计算机组成原理课后习题答案(第　三　章)

第三章 作业解答

3.1 已知$[x]_{补}$、$[y]_{补}$，计算$[x+y]_{补}$和$[x-y]_{补}$，并判断溢出情况。

（1） $[x]_{补}=0.11011$　　$[y]_{补}=0.00011$　　（2） $[x]_{补}=0.10111$　　$[y]_{补}=1.00101$

（3） $[x]_{补}=1.01010$　　$[y]_{补}=1.10001$

解：（1） $[x]_{补}=0.11011$　　$[y]_{补}=0.00011$　　$[-y]_{补}=1.111101$

$[x+y]_{补}=0.11011+0.00011=0.11110$

$[x-y]_{补}=0.11011+1.111101=0.11000$

（2）$[x]_{补}=0.10111$　　$[y]_{补}=1.00101$　　$[-y]_{补}=0.11011$

$[x+y]_{补}=0.10111+1.00101=1.11100$

$[x-y]_{补}=0.10111+0.11011=1.10010$　溢出

（3）$[x]_{补}=1.01010$　　$[y]_{补}=1.10001$　　$[-y]_{补}=0.01111$

$[x+y]_{补}=1.01010+1.10001=0.11011$　溢出

$[x-y]_{补}=1.01010+0.01111=1.11001$

3.2 已知$[x]_{补}$、$[y]_{补}$，计算$[x+y]_{变形补}$和$[x-y]_{变形补}$，并判断溢出情况。

（1） $[x]_{补}=100111$　　$[y]_{补}=111100$　　（2） $[x]_{补}=011011$　　$[y]_{补}=110100$

（3） $[x]_{补}=101111$　　$[y]_{补}=011000$

解：（1）$[x]_{变形补}=1100111$　　$[y]_{变形补}=1111100$　　$[-y]_{变形补}=0000100$

$[x+y]_{变形补}=1100111+1111100=1100011$

$[x-y]_{变形补}=1100111+0000100=1101011$

（2）$[x]_{变形补}=0011011$　　$[y]_{变形补}=1110100$　　$[-y]_{变形补}=0001100$

$[x+y]_{变形补}=0011011+1110100=0001111$

$[x-y]_{变形补}=0011011+0001100=0100111$　溢出

（3） $[x]_{变形补}=1101111$　　$[y]_{变形补}=0011000$　　$[-y]_{变形补}=1101000$

$[x+y]_{变形补}=1101111+0011000=0000111$

$[x-y]_{变形补}=1101111+1101000=1010111$　溢出

3.3 设某机字长为 8 位，给定十进制数：x＝＋49，y＝－74。试按补码运算规则计算下列各题，并判断溢出情况。

（1） $[x]_{补}+[y]_{补}$　　（2） $[x]_{补}-[y]_{补}$

（3） $[-x]_{补}+[\frac{1}{2}y]_{补}$　　（4） $[2x-\frac{1}{2}y]_{补}$

（5） $[\frac{1}{2}x+\frac{1}{2}y]_{补}$　　（6） $[-x]_{补}+[2y]_{补}$

解：$[x]_{补}=00110001$　$[y]_{补}=10110110$　　$[-y]_{补}=01001010$

（1） $[x]_{补}+[y]_{补}=00110001+10110110=11100111$

（2） $[x]_{补}-[y]_{补}=00110001+01001010=01111011$

（3） $[-x]_{补}+[\frac{1}{2}y]_{补}=11001111+11011011=10101010$

（4） $[2x-\frac{1}{2}y]_{补}=01100010+00100101=10000111$　溢出

（5） $[\frac{1}{2}x+\frac{1}{2}y]_{补}=00011000+11011011=11110011$

（6） $[-x]_{补}+[2y]_{补}$　$[2y]_{补}$溢出，故$[-x]_{补}+[2y]_{补}$的结果溢出

