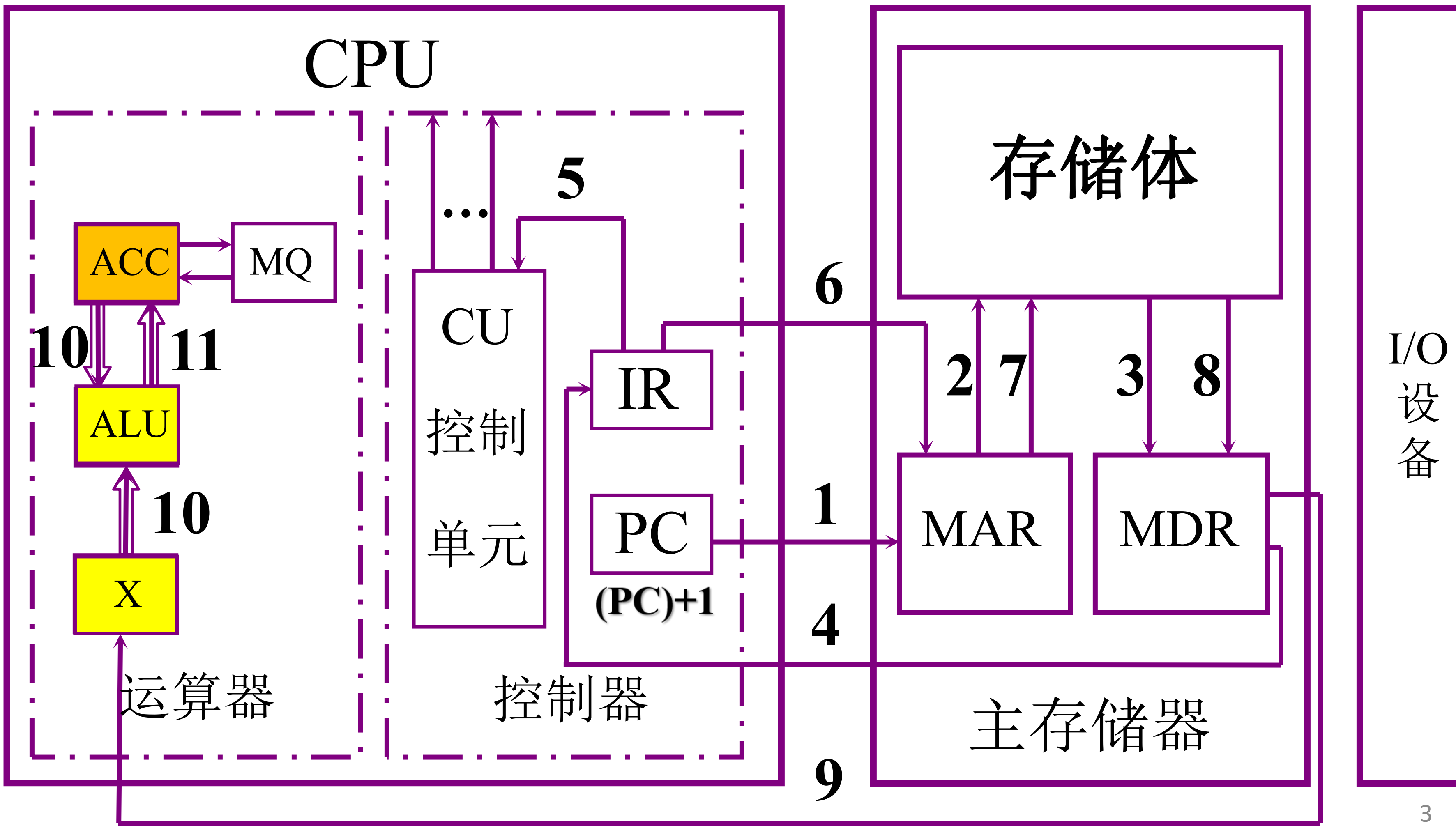
习题解答

hw1

课后习题: 1.7, 1.9, 1.10, 1.11, 9.8

参考解答：以ADD M为例

1.9 画出主机框图，分别以存数指令“STA M”和加法指令“ADD M”（M 均为主存地址）为例，在图中按序标出完成该指令（包括取指阶段）的信息流程（如$\xrightarrow{①}$）。假设主存容量为 256 M × 32 位，在指令字长、存储字长、机器字长相等的条件下，指出图中各寄存器的位数。



知识点解析

- 知识点
  - 取指和运算流程
  - 各模块名称、功能
  - 概念区分：指令字长、存储字长、机器字长

参考解答

1.10 根据迭代公式$\sqrt{x}=\frac{1}{2}\left(y_n+\frac{x}{y_n}\right)$，设初态 $y_0=1$，要求精度为 ε，试编制求$\sqrt{x}$的解题程序(指令系统自定)，并结合所编程序简述计算机的解题过程。

