第一部分 C 语言程序设计

C 语言程序设计同步练习答案

第 1 章 绪论

一、单选题

1-5 DBCAC

二、判断题

1. 错　　2. 错

第 2 章　C 语言基本数据类型、运算符和表达式

一、单选题

1-5 ADADC　　6-10 ADBBA　　11-15 ADCAD　　16-20 DACBD

21-25 CADBB 26 -27 CA

二、填空题

1. 1　　2. 81　　3. 1　　4. 3　　5. 6

6. 1

7. 字母 下划线　　8. 数据类型 变量名表列

9. a=a+b ;　b=a-b ; a=a-b ;　　10. double

三、判断题

1. 错　2. 错　　3. 错　　4. 对　　5. 错　　6.错　　7.对

第 3 章 顺序结构程序设计

一、单选题

1-5 CDACD　　6-10 CACDA　　11-15 BDABC　　16-20 BBCAC

21-25 DDCBB　　26-28 CDC

三、阅读程序题

1. 3　3　　2. a=12345,b=-1.98e+002,c=　6.50

3. -2345,-12.30　　4. yes　　5. 8　8.　2f.　2f

第 4 章 选择结构

一、选择题

1-5 CDABC　　6-9 BCDD

二、阅读程序题

1. *&　　2. pass fail

三、程序填空题

1. 【1】a==b||a==c || b==c　【2】a==b && b==c

【3】a2+b2==c2||a2+c2==b2||b2+c2==a2

四、编程题

1. #include "stdio.h"

```
void main()
{
    float a;
    printf("请输入一个实数：");
```

```
    scanf("%f", &a);
    if(a>=0)
        a = a;
    else
        a = -a;
    printf("%f\n", a);
}
```

2.
```
#include "stdio.h"
void main()
{
    int a;
    printf("请输入一个整数：");
    scanf("%d", &a);
    if(a%2==0)
        printf("%d 是一个偶数\n", a);
    else
        printf("%d 是一个奇数\n", a);
}
```

3.
```
#include<stdio.h>
void main()
{
    int score,grade,temp;
   printf("Your scroe:");
   scanf("%d",&score);
   if(score>100||score<0)
{  printf("error\n");   }
   else
   {  temp=score/10;
      switch(temp)
    {
      case 10: case 9:grade='A';break;
      case 8:grade='B';break;
      case 7:grade='C';break;
      case 6: grade='D';break;
      default: grade='E';break;
    }
    printf("grade:%c\n",grade);
   }
}
```

第 5 章 循环结构

一、选择题

1-5 BBBCC　　6-10 DCCAC　　11-15 ABCCB　　16-20 BBCCA

21-25 BBACB　　26-30 DCACC 31 A

二、阅读程序题

1. i=10,j=23　　2. 28 70　　3. 2,0　　4. 8　　5. 3

6. 3, 1, -1,　　7. a=16 y=60　　8. i=6,k=4　　9. 1,-2

10. MNO JKL MNO GHI

三、填空题

1. 【1】f2+f1　　2. 【1】m/n　【2】n　【3】m%n

3. 【1】m=n　【2】m　【3】m=m/10

4. 【1】i<=4　【2】count++　5. 【1】m%i==0

6. 【1】(cx=getchar())　【2】front!=' '　【3】cx

7. 【1】printf(“\n”)

8.【1】t=sign*i　【2】sign=-sign

四、编程题

1. #include<stdio.h>

```
void main()
{int s=0,i;
 for(i=1;i<100;i=i+2)
  s+=i;
 printf(“%d”,s);}
```

2. #include<stdio.h>

```
void main()
{
    int n, i, sum;
    printf("请输入 n：");
    scanf("%d", &n);
    if (n%2==0)
        for(i=0, sum=0; i<=n/2; i++)
            sum = sum + 2*i;
    else
        for(i=0, sum=0; i<=n/2; i++)
            sum = sum + 2*i+1;
    printf("sum=%d\n", sum);
}
```

