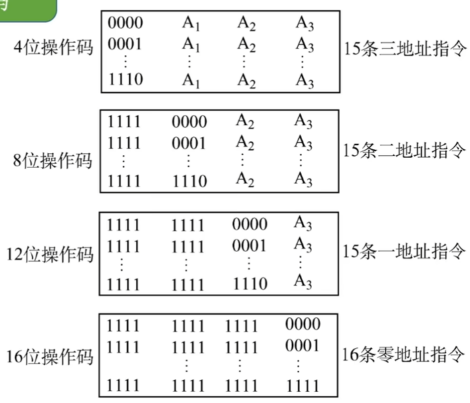
还有其他扩展操作码设计方法。



对比哈夫曼树“前缀编码”

- 在设计扩展操作码指令格式时，必须注意以下两点：
- 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。
- 2）各指令的操作码一定不能重复。

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。



扩展操作码举例

设指令字长固定为16位，试设计一套指令系统满足：

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 ×2^4=16种
留出16-12=4种
c) 有62条一地址指令
共4 ×2^4=64种
留出64-62=2种
d) 有32条零地址指令
共2 ×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

设地址长度为n，上一层留出m种状态，下一层可扩展出m×2^n种状态