

# 计算机系统结构 · Hw2

计01 容逸朗 2020010869

## 2.11

- 不同方法的编码结果如下：

指令	频数 $p_i$	赫夫曼编码	码长 $l_{1,i}$	3/3/3 扩展编码	码长 $l_{2,i}$	2/7 扩展编码	码长 $l_{3,i}$
ADD	0.43	0	1	00	2	00	2
CLA	0.22	10	2	01	2	01	2
SUB	0.13	110	3	10	2	1000	4
JOM	0.07	11100	5	1100	4	1001	4
JMP	0.06	11101	5	1101	4	1010	4
STO	0.05	11110	5	1110	4	1011	4
CIL	0.02	111110	6	111100	6	1100	4
STP	0.01	1111110	7	111101	6	1101	4
SHR	0.01	1111111	7	111110	6	1110	4

- 采用赫夫曼编码，上述九条指令的平均码长为  $L = \sum_{i=1}^9 p_i \cdot l_{1,i} = 2.42$ ；
- 采用 3/3/3 扩展编码，上述九条指令的平均码长为  $L = \sum_{i=1}^9 p_i \cdot l_{2,i} = 2.52$ ；
- 采用 2/7 扩展编码，上述九条指令的平均码长为  $L = \sum_{i=1}^9 p_i \cdot l_{3,i} = 2.70$ 。

## 2.12

- 双地址指令操作码字段有 4 位，操作条数为 A，剩余  $2^4 - A$  个编码供单地址操作使用；
- 单地址指令比双地址指令少一个地址码位，因此其操作码字段可以比双地址的长 6 位；通过和上一步余下的地址相互组合，可以得到所有单地址的操作码；
- 因此单地址指令最多可以有  $(2^4 - A) \times 2^6 = 1024 - 64A$  条。

## 2.13

- 首先为三地址指令编码，12 位的指令字长除去 3 个 3 位的地址码后剩余 3 位，由于系统共有 4 条三地址指令，编码后剩余一位需要作为三地址及其他指令的区分；  
(即三地址指令为 000 R1 R2 R3 - 011 R1 R2 R3，其余指令为 1 开头的指令字)
- 此时未分配的指令字长为 11 位，考虑 16 条零地址指令，扣除和单地址指令地址码位重叠部分（3 位，表示 8 条指令）后，在剩余 8 位中还需要占据 2 个操作码；
- 最后考虑一地址指令，除去 1 个 3 位的地址码后剩余 8 位，可以提供 256 个不同的操作码，扣除零地址占用的 2 个后，最多只能提供 254 个操作码给单地址指令使用。

- 因此，不能对 255 条单地址指令作扩展编码，但可以对 254 条单地址指令作扩展编码，一个可行方案如下所述：

指令类型	编码	条数
三地址指令	0 00 [R1] [R2] [R3] - 0 11 [R1] [R2] [R3]	4
单地址指令	1 00000000 [R1] - 1 11111101 [R1]	254
零地址指令	1 11111110 000 - 1 11111111 111	16