操作码	操作性质
000001	取数：将指令地址码指示的存储单元中的数取到ACC
000010	存数：将ACC中数存到指令地址码指示的存储单元中
000100	加：将ACC中数与指令地址码指示的存储单元中的数相加，结果存于ACC中
000111	除：将ACC中数与指令地址码指示的存储单元中的数相除，结果存于ACC中
001000	减：将ACC中数与指令地址码指示的存储单元中的数相减，结果存于ACC中
001001	取绝对值：将ACC中数取绝对值后存入ACC
001010	将ACC中数与0做比较，小于则执行下一条，大于则跳转

- 开放性题目，讲清楚思路即可
- 注意：说明具体需要哪些步骤，每个步骤什么指令，及每条指令的操作码、操作数的地址码

主存地址	指令		注释
	操作码	地址码	
0	000001	17	取数y_{n+1}至ACC
1	000010	16	存数，将y_{n+1}存于y_n单元中
2	000001	13	取数x至ACC
3	000111	16	(除) 初y_n得$^x/_{y_n}$存于ACC
4	000100	16	加y_n得$y_n+{}^x/_{y_n}$存于ACC
5	000111	15	除2得$(y_n+{}^x/_{y_n})/2$存于ACC
6	000010	17	存数，将$(y_n+{}^x/_{y_n})/2$存入y_{n+1}单元中
7	001000	16	减y_n得y_{n+1}-y_n存于ACC中
8	001001	-	$\lvert y_{n+1}-y_n\rvert$存于ACC中
9	001000	14	$\lvert y_{n+1}-y_n\rvert-\varepsilon$存于ACC中
10	001010	0	比较
11	000101	17	打印
12	000110	-	停机
13	x		
14	ε		
15	2		
16	y_n		
17	y_{n+1}		开始之前将y0放入此单元

知识点解析

- 知识点
  - 指令和数据存储方式
  - 指令格式和指令系统
  - 循环实现方法

参考解答

9.8 某计算机的主频为 6 MHz，各类指令的平均执行时间和使用频度如下表所示，试计算该机的速度（单位用 MIPS 表示），若上述 CPU 芯片升级为 10 MHz，则该机的运行速度又为多少？

指令类别	存取	加、减、比较、转移	乘除	其他
平均指令执行时间	0.6 μs	0.8 μs	10 μs	1.4 μs
使用频度	35%	45%	5%	15%

- 平均指令执行时间

$$0.6\mu s\times 35\% + 0.8\mu s\times 45\% + 10\mu s\times 5\% + 1.4\mu s\times 15\% = 1.28\mu s$$

- 该机的速度

$$\frac{1}{1.28us} = \frac{1}{1.28\times 10^{-6}s} = \frac{1}{1.28}MIPS = 0.78125\ MIPS$$

- 主频提升为10MHz，该机的运行速度

$$0.78125\,MIPS\times\frac{10\,MHz}{6\,MHz} = 1.302\,MIPS$$

知识点解析

- 知识点
  - CPI和MIPS定义和计算公式
  - 时钟频率、时钟周期、CPI、MIPS关系
    - 对于确定的处理器和指令实现，提高主频不会改变CPI
    - 但主频提高使得时钟周期变短，指令执行时间变短，因此MIPS会提高
    - CPI的单位不是μs,而是时钟周期数
    - MIPS是指每秒处理的百万指令数，MHz已经有百万的含义

hw2

课后习题：6.3, 6.5, 6.6, 6.9

参考解答

6.3 设 x 为整数，$[x]_{补}=1,x_1x_2x_3x_4x_5$，若要求 $x<-16$，试问 $x_1\sim x_5$ 应取何值？

答: $[x]_{补}$的符号位为1，所以x一定是负数

$x=-16$时，$[x]_{补}=1,10000$

当使用补码表示时，绝对值越大，数值越小

因此 $x<-16$时，$x^*>10000$，

又x为负数时 $x^*=\overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}\,\overline{x_5}+1$ 得$\overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}\,\overline{x_5}>01111$

则$\overline{x_1}=1$，$x_2\sim x_5$任意

即$x_1=0$，$x_2\sim x_5$任意

知识点解析

- 知识点
  - 补码的定义和计算公式
  - 负数补码比较
  - 补码和真值、原码之间关系

参考解答

6.9 当十六进制数 9BH 和 FFH 分别表示为原码、补码、反码、移码和无符号数时，所对应的十进制数各为多少(设机器数采用 1 位符号位)?