3.4 分别用原码一位乘法和补码一位乘法计算$[x\times y]_{原}$和$[x\times y]_{补}$。

（1） x＝0.11001　　y＝0.10001　　（2） x＝0.01101　　y＝－0.10100

（3） x＝－0.10111　　y＝0.11011　　（4） x＝－0.01011　　y＝－0.11010

解：（1）$[x\times y]_{原}=0.0110101001$　$[x\times y]_{补}=0.0110101001$

（2）$[x\times y]_{原}=1.010*******$　$[x\times y]_{补}=1.1011111100$

（3）$[x\times y]_{原}=1.1001101101$　$[x\times y]_{补}=1.0110010011$

（4）$[x\times y]_{原}=0.010*******$　$[x\times y]_{补}=0.010*******$

3.5 分别用原码两位乘法和补码两位乘法计算$[x\times y]_{原}$和$[x\times y]_{补}$。

（1） x＝0.11001　　y＝0.10001　　（2） x＝0.10101　　y＝－0.01101

（3） x＝－0.01111　　y＝0.11101　　（4） x＝－0.01001　　y＝－0.10010

解：　**（1）**$[x\times y]_{原}=0.0110101001$　$[x\times y]_{补}=0.0110101001$

（2）$[x\times y]_{原}=1.010*******$　$[x\times y]_{补}=1.1011101111$

（3）$[x\times y]_{原}=1.0110110011$　$[x\times y]_{补}=1.1001001101$

（4）$[x\times y]_{原}=0.0010100010$　$[x\times y]_{补}=0.0010100010$

3.6　分别用原码不恢复余数法和补码不恢复余数法计算$[x/y]_{原}$和$[x/y]_{补}$。(1) (4)

(1)　x＝0.01011　　y＝0.10110

$[x/y]_{原}$＝0.10000　　$[x/y]_{补}$＝0.10000　　or　　$[x/y]_{补}$＝0.10001

(2)　x＝0.10011　　y＝－0.11101

$[x/y]_{原}$＝1.10100　　$[x/y]_{补}$＝1.01100　　or　　$[x/y]_{补}$＝1.01011

(3)　x＝－0.10111　　y＝－0.11011

$[x/y]_{原}$＝0.11100　　$[x/y]_{补}$＝0.11101　　or　　$[x/y]_{补}$＝0.11100

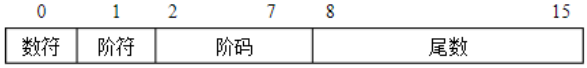
(4)　x＝＋10110　　y＝－00110

$[x/y]_{原}$＝100011　　$[x/y]_{补}$＝111101

3.7 在进行浮点加减运算时，为什么要进行对阶？说明对阶的方法和理由。

答：

3.8 已知某模型机的浮点数据表示格式如下：



其中，浮点数尾数和阶码的基值均为 2，均采用补码表示。

（1） 求该机所能表示的规格化最小正数和非规格化最小负数的机器数表示及其所对应的十进制真值。

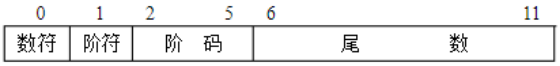
（2）已知两个浮点数的机器数表示为EF80H和FFFFH，求它们所对应的十进制真值。

（3）已知浮点数的机器数表示为：

$[x]_{补}$＝1 1111001 00100101，$[y]_{补}$＝1 1110111 00110100

试按浮点加减运算算法计算$[x \pm y]_{补}$。

3.9　已知某机浮点数表示格式如下：



其中，浮点数尾数和阶码的基值均为 2，阶码用移码表示，尾数用补码表示。设：

$x=0.110101\times2^{-001}$　　$y=-0.100101\times2^{+001}$

试用浮点运算规则计算 x＋y、x－y、x×y、x/y。（要求写出详细运算步骤，并进行规格化）。

解：机器数　$[x]_{补}$＝0 01111 110101　$[y]_{补}$＝1 10001 011011　$[-y]_{补}$＝0 10001 100101