3. #include "stdio.h"

```
void main()
{
    int n, i, m;
    printf("请输入 n：");
    scanf("%d", &n);
    m = 0;
    while (n)
    {
        m = m*10 + n%10;
        n = n / 10;
    }
```

```
    printf("逆序数是%d\n", m);
}
4. #include "stdio.h"
void main()
{
    int n, f, i;
    printf("请输入 n: ");
    scanf("%d", &n);

    for (i=1, f=1; i<=n; i++)
    {
        f = f * i;
        printf("%d!=%d\n", i, f);
    }
}
5. #include "stdio.h"
void  main()
{
    int i,a,b,c;
    for(i=100;i<1000;i++)
    {
        c=i/100;
        b=i/10%10;
        a=i%10;
        if(a*a*a+b*b*b+c*c*c==i)
            printf("%d,%d",a*a*a+b*b*b+c*c*c,i);
    }
}
6. #include "stdio.h"
#define EPS 1e-5
void main()
{
    double e, item;
    int i;
    item = 1;
    e = 1;
    i = 1;
    do {
        item = item * i;
        e = e + 1.0/item;
        i++;
    }while (1.0/item>EPS);
    printf("e=%lf\n", e);
    printf("共累加了%d 项\n", i-1);
```

```
}
7. #include "stdio.h"
void main()
{
    int n, i, term, sum;
    printf("请输入正整数 n：");
    scanf("%d", &n);
    i = 0;
    term = 1;
    sum = 0;
    do  {
        i++;
        term = term * i;
        sum = sum + term;
    } while(sum<n);
    printf("m=%d\n", i-1);
}
8. #include "stdio.h"
void main()
{
    int chicks, rabbits;
    for (rabbits=0; rabbits<=35; rabbits++)
    {
        chicks = 35 - rabbits;
        if (2*chicks+4*rabbits==94)
            printf("chicks:%d\trabbits:%d\n", chicks, rabbits);
    }
}
9. #include<stdio.h>
void main( )
{
    int a,b,c;
    for(a=0;a<=20;a++)          //鸡翁最多可买 20 只
        for(b=0;b<=33;b++)      //鸡母最多可买 33 只
        {
            c=100-a-b;          //鸡雏的数目
            if(5*a+3*b+(float)c/3==100)  //判断百钱
                printf("鸡翁%d 只，鸡母%d 只，鸡雏%d 只\n",a,b,c);
        }
}
10. #include "stdio.h"
void main()
{
    double money, term;
```

```
    int i;
    for (i=1, term=1, money=1; i<=30; i++)
    {
        term *= 2;
        money += term;
    }
    printf("陌生人给百万富翁%d 万元\n", 10*30);
    printf("百万富翁给陌生人%lf 万元\n", money/100/10000);
}
11. #include<stdio.h>
void main()
{
 int i,score,grade,temp;
 for(i=0; i<100; i++)
 {
   printf("Your scroe:");
   scanf("%d",&score);
   if(score>100||score<0)
{  printf("error\n");   }
   else
   {  temp=score/10;
      switch(temp)
    {
      case 10: case 9:grade='A';break;
      case 8:grade='B';break;
      case 7:grade='C';break;
      case 6: grade='D';break;
      default: grade='E';break;
    }
    printf("grade:%c\n",grade);
   }
 }
}
```

第 6 章 数组

一、单选题

1-5 CBBAC　　6-10 DAACC　　11-15 ABCDD

16-20ADBCD　21-25 DBBAD　　26-30 BBBBD

31-35 DDCAB　　36 A

二、阅读程序题

1. 1 0 0 0 0　　2. AQM　　3. AzyD　　4. 9198　　5. 5 7 4 8 9 1

   0 1 0 0 0　　1 5 7 4 8 9

   0 0 1 0 0　　9 1 5 7 4 8

0 0 0 1 0　　8 9 1 5 7 4
0 0 0 0 1　　4 8 9 1 5 7
7 4 8 9 1 5

三、填空题

1.【1】x[i++]　　2. 【1】a[9-i]

3. 【1】float a[10],x　【2】i<10　【3】a[j]>a[j+1]　’

【4】a[j]=a[j+1]　【5】(i)%5==0

4. 【1】a　【2】a　【3】x[i]<ave

5. 【1】i==j　　【2】a[i][i]