(1) 移码定义

将每一个数值加上一个偏置常数

$$[x]_{移} = 2^n + x \quad (2^n > x \geq -2^n)$$

x 为真值，n 为 **整数的位数**

- 题目中：9BH, FFH表示移码时，求真值是多少。根据定义可以直接使用$[x]_{移} - 2^n$得到真值
- 另，移码也与补码除符号位外相同，可以通过补码求得移码

知识点解析

- 知识点
  – 移码的定义和计算公式
  – 移码和真值、补码之间关系
  – 各类码概念的区分，计算关系

hw3

课后习题：6.10, 6.15, 6.16

参考解答

6.15　什么是机器零？若要求全 0 表示机器零，浮点数的阶码和尾数应采用什么机器数形式？

- 若要求用“全0”表示浮点机器零，则浮点数的阶码应用移码、尾数用补码表示。
- 此时阶码为最小阶、尾数为零，而移码的最小码值正好为“0”，补码的零的形式也为“0”，拼起来正好为一串0的形式。

知识点解析

- 知识点
  - 机器零概念
  - 浮点数的表示方法
  - 阶码和尾数不同码表示的公式

参考解答

6.16 设机器数字长为 16 位，写出下列各种情况下它能表示的数的范围。设机器数采用 1 位符号位，答案均用十进制数表示。

（1）无符号数。

（2）原码表示的定点小数。

（3）补码表示的定点小数。

（4）补码表示的定点整数。

（5）原码表示的定点整数。

（6）浮点数的格式为：阶码 6 位（含 1 位阶符），尾数 10 位（含 1 位数符）。分别写出正数和负数的表示范围。

（7）浮点数格式同（6），机器数采用补码规格化形式，分别写出其对应的正数和负数的真值范围。

- （1）部分同学没有考虑无符号小数
- 0.1111 1111 1111 1111
- 0 ~ 1 − 2^−16，即0~0.99998

（6）浮点数的阶码6位（含1位阶符），尾数10位（含1位数符），正数和负数的表示范围。

答：当阶码和尾数均采用原码，非规格化表示时，

<u>最大正数</u>：$\mathbf{0,11111;0.111111111}$，

即$(1-2^{-9})\times 2^{(2^5-1)}=(1-2^{-9})\times 2^{31}$

<u>最小正数</u>：$\mathbf{1,11111;0.000000001}$，

即$2^{-9}\times 2^{-(2^5-1)}=2^{-9}\times 2^{-31}$

则正数范围为：$2^{-9}\times 2^{-31}$~$(1-2^{-9})\times 2^{31}$

即正数范围为：9.09×10^{-13}~2.14×10^{9}

（6）浮点数的阶码6位（含1位阶符），尾数10位（含1位数符），正数和负数的表示范围。

答：当阶码和尾数均采用原码，非规格化表示时，

最大负数：$\mathbf{1,11111;1.000000001}$，

即$-2^{-9}\times 2^{-(2^{5}-1)}=-2^{-9}\times 2^{-31}$

最小负数：$\mathbf{0,11111;1.111111111}$，

即$-(1-2^{-9})\times 2^{(2^{5}-1)}=-(1-2^{-9})\times 2^{31}$

则负数范围为：$-(1-2^{-9})\times 2^{31}\sim -2^{-9}\times 2^{-31}$

即负数范围为 $-2.14\times 10^{9}\sim -9.09\times 10^{-13}$

（7）浮点数的阶码6位（含1位阶符），尾数10位（含1位数符），采用补码规格化形式。

答：当采用补码规格化形式时，若无隐含位，

最大正数：$0,11111;0.111111111$，

即$(1-2^{-9})\times 2^{(2^5-1)}=(1-2^{-9})\times 2^{31}$

最小正数：$1,00000;0.100000000$，

即$2^{-1}\times 2^{-2^5}=2^{-1}\times 2^{-32}$

则正数范围为：$2^{-1}\times 2^{-32}$~$(1-2^{-9})\times 2^{31}$

即正数范围为：1.16×10^{-10}~2.14×10^{9}

（7）浮点数的阶码6位（含1位阶符），尾数10位（含1位数符），采用补码规格化形式。

答：当采用补码规格化形式时，若无隐含位，

最小负数：$\mathbf{0,11111;1.\ 000000000}$，（特例：−1为规格化的数，$-\frac{1}{2}$不是规格化的数），即

$-1\times 2^{(2^5-1)}=-2^{31}$

最大负数：$\mathbf{1,00000;1.011111111}$，即

$-(\mathbf{2^{-1}+2^{-9}})\times\left(2^{-2^5}\right)=-(2^{-1}+2^{-9})\times 2^{-32}$

则负数范围为：$-2^{31}\sim-(2^{-1}+2^{-9})\times 2^{-32}$

即负数范围为：$-2.15\times 10^9\sim -1.17\times 10^{-10}$

知识点解析

- 知识点
  - 定点数的原码补码表示
  - 浮点数的阶码尾数计算
  - 浮点数的规格化
  - 浮点数的表示范围和精度
  - 区分一些特殊值的定义：如-1和-1.0
- 技巧
  - 认真读题目，例如给定机器字长一定有用

hw4

课后习题：6.17, 6.19

参考解答

6.17 设机器数字长为8位(含1位符号位),对下列各机器数进行算术左移一位、两位,算术右移一位、两位,讨论结果是否正确。

$[x]_{原}=0.0011010$;$[x]_{补}=0.1010100$;$[x]_{反}=1.0101111$;