（1）x＋y　　机器数 **$[x+y]_{补}$＝1 10000 010000**　　**$x+y=-0.110000\times2^{0}$**

对阶：　$[\Delta e]_{移}=[e_x]_{移}+[-e_y]_{补}$＝01111＋11111＝01110，$\Delta e=e_x-e_y$＝－00010

小阶对大阶：$[x]_{补}$＝0 10001 001101

$[x+y]_{补}$＝1 10000 010000　　$x+y=-0.110000\times2^{0}$

（2）x－y

$[x-y]_{补}$＝0 10001 110010　　$x-y=0.110010\times2^{1}$

（3）x×y　　**$x\times y=-0.111110\times2^{-001}=-0.111110\times2^{-1}$**

阶码相加：$[e_x+e_y]_{移}=[e_x]_{移}+[e_y]_{补}$＝01111＋00001＝10000

尾数可采用定点补码乘法（双符号位）：$[S_x\times S_y]_{补}=[S_x]_{补}\times[S_y]_{补}$＝11.100001010111

规格化：**$[x\times y]_{补}$＝1 01111 000010　　$x\times y=-0.111110\times2^{-001}=-0.111110\times2^{-1}$**

（4）x/y

尾数$|S_x|>|S_y|$，S_x右移得：$[S_x]_{补}$＝00.011010，$[e_x]_{移}$＝10000，

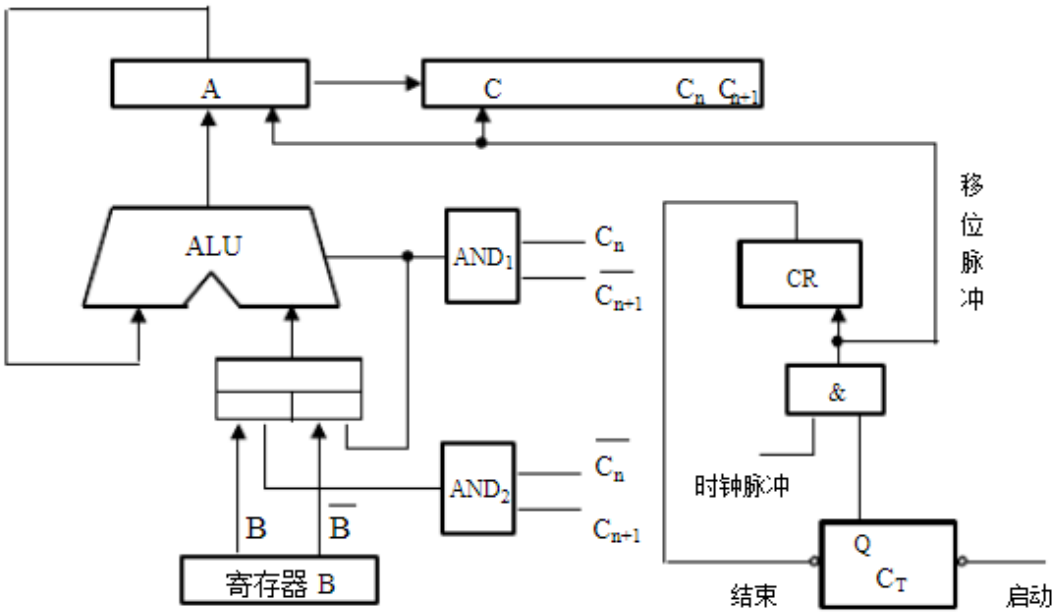
阶码相减：$[e_x-e_y]_{移}=[e_x]_{移}+[-e_y]_{补}$＝10000＋11111＝01111