6. 【1】n%base　　【2】n/base　　【3】j=i;j>=0;j—

7. 【1】j=i　　【2】k=i　　【3】a[j]=max,a[k]=min

8. 【1】j=i-1　　【2】a[i--]=a[j]　【3】a[j]=k

9. 【1】b[i][j]=a[i][j-1]　　【2】i=0;i<3;i++

10. 【1】strlen(t)　　【2】t[k]==c

11. 【1】i=0　　【2】i<10

12. 【1】b[j]!=’ \0’　　【2】b[j]=’ \0’

四、编程题

1. #include <stdio.h>

```
void main()
{
char b[16]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};
unsigned int c[64],d,i=0,n,base,k;
scanf("%u",&n);
for(k=1;k<=3;k++)
{
    if(k==1)  base=2;
    else if(k==2)   base=8;
         else base=16;
    do
    {
        c[i]=n%base;
        i++;
        n=n/base;
    }while(n!=0);
    for(--i;i>=0;i--)
    {
        d=c[i];
        printf("%c",b[d]);
    }
}
}
```

2. #include <stdio.h>

```
#define N 10
```

```
void main()
{
    int a[N][N], n;
    int i, j, sum=0;
    printf("Input n:");
    scanf ("%d", &n);
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            scanf("%d", &a[i][j]);
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            if (i==j || i+j==n-1)
                sum += a[i][j];
        printf("Sum = %d\n", sum);
}
```

第 7 章 函数

一、单选题

1-5 CACBA　　6-10 DABDA　　11-15 ADBAD

16-20 BABAD　　21-25 BCCDC　　26 A

二、阅读程序题

1. 2 , 6 , 42 , 3

2.　1! =1

2! =2

3! =6

4! =24

5! =120

3. 8,17　　4.　0 2 4 6 8 10 12 14 16 18　　5.　-125=-5*5*5

0 2 4 6 8 10 12 14 16 18

6.　i=5　　7. 程序功能为：找出所有水仙花数　　8. 6　　9. 6　　10

i=2

i=2

i=4

i=2

三、填空题

1. 【1】x　　2.【1】n==1　　【2】n*fac(n-1)　　【3】y=fac(n)

3. 【1】x2=mid-1　　【2】x1=mid+1

4. 【1】p++　　【2】a[i]=a[i+1]

5. 【1】float fun(float,float); 【2】x,y　　【3】y,z

6. 【1】i<=10　　【2】array[i-1]　　【3】return avgr

四、编程题

1. #include <stdio.h>

```
long Fact(int n)
{
```

```
    int i;
    long result = 1;
    for (i=2; i<=n; i++)
        result = result * i;
}
void main()
{
    int m;
    long ret;
    printf("Input m: ");
    scanf("%d", &m);
    ret = Fact(m);      //调用函数 Fact(m)，并将函数的返回值存入 ret
    printf("%d! = %ld\n", m, ret);
}
```

2.
```
#include "stdio.h"
long Fact(int n);
void main()
{
    int n;
    long result;
    printf("Input n: ");
    scanf("%d", &n);
    result = Fact(n);       //调用函数 Fact(n)
    if (result==-1)         //处理非法数据
        printf("n<0, data error!\n");
    else
        printf("%d! = %ld\n", n, result);
}
long Fact(int n)
{
    if (n<0)
        return -1;
    else if (n==0 || n==1)
        return 1;
    else
        return (n * Fact(n-1));
}
```

3.
```
#include "stdio.h"
long Fib(int n);
void main()
{
    int n, i, k;
    printf("Input n: ");
    scanf("%d", &n);
```

```
    for (i=1; i<=n; i++)
    {
        x = Fib(i);
        printf("Fib(%d) = %d\n", i, x);
    }
}
long Fib(int n)
{
    if (n==0)
        return 0;
    else if (n==1)
        return 1;
    else
        return (Fib(n-1)+Fib(n-2));
}
```

4.
```
#include <stdio.h>
int lcm(int m,int n);
void main()
{
int m,n;
do{
scanf(“%d%d”,&m,&n);
}while(n<=0||m<=0);
printf(“%d,%d 的最小公倍数是%d”,m,n,lcm(m,n);
}
int lcm(int m,int n)
{
int result;
for(result=1;;result++)
if(result%m==0&&result%n==0)
break;
return result;
}
```