$[x]_{原}=1.1101000$;$[x]_{补}=1.1101000$;$[x]_{反}=1.1101000$;

$[x]_{原}=1.0011001$;$[x]_{补}=1.0011001$;$[x]_{反}=1.0011001$。

- 原码：左移丢1时出错，右移丢1时有误差；
- 反码：正数同原码；负数左移丢0出错，右移丢0有误差；
- 补码：正数同原码；负数左移丢0出错，右移丢1有误差；

知识点解析

- 知识点
  - 算术左移和算术右移
  - 算术运算和逻辑运算区别

hw5

课后习题：6.20 , 6.21 , 6.23

参考解答

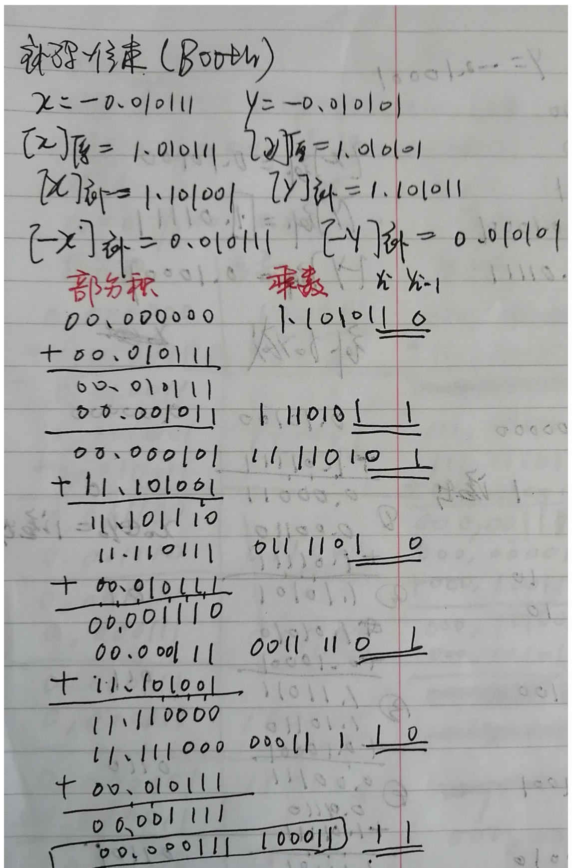
6.20 用原码一位乘、两位乘和补码一位乘(Booth 算法)、两位乘计算 $x \cdot y$。