尾数用补码不恢复余数法：$[S_x/S_y]_{补}=[S_x]_{补}/[S_y]_{补}$＝1.010011（恒置 1）　OR　1.010100（校正）

规格化：**$[x/y]_{补}$＝1 01111 010011**　　OR　**1 01111 010100**

$x/y=-0.101101\times2^{-001}$　OR　**-0.101100×2^{-001}**

3.10



00. 0 0 0 0 0 0 1 0 0 1 1 0 0

→ 00. 0 0 0 0 0 0 0 1 0 0 1 1 0

−x 00. 1 1 0 0 1

00. 1 1 0 0 1

→ 00. 0 1 1 0 0 1 0 1 0 0 1 1

→ 00. 0 0 1 1 0 0 1 0 1 0 0 1

+x 11. 0 0 1 1 1

11. 0 1 1 0 1

→ 11. 1 0 1 1 0 1 0 1 0 1 0 0

→ 11. 1 1 0 1 1 0 1 0 1 0 1 0

−x 00. 1 1 0 0 1

00. 1 0 1 0 0 0 1 0 1 0 1 0

得 $[X\times Y]_{补}=0.1010001010$　　$X\times Y=0.1010001010$

寄存器	A	B	C
运算初态	00 00000	11 00111	1001100
运算终态	00 10100	11 00111	0101010

3.11　说明定点补码和浮点补码加减运算的溢出判断方法。

答：(1) 定点补码加减运算的溢出判断方法：

① 根据两个操作数的符号与结果的符号判别溢出：$OVR=\bar{x}_f\bar{y}_f s_f+x_f y_f\bar{s}_f=(x_f\oplus s_f)(y_f\oplus s_f)$

② 根据两数相加时产生的进位判别溢出：$OVR=C_f\oplus C_1$

③ 根据变形补码运算后的符号判别溢出：

$s_{f1}s_{f2}=00$，表示结果为正数，无溢出；

$s_{f1}s_{f2}=11$，表示结果为负数，无溢出；

$s_{f1}s_{f2}=01$，表示结果为正溢出；

$s_{f1}s_{f2}=10$，表示结果为负溢出。

(2) 浮点补码加减运算的溢出判断方法

浮点补码加减运算的溢出通常是指浮点数上溢，浮点数是否溢出是由阶码是否大于浮点数所能表示的最大正阶来判断的。

例如，设浮点数的阶码采用补码表示，双符号位，这时浮点数的溢出与否可由阶码的符号进行判断：

若阶码 $[j]_{补}=\underbrace{01}_{符号}\times\times\cdots\times$，则表示出现上溢，需作溢出处理；

若阶码 $[j]_{补}=10\times\times\cdots\times$，则表示出现下溢，按机器零处理。

3.12 说明定点原码除法和定点补码除法运算的溢出判断方法。

答：定点原码不恢复余数除法的溢出算法为：

因为在定点小数运算时，若|被除数|>|除数|，则除法将发生溢出，不能进行除法运算。因此，如果在第一次上商时得到的商为“1”，则表示除法发生溢出。

定点补码不恢复余数除法的溢出算法为：

当被除数$[x]_{补}$与除数$[y]_{补}$同号时，如果余数$[r]_{补}$与$[y]_{补}$同号，且上商为“1”，则表示商溢出。当被除数$[x]_{补}$与除数$[y]_{补}$异号时，如果余数$[r]_{补}$与$[y]_{补}$异号，且上商为“0”，则表示商溢出。

3.13 比较舍入方法中截断法、恒置“1”法和0舍1入法的优缺点。

答：(1) 截断法（恒舍法）

截断法是：将右移移出的值一律舍去，余下的不作任何改变。该方法简单，精度较低。

⑵ 0 舍 1 入法

0 舍 1 入法的方法是：若右移时被丢掉数位的最高位为 0，则舍去；若右移时被丢掉数位的最高位为 1，则将 1 加到保留的尾数的最低位。

“0 舍 1 入”法类似于十进制数的“四舍五入”。其主要优点是单次舍入引起的误差小，精度较高；其缺点是加 1 时需多做一次运算，而且可能造成尾数溢出，需要再次右规。