5.
```
#include <stdio.h>
void swap(int *arr,int n)
{
int t,i;
for(i=0;i<n/2;i++)
{t=arr[i];arr[i]=arr[n-i-1];arr[n-i-1]=t;}
}

void main()
{
    int a[10]={1,3,5,7,9,11,13,15,17,19},i,n;
```

```
    scanf("%d",&n);
    for(i=0;i<n;i++)
        printf("%d ",a[i]);
    printf("\n");
    swap(a,n);
    for(i=0;i<n;i++)
        printf("%d ",a[i]);
}
```

第 8 章 指针

一、单选题

1-5 BBCCD　　6-10 BBDCB　　11-15 ABBDC　　16-20 DCCCB

21-25 DCAAA　　26-30 CCBDC

二、程序阅读题

1. aa　　2. 0987654321

三、填空题

1. 【1】void inv(int * ,int); 【2】i<10　【3】int *x 【4】j=x+n-1 【5】*i=*j

2. 【1】 pc, pb　　【2】 pa, pc　　【3】pa, pb

3. 【1】break　　【2】*s1-*s2

4. 【1】*temp++=*src++　　【2】dest

5. 【1】'\0'　　【2】i　　【3】;　　【4】k　　【5】count

四、编程题

1.
```
#include "stdio.h"
#define N 10
void Swap (int *x, int *y);
void Readingarr (int a[], int n);
void Printarr (int a[], int n);
void main()
{
    int a[N], b[N], i, n;
    printf("Input n");
    scanf("%d", &n);
    printf("输入数组 A， %d 个整数：", n);
    Readingarr(a, n);
    printf("输入数组 B， %d 个整数：", n);
    Readingarr(b, n);
    //交换两个数组的对应值
    for (i=0; i<n; i++)
        Swap(&a[i], &b[i]);
    Printarr(a, n);
    Printarr(b, n);
}
void Readingarr (int a[], int n)
{
```

```
    int i;
    for (i=0; i<n; i++)
        scanf("%d", &a[i]);
}
void Printarr (int a[], int n)
{
    int i;
    for (i=0; i<n; i++)
        printf("%5d", a[i]);
    printf("\n");
}
void Swap(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
```

2. int strcmp(char *p1,char *p2)
{
while(*p1++==*p2++);
if(*p1<*p2) return *p1-*p2;
else if(*p1==*p2) return 0;
     else return *p1-*p2;
}

第 9 章 编译预处理

一、单选题
1-3 AAB
二、判断题
1. 错　2. 错

第 10 章 结构体和共用体
一、单选题
1-5 BDCDD　　6-10 CCDCB
二、阅读程序题
1. A　　2. 1001,ZhangDa,1098.0
2
三、判断题
1. 错　　2. 错
四、编程题
1. #include <stdio.h>
struct date
{

```
    int   year;
    int   month;
    int   day;
};
struct student
{
    long  studentID;            /* 学号 */
    char  studentName[10];       /* 姓名 */
    char  studentSex;           /* 性别 */
    struct date  birthday;     /* 出生日期 */
    int score[4];                /* 4门课程的成绩 */
};
void main()
{
    int i, j, sum[30];
    struct student stu[30] = {{100310121,"王刚",'M',{1991,5,19},{72,83,90,82}},
    {100310122,"李小明",'M',{1992,8,20},{88,92,78,78}},
    {100310123,"王丽红",'F',{1991,9,19},{98,72,89,66}},
    {100310124,"陈莉莉",'F',{1992,3,22},{87,95,78,90}}
    };
    for (i=0; i<4; i++)
    {
        sum[i] = 0;
        for (j=0; j<4; j++)
        {
            sum[i] = sum[i] + stu[i].score[j];
        }
        printf("%10ld%8s%3c%6d/%02d/%02d%4d%4d%4d%4d%6.1f\n",
            stu[i].studentID,
            stu[i].studentName,
            stu[i].studentSex,
            stu[i].birthday.year,
            stu[i].birthday.month,
            stu[i].birthday.day,
            stu[i].score[0],
            stu[i].score[1],
            stu[i].score[2],
            stu[i].score[3],
            sum[i]/4.0);
    }
}
```