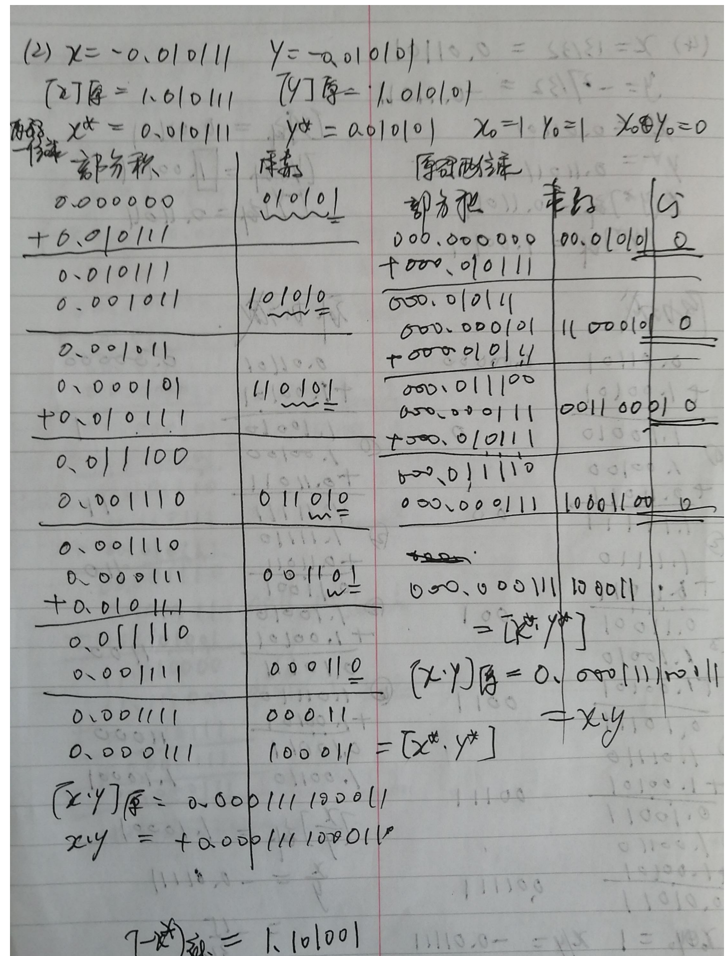
(1) $x=0.110111, y=-0.101110$。

(2) $x=-0.010111, y=-0.010101$。

(3) $x=19, y=35$。

(4) $x=0.11011, y=-0.11101$。





知识点解析

- 知识点
  - 各乘法算法的原理、计算公式
  - 各算法的对比
  - 算法的加法、移位次数

参考解答

6.21 用原码加减交替法和补码加减交替法计算 $x \div y$。

(1) $x = 0.100111, y = 0.101011$。

(2) $x = -0.10101, y = 0.11011$。

(3) $x = 0.10100, y = -0.10001$。

(4) $x = \frac{13}{32}, y = -\frac{27}{32}$。

(4) $x = 13/32 = 0.01101$

$y = -27/32 = -0.11011$

$x^* = 0.01101$

$y^* = 0.11011$

$[y^*]_补 = 0.11011$

$[-y^*]_补 = 1.00101$

原加减

$$\begin{array}{rl}
 & 0.01101 \quad\quad 0.00000 \\
+ & 1.00101 \\
\hline
 & 1.10010 \quad\quad 0 \\
① & 1.00100 \\
+ & 0.11011 \\
\hline
 & 1.11111 \quad\quad 00 \\
② & 1.11110 \\
+ & 0.11011 \\
\hline
 & 0.11001 \quad\quad 001 \\
③ & 1.10010 \\
+ & 1.00101 \\
\hline
 & 0.10111 \quad\quad 0011 \\
④ & 1.01110 \\
+ & 1.00101 \\
\hline
 & 0.10011 \quad\quad 00111 \\
⑤ & 1.00110 \\
+ & 1.00101 \\
\hline
 & 0.01011 \quad\quad 0.01111
\end{array}$$

$x_0 \oplus y_0 = 1 \quad x/y = -0.01111$

$= -\frac{15}{32}$

$r^* = 0.00000\,01011$

$[x]_补 = 0.01101$

$[y]_补 = \boxed{1}.00101$

$[-y]_补 = 0.11011$

补加减.

$$\begin{array}{rl}
 & 0.01101 \quad\quad 0.00000 \\
+ & 1.00101 \\
\hline
 & 1.10010 \quad\quad 1 \\
① & 1.00100 \\
+ & 0.11011 \\
\hline
 & 1.11111 \quad\quad 11 \\
② & 1.11110 \\
+ & 0.11011 \\
\hline
 & 0.11001 \quad\quad 110 \\
③ & 1.10010 \\
+ & 1.00101 \\
\hline
 & 0.10111 \quad\quad 1100 \\
④ & 1.01110 \\
+ & 1.00101 \\
\hline
 & 0.10011 \quad\quad 11000 \\
 & 1.00110 \quad\quad 1.10001
\end{array}$$

$[\frac{x}{y}]_补 = 1.10001$

$\frac{x}{y} = -0.01111$

$= -\frac{15}{32}$

知识点解析

- 知识点
  - 各除法算法的原理、计算公式
  - 各算法的优缺点、意义
  - 算法的加法、移位次数
  - 溢出判断
    - 1. 原码小数除法：第一位上商为1，则溢出
    - 2. 补码小数除法：第一位上商为1，两操作数符号位异或为0，溢出；第一位上商为0，两操作数符号位异或为1，溢出