⑶ 末位恒置 1 法

末位恒置 1 法也称冯·诺依曼舍入法。其方法是：尾数右移时，无论被丢掉的数位的最高位为 0 还是为 1，都将保留的尾数的最低位恒置为 1。

末位恒置 1 法的主要优点是舍入处理不用做加法运算，方法简单、速度快且不会有再次右规的可能，并且没有积累误差，是常用的舍入方法。其缺点是单次舍入引起的误差较大。

3.14 利用用十进制加减运算算法计算下列各题：

（1） 125＋436＝？ （2） 125－436＝？ （3） 436－125＝？

解：（1） 125＋436＝561

（2） 125－436＝－311

（3） 436－125＝311

3.15 参照第二章表 2-12 给出的余 3 码的编码规则，设计利用余 3 码进行十进制加法的修正逻辑。

答：余 3 码的编码规则：

十进制数	余 3 码
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

余 3 码十进制加法器运算结果的修正关系

十进制数	用余 3 码表示的 十进制和数 C'_4 $F_4F_3F_2F_1$	两个余 3 码按二进制规 则相加得到的和数 C_4 $S_4S_3S_2S_1$	修正逻辑
0	0 0011	0 0110	加 “1101” 修 正 “-3”
1	0 0100	0 0111	
2	0 0101	0 1000	
3	0 0110	0 1001	
4	0 0111	0 1010	
5	0 1000	0 1011	
6	0 1001	0 1100	
7	0 1010	0 1101	
8	0 1011	0 1110	
9	0 1100	0 1111	
10	1 0011	1 0000	加 “0011” 修 正 “+3”
11	1 0100	1 0001	
12	1 0101	1 0010	
13	1 0110	1 0011	
14	1 0111	1 0100	
15	1 1000	1 0101	
16	1 1001	1 0110	
17	1 1010	1 0111	
18	1 1011	1 1000	
19	1 1100	1 1001	

3.16　设有一个 16 位定点补码运算器，数据最低位的序号为 1。运算器可实现下述功能：

（1）　A±B→A

（2）　B×C→A、C（乘积高位在 A 中）

（3）　A÷B→C（商在 C 中）

请设计并画出运算器第 3 位及 A、C 寄存器第三位输入逻辑。加法器本身逻辑可以不画，原始操作数输入问题可以不考虑。

解：见附页

3.19 设一个 8 位寄存器中的内容为十六进制数 C5H，连续经过一次算术右移、一次逻辑左移、一次大循环右移、一次小循环左移。写出每次移位后寄存器的内容和进位标志 C 的状态。

解：C5H＝11000101

	C	寄存器
一次算术右移：	1	11100010
一次逻辑左移：	1	11000100
一次大循环右移：	0	11100010
一次小循环左移：	1	11000101

3.20 已知寄存器 A 的内容为 01011010，寄存器 B 的内容为 11011011，分别写出经过下列移位操作后，寄存器 A、B 中的内容。

（1）算术左移两位。

（2）逻辑左移两位。

（3）算术右移两位。

（4）逻辑右移两位。

解：寄存器 A 的内容为 01011010　　寄存器 B 的内容为 11011011

	c			c		
（1）算术左移两位	1	01101000	（1）算术左移两位。	1	01101100	移位溢出
（2）逻辑左移两位。	1	01101000	（2）逻辑左移两位。	1	01101100	
（3）算术右移两位。	1	00010110	（3）算术右移两位。	1	11110110	
（4）逻辑右移两位。	1	00010110	（4）逻辑右移两位。	1	00110110	