2.
```
#include <stdio.h>
#include <string.h>
#include <stdio.h>
```

```
#include <string.h>
#define  stu_amount 20  //学生人数
//定义学生结构体
struct student
{
    char name[8];//姓名
    char number[4];//学号
    float c_score;//C 语言成绩
};

void  main()
{
    int index;
    float score=0.0;//成绩汇总
    float avg_score;
    struct student stu[stu_amount];
    for(index =0; index <stu_amount; index++)
    {
       /*一个个录入成绩*/
        printf("请输入第%d 位学生的信息：\n",index);
        printf("姓名：");
        scanf("%s",stu[index].name);
        printf("学号：");
        scanf("%s",stu[index].number);
        printf("成绩：");
        scanf("%f",&stu[index].c_score);
    }
    printf("不及格的学生有：\n");
    printf("姓名\t 学号\t 成绩\n");
    for(index =0; index <stu_amount; index++)
    {
        score+=stu[index].c_score;
        if(stu[index].c_score<60)
        {
printf("%s\t%s\t%f\n",stu[index].name,stu[index].number,stu[index].c_score);
        }
    }
    avg_score=score/stu_amount;
    printf("平均成绩是：%f\n",avg_score);
    return 0;
}
3. #include "stdio.h"
struct staff
{
```

```
char name[10];float salary;int age;}s[5];
void main()
{
int i;
for(i=0;i<5;i++)
{
    printf("请输入：");
    printf("第%d 个职员姓名：",i+1);
    scanf("%s",s[i].name);
    printf("工资：");
    scanf("%f",&s[i].salary);
    s[i].salary*=1.3;
    printf("年龄：");
    scanf("%d",&s[i].age);
    s[i].age++;
}
printf("修改后：\n");
for(i=0;i<5;i++)
{
    printf("第%d 个职员姓名：",i+1);
    printf("%s  ",s[i].name);
    printf("工资：");
    printf("%f  ",s[i].salary);
    printf("年龄：");
    printf("%d\n",s[i].age);
}
}
```

4.
```
#include "stdio.h"
#define M 40
struct stu
{
int xh;
char xm[8];
float yw;
float sx;
float yy;
float pjcj;
};
void main()
{
struct stu xs[M],a;
int i,j,n;
scanf("%d",&n);   /*学生人数*/
for(i=0;i<n;i++)
```

```
scanf("%d%s%f%f%f",&xs[i].xh,xs[i].xm,&xs[i].yw,&xs[i].sx,&xs[i].yy);
for(i=0;i<n;i++)
xs[i].pjcj=(xs[i].yw+xs[i].sx+xs[i].yy)/3;
for(i=0;i<n-1;i++)
  for(j=0;j<=n-i;j++)
      if(xs[j].pjcj<xs[j+1].pjcj)
      {a=xs[j];xs[j]=xs[j+1];xs[j+1]=a;}
for(i=0;i<n;i++)
printf("%d %s %f %f %f %f\n",xs[i].xh,xs[i].xm,xs[i].yw,xs[i].sx,xs[i].yy,xs[i].p
jcj);
}
```

5.
```
#include <stdio.h>
struct aa
{
    char city[10];
    float spring;
    float summer;
    float autumn;
    float winter;
    float avgtem;
};
void main()
{
    struct aa x[10],t;
    int i,j;
    for(i=0;i<10;i++)

    scanf("%s%f%f%f%f",x[i].city,&x[i].spring,&x[i].summer,&x[i].autumn,&x[i].w
inter);
    for(i=0;i<10;i++)
        x[i].avgtem=(x[i].spring+x[i].summer+x[i].autumn+x[i].winter)/4;
    for(i=0;i<9;i++)
        for(j=0;j<9-i;j++)
        {
            if(x[j].avgtem>x[j+1].avgtem)
            {t=x[j];x[j]=x[j+1];x[j+1]=t;}
        }
    for(i=0;i<10;i++)
    printf("%s,%f,%f,%f,%f,%f\n",x[i].city,x[i].spring,x[i].summer,x[i].autumn,
x[i].winter,x[i].avgtem);
}
```