hw6

课后习题：6.27 , 6.31

参考解答

6.27 假设阶码取 3 位，尾数取 6 位（均不包括符号位），计算下列各题。

(1) $[2^5 \times \frac{11}{16}] + [2^4 \times (-\frac{9}{16})]$。

(2) $[2^{-3} \times \frac{13}{16}] - [2^{-4} \times (-\frac{5}{8})]$。

(3) $[2^3 \times \frac{13}{16}] \times [2^4 \times (-\frac{9}{16})]$。

(4) $[2^6 \times (-\frac{11}{16})] \div [2^3 \times (-\frac{15}{16})]$。

(5) $[2^3 \times (-1)] \times [2^{-2} \times \frac{57}{64}]$。

(6) $[2^{-6} \times (-1)] \div [2^7 \times (-\frac{1}{2})]$。

(7) 3.3125 + 6.125。

(8) 14.75 − 2.4375。

6.27.(1) $[2^5 \times \frac{11}{16}] + [2^4 \times (-\frac{9}{16})]$

$[x]_{补} = 00,101;\ 00.1011$

$[y]_{补} = 00,100;\ 11.0111$

(1) 对阶 ✓

$j_x - j_y = 00,101 - 00,100 = 00,001$

$S_y \to 1,\ j_y + 1$

$[y]_{补}' = 00,101;\ 11.101110$

(2) 尾数求和 ✓

$00.101100 + 11.101110$

$= 00.011010$

(3) 左规1位 ✓

$00,100;\ 00.110100$

(4) 无需舍入 ✓

(5) 未溢出

结果 $2^4 \times \frac{13}{16}$ ✓

- （6）$[2^{-6}\times(-1)]\div[2^{7}\times(-\frac{1}{2})]$
- 可先将除数$2^7\times$（$-\frac{1}{2}$）尾码左移一位，1.100000规格化为1.000000，其阶码减一，避免除法结果溢出
- $[2^{-6}\times(-1)]\div[2^{7}\times(-\frac{1}{2})]$=$[2^{-6}\times(-1)]\div[2^{6}\times(-1)]$
- $[-6]_补+[-6]_补$

$$
\begin{array}{r}
= 11,010 \\
+ 11,010 \\
\hline
10,100
\end{array}
$$

- 下溢，按机器零处理

知识点解析

- 知识点
  - 浮点加减法分步骤记忆
  - 浮点乘除法溢出判断
  - 数值的浮点表示方法

补充：浮点乘除法运算规则及流程

- 在进行浮点运算之前，检查两个操作数（被乘数及乘数，被除数及除数）的尾数是否均为规格化表示；如果不是，则需进行规格化
- 其次，进行浮点运算
  - 如果发现被乘数或除数（即X寄存器中的数）的尾数为“-1”，则改为“-1/2”，阶码相应+1（将-1改为-1/2，是因为+1.0不能用补码表示）；注意，此时是在运算过程中，因此运算过程中的值（-1/2）不是规格化是允许的
  - 对于除法，需保证“被除数绝对值<除数绝对值<1”，否则应进行尾数调整（即，右移被除数且阶码++，如-1变为-1/2，-1/2变为-1/4），使得尾数除法不溢出
  - 阶码运算：若阶码采用补码表示，则乘法时阶码相加、除法时阶码相减
  - 进行尾数的零判断：乘法中若一个尾数为0则乘积为0、除法中若被除数为0则商为0、除法中若除数为0则商为无穷大，结束运算
  - 尾数运算：两个尾数相乘或相除
- 尾数运算结果规格化（乘法时需考虑舍入处理），同时根据阶码判断浮点运算结果是否溢出

参考解答

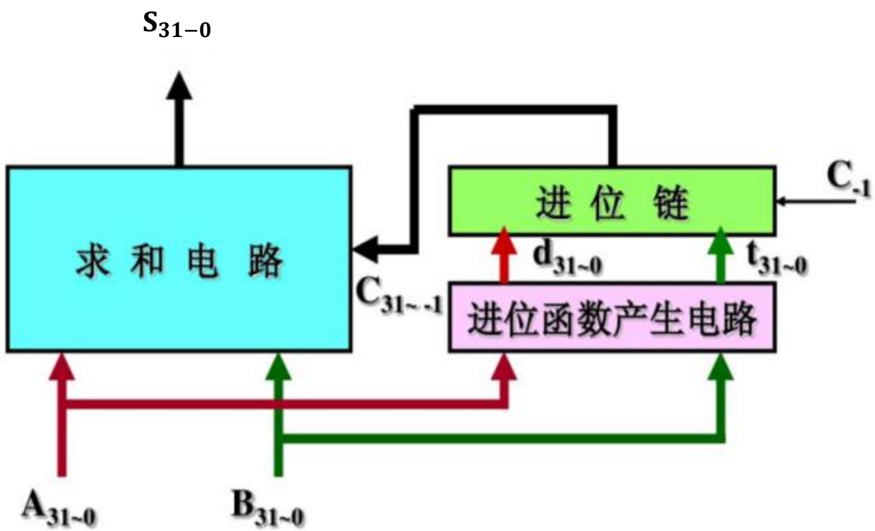
6.31 设机器字长为 32 位，用与非门和与或非门设计一个并行加法器（假设与非门的延迟时间为 30 μs，与或非门的延迟时间为 45 μs），要求完成 32 位加法时间不得超过 0.6 μs。画出进位链及加法器逻辑框图。