3.21 选择题

（1）　运算器的核心部分是__C__。

A. 数据总线　　B. 累加寄存器　　C. 算术逻辑运算单元　　D. 多路开关

（2）　在浮点运算中下面的论述正确的是__C__。

A. 对阶时应采用向左规格化

B. 对阶时可以使小阶向大阶对齐，也可以使大阶向小阶对齐

C. 尾数相加后可能会出现溢出，但可采用向右规格化的方法得出正确结论

D. 尾数相加后不可能得出规格化的数

（3）　当采用双符号位进行数据运算时，若运算结果的双符号位为 01，则表明运算__B__。

A. 无溢出　　B. 正溢出　　C. 负溢出　　D. 不能判别是否溢出

（4）　补码加法运算的规则是__B__。

A. 操作数用补码表示，符号位单独处理

B. 操作数用补码表示，连同符号位一起相加

C. 操作数用补码表示，将加数变补，然后相加

D. 操作数用补码表示，将被加数变补，然后相加

（5）　原码乘除法运算要求__C__。

A. 操作数必须都是正数　　B. 操作数必须具有相同的符号位

C. 对操作数符号设有限制　　D. 以上都不对

（6）　进行补码一位乘法时，被乘数和乘数均用补码表示，运算时__A__。

A. 首先在乘数最末位 y_n 后增设附加位 y_{n+1}，且初始 $y_{n+1}=0$，再依照 y_ny_{n+1} 的值确定下面的运算。

B. 首先在乘数最末位 y_n 后增设附加位 y_{n+1}，且初始 $y_{n+1}=1$，再依照 y_ny_{n+1} 的值确定下面的运算。

C. 首先观察乘数符号位，然后决定乘数最末位 y_n 后附加位 y_{n+1} 的值，再依照 y_ny_{n+1} 的值确定下面的运算。

D. 不应在乘数最末位 y_n 后增设附加位 y_{n+1}，而应直接观察乘数的末两位 $y_{n-1}y_n$ 确定下面的运算。

（7） 下面对浮点运算器的描述中正确的是__A__。

A. 浮点运算器由阶码部件和尾数部件实现。

B. 阶码部件可实现加、减、乘、除四种运算。

C. 阶码部件只能进行阶码的移位操作。

D. 尾数部件只能进行乘法和加法运算。

（8） 若浮点数的阶码和尾数都用补码表示，则判断运算结果是否为规格化数的方法是__C__。

A. 阶符与数符相同为规格化数。

B. 阶符与数符相异为规格化数。

C. 数符与尾数小数点后第一位数字相异为规格化数。

D. 数符与尾数小数点后第一位数字相同为规格化数。

（9） 已知$[x]_{补}=1.01010$，$[y]_{补}=1.10001$，下列答案正确的是__D__。

A. $[x]_{补}+[y]_{补}=1.11011$　　B. $[x]_{补}+[y]_{补}=0.11011$

C. $[x]_{补}-[y]_{补}=0.11011$　　D. $[x]_{补}-[y]_{补}=1.11001$

（10） 下列叙述中概念正确的是__D__。

A. 定点补码运算时，其符号位不参加运算。

B. 浮点运算中，尾数部分只进行乘法和除法运算。

C. 浮点数的正负由阶码的正负符号决定。

D. 在定点小数一位除法中，为了避免溢出，被除数的绝对值一定要小于除数的绝对值。

3.22 填空题

（1） 在补码加减运算中，符号位与数据__①__参加运算，符号位产生的进位__②__。

答：① 按同样规则一起　② 自动丢失

（2） 在采用变形补码进行加减运算时，若运算结果中两个符号位__①__，表示发生了溢出。若结果的两个符号位为__②__，表示发生正溢出；为__③__，表示发生负溢出。

答：① -55　② 11110010　③ +73　④ 01001001

（3） 在原码一位乘法的运算过程中，符号位与数值位__①__参加运算，运算结果的符号位等于__②__。

答：① 分别　② 两操作数的符号的模 2 加（异或）

（4） 浮点乘除法运算的运算步骤包括：__①__、__②__、__③__、__④__和__⑤__。

答：① 阶码运算　② 溢出判断　③ 尾数乘除运算　④ 结果规格化处理　⑤ 舍入处理

（5） 在浮点运算过程中，如果运算结果的尾数部分不是__①__形式，则需要进行规格化处理。设尾数采用补码表示形式，当运算结果__②__时，需要进行右规操作；当运算结果__③__时，需要进行左规操作。

答：① 规格化　② 溢出　③ 不是规格化数

（6） 将两个 8421BCD 码相加，为了得到正确的十进制运算结果，需要对结果进行修正，其修正方法是①__。

答：① 两个 8421 码相加后，若相加的和数＜10，则不需修正，按二进制规则相加的结果就是正确的 8421 码的和数；若相加的和数≥10，则需在二进制相加的结果上加“0110”进行修正。