①串行进位：$T = 2T_{NAND} * 32 = 2 * 30 * 32 = 1920ns$ 不满足要求

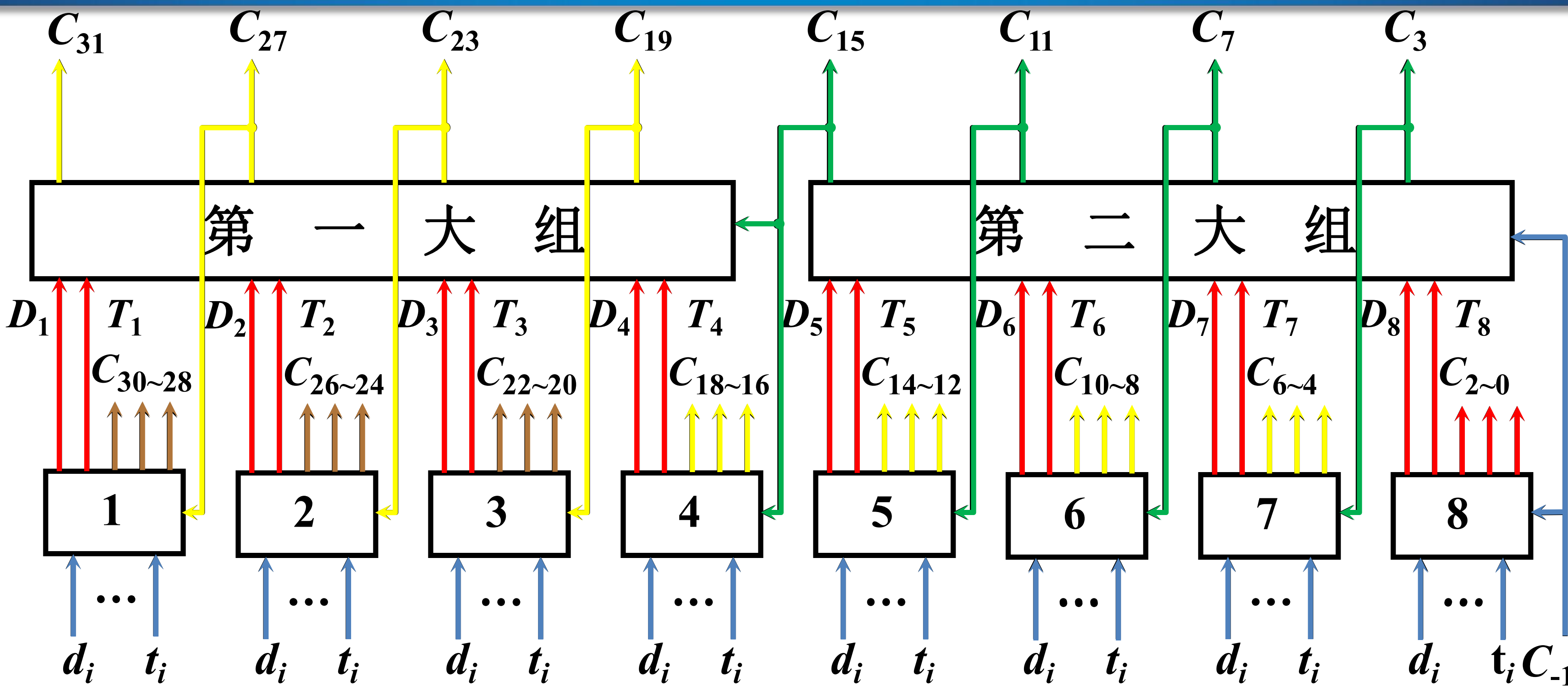
②单重分组：4位一组需 $T = (T_{AND-OR-NOT} + T_{NAND}) * 8$组 $= (45 + 30) * 8 = 600ns$ 而d_i，t_i 及S_i高位产生也需要时间，故不满足0.6μs要求

③双重分组： 4位一组需 $T = (T_{AND-OR-NOT} + T_{NAND}) * 4$级 $= (45 + 30) * 4 = 300ns$

- 双重分组跳跃进位链见教材P286图6.23
- 加法器逻辑框图：



n = 32 双重分组跳跃进位链



当 d_i, t_i 形成后 经 2.5 t_y 产生 C_2、C_1、C_0、$D_1 \sim D_8$、$T_1 \sim T_8$

5 t_y 产生 C_{15}、C_{11}、C_7、C_3

7.5 t_y 产生 $C_{18} \sim C_{16}$、$C_{14} \sim C_{12}$、$C_{10} \sim C_8$、$C_6 \sim C_4$
C_{31}、C_{27}、C_{23}、C_{19}

1 0 t_y 产生 $C_{30} \sim C_{28}$、$C_{26} \sim C_{24}$、$C_{22} \sim C_{20}$

知识点解析

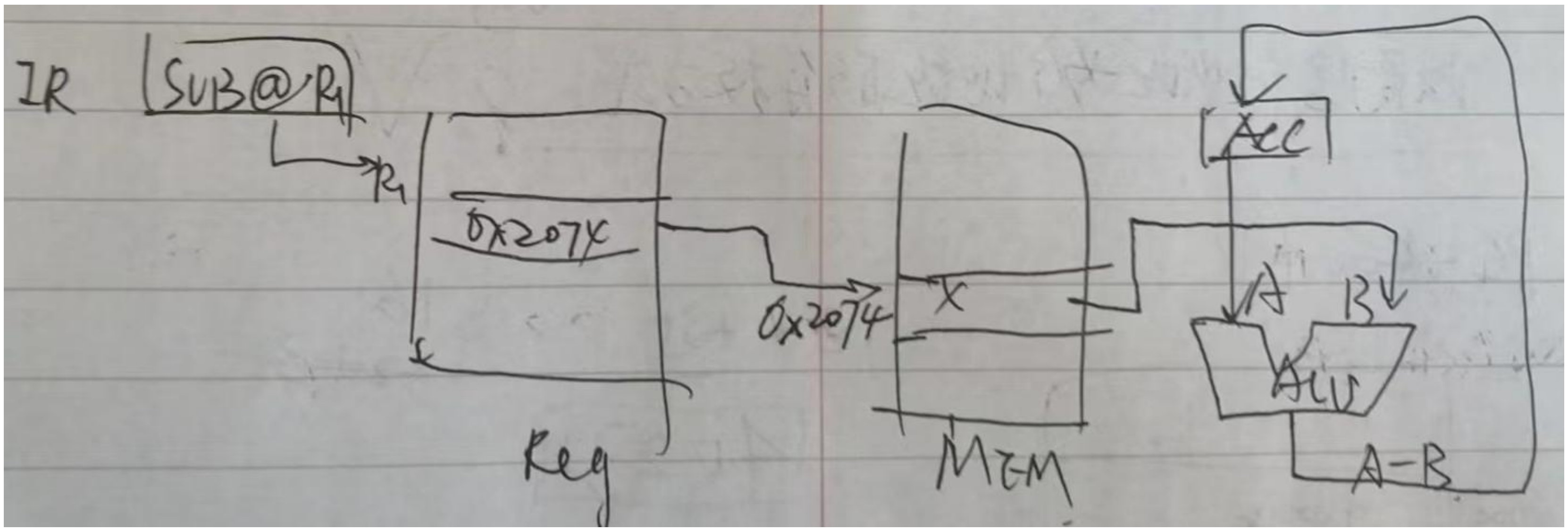
- 知识点
  - 加法器框图要素：输入、输出、进位
  - 各进位链框图
  - 进位链设计的意义、计算公式

hw7

课后习题：7.12，7.13，7.15，7.16

参考解答

7.12 画出"SUB @R1"指令对操作数的寻址及减法过程的流程图。设被减数和结果存于 ACC 中，@表示间接寻址，R1 寄存器的内容为 2074H。



知识点解析

- 知识点
  - 结合主机框图的各单元画寻址流程
  - 各寻址方法、优缺点都要掌握
  - 各寻址方法的地址计算，区别和联系

参考解答

7.15　一相对寻址的转移指令占3个字节，第一字节是操作码，第二、三字节为相对位移量，而且数据在存储器中采用以高字节地址为字地址的存放方式。假设PC当前值是4000H。试问当结果为0，执行“JZ * +35”和“JZ * -17”指令时，该指令的第二、第三字节的机器代码各为多少？

PC当前为4000H，取出三字节指令后，PC值变成4003H

JZ * +35, JZ * -17指令的相对位移量：

35 – 3 = 32, -17 – 3 = -20

+32对应的机器代码（补码）：00H 20H

-20对应的机器代码（补码）：FFH ECH

知识点解析

- 知识点
  - 大小端的定义
  - 地址指令格式
  - 各寻址方法的地址计算

参考解答

7.16 某机主存容量为4 M×16位，且存储字长等于指令字长，若该机指令系统可完成108种操作，操作码位数固定，且具有直接、间接、变址、基址、相对、立即等六种寻址方式，试回答以下问题。

(1) 画出一地址指令格式并指出各字段的作用。

(2) 该指令直接寻址的最大范围。

(3) 一次间接寻址和多次间接寻址的寻址范围。

(4) 立即数的范围(十进制表示)。

(5) 相对寻址的位移量(十进制表示)。

(6) 上述六种寻址方式的指令中哪一种执行时间最短，哪一种最长，为什么？哪一种便于程序浮动，哪一种最适合处理数组问题？

(7) 如何修改指令格式，使指令的寻址范围可扩大到4 M？

(8) 为使一条转移指令能转移到主存的任一位置，可采取什么措施？简要说明之。

(3)多次间址，因为存储字的首位用来标志间接寻址是否结束，因此寻址范围为2^{15}

(7)指令格式改为双字指令，地址字段16+6=22位，直接寻址范围扩大到4M

(8)能转移到任意主存地址，寻址范围需达4M，除了(7)方法，还可以采用(1)的一地址指令格式，并且配置22位的基址寄存器或22位的变址寄存器，使EA=(BR)+A或EA=(IX)+A，便可访问4M存储空间；还可以通过16位的基址寄存器左移6位(低位补0)再和形式地址A相加，也可达到同样效果

知识点解析

- 知识点
  - 各寻址方式寻址范围的计算
  - 各寻址方法的特点
  - 地址指令格式

Q & A ?