（7） 浮点运算器由__①__和__②__两部分组成，它们本身都是定点运算器，其中①要求能够进行__③__运算；②要求能够进行__④__运算。

答：① 阶码部件　② 尾数部件　③ 加减　④ 加减乘除

（8） 设有一个 16 位的数据存放在由两个 8 位寄存器 AH 和 AL 组成的寄存器 AX 中，其中数据的高 8 位存放在 AH 寄存器中，低 8 位存放在 AL 寄存器中。现需要将 AX 中的数据进行一次算术左移，其操作方法是：先对__①__进行一次__②__操作，再对__③__进行一次__④__操作。

答：① AL　② 算术左移　③ AH　④ 带进位循环左移

3.23 是非题

（1）运算器的主要功能是进行加法运算。　×

（2）加法器是构成运算器的主要部件，为了提高运算速度，运算器中通常都采用并行加法器。 √

（3）在定点整数除法中，为了避免运算结果的溢出，要求|被除数|＜|除数|。 √

（4）浮点运算器中的阶码部件可实现加、减、乘、除运算。 ×

（5）根据数据的传递过程和运算控制过程来看，阵列乘法器实现的是全并行运算。 √

（6）逻辑右移执行的操作是进位标志位移入符号位，其余数据位依次右移 1 位，最低位移入进位标志位。×

第四章 作业解答

4.1 静态 MOS 存储器与动态 MOS 存储器存储信息的原理有何不同？为什么动态 MOS 存储器需要刷新？一般有哪几种刷新方式？

答：静态 MOS 存储器利用一个双稳态触发器存储一个二进制位，只要不断电就可以保持其中存储的二进制数据不丢失。

动态 MOS 存储器使用一个 MOS 管和一个电容来存储一位二进制信息。用电容来存储信息减少了构成一个存储单位所需要的晶体管的数目。

由于动态 MOS 存储器中的电容会产生漏电，因此 DRAM 存储器芯片需要频繁的刷新操作。

动态存储器的刷新方式通常有：

集中式刷新方式、分散式刷新方式、异步式刷新方式

4.2 某一 64K×1 位的动态 RAM 芯片，采用地址复用技术，则除了电源和地引脚外，该芯片还应有那些引脚？各为多少位？

答：地址线：采用地址复用技术，可为 16/2＝8 位

数据线：1 位；读写线 R/W：1 位；片选信号 $\overline{CS}$：1 位

或 行选通信号 $\overline{RAS}$：1 位；列选通信号 $\overline{CAS}$：1 位

4.3 在页模式 DRAM 中，“打开一页”指什么？在打开一页的操作中，信号 $\overline{RAS}$ 和 $\overline{CAS}$ 的作用是什么？

答：在页模式 DRAM 中，打开一页是指选中存储矩阵中的一行。

在打开一页的操作中，信号 $\overline{RAS}$ 的作用是：将行地址锁存到行地址译码器，选中存储矩阵中的一行。

信号 $\overline{CAS}$ 的作用是：将列地址锁存到列地址译码器，选中存储矩阵中的某一行中的一列。

阶码和尾数均用补码表示时的浮点数表示范围

	数符	阶符	阶码 (m位)	尾数 (n位)	真 值
非规格化最小正数	0	1	00…00	00…01	$2^{-n} \times 2^{-2^m}$
规格化最小正数	0	1	00…00	10…00	$2^{-1} \times 2^{-2^m}$
最大正数	0	0	11…11	11…11	$(1-2^{-n}) \times 2^{(2^m-1)}$
非规格化最大负数	1	1	00…00	11…11	$-2^{-n} \times 2^{-2^m}$
规格化最大负数	1	1	00…00	01…11	$-(2^{-1}+2^{-n}) \times 2^{-2^m}$
最小负数	1	0	11…11	00…00	$-1 \times 2^{(2^m-1)}$