6.
```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
```

```
#include <string.h>
struct CARD
{
   char suit[10];
   char face[10];
};
struct CARD card[52];     /*顺序存放扑克牌*/
char *Suit[] = {"3","4","5","6"};
char *Face[] = {"A","2","3","4","5","6","7","8","9","X","j","Q","K"};
void main()
{
int count=0;
int num=0,k;
struct CARD t;
srand(time(NULL));
for (k=0; k<52; k++)
{
strcpy(card[k].suit, Suit[k/13]);
strcpy(card[k].face, Face[k%13]);
}
for (k=0; k<52; k++)   //输出顺序结果
{if(strcmp(card[k].suit,"3")==0) printf("%c",3);
 else if(strcmp(card[k].suit,"4")==0) printf("%c",4);
      else if(strcmp(card[k].suit,"5")==0) printf("%c",5);
           else if(strcmp(card[k].suit,"6")==0) printf("%c",6);
printf(" %s  ", card[k].face);
 if((k+1)%13==0) printf("\n");
}
printf("\n");
printf("\n");

while(count<52)     //随机发牌
{
k=rand()%52;
 t=card[count];
 card[count]=card[k];
 card[k]=t;
 count++;
 }

for (k=0; k<52; k++) /*输出发牌结果*/
{if(strcmp(card[k].suit,"3")==0) printf("%c",3);
 else if(strcmp(card[k].suit,"4")==0) printf("%c",4);
      else if(strcmp(card[k].suit,"5")==0) printf("%c",5);
```

```
            else if(strcmp(card[k].suit,"6")==0) printf("%c",6);
    printf(" %s  ", card[k].face);
 if((k+1)%13==0) printf("\n");
}
}
```

第二部分　微机原理与接口含汇编语言

第 1 章 微型计算机基础

1.1　微型计算机的概述

一、判断题

1. √

2. ×

3. √

4. ×

二、名词解释

1. 位（Bit）：位是指计算机中使用的二进制数的一位，它是存储信息中的最小单位。只有“0” 和“1”两种状态。

2. 字节（Byte）:计算机存储数据时，通常把 8 位二进制数作为一个存储单元，一个存储单元也叫一个字节。字节的长度固定，它是存储器存取信息的最小单位。

3. 字（Word）：字是计算机中处理和传送信息的最基本单位。它通常与寄存器、运算器、传输线的宽度一致。

4.字长:一个字所包含二进制数的长度称为字长。实际上字长所表示的是 CPU 并行处理的 最大位数。如 16 位机字长为 16 位，占 2 个字节。32 位机的字长为 32 位，占 4 个字节。

5. 存储容量：是指 CPU 构成的系统所能访问的存储单元的字节数。

6. 指令：计算机能识别和执行的基本操作命令。有两种方式：机器码和助记符。

7. 指令系统：计算机所能执行的全部指令的集合，称为该计算机的指令系统。

8. 程序：为完成某一任务所作的指令（或语句）的有序集合称为程序。

9. 运算速度：计算机完成一个具体任务所用的时间就是完成该任务的时间指标，计算机的速度越高，所用的时间越短。

1.2 微型计算机的组成

一、名词解释

1. 中央处理器（CPU）：它由运算器、控制器和寄存器 3 大部分组成。

2. 存储器：主要是存储代码和运算数据的。

3. 接口：是连接主机和外设的桥梁。

4. 输入/输出（I/O）设备：能把外部信息传送到计算机的设备叫输入设备。将计算机处理完 的结果转换成人和设备都能识别的和接收的信息的设备叫输出设备。

5. 总线：连接各硬件部分的线路。一是用来传递数据信息的叫数据总线简称 DB；二是用来传递地址信息的简称 AB；三是专门用来传递控制信息简称 CB。

二、简答题

1. 答：微机主要有存储器、I/O 设备和 I/O 接口、CPU、系统总线、操作系统和应用软件组成，各部分功能如下：

CPU：统一协调和控制系统中的各个部件

系统总线：传送信息

存储器：存放程序和数据

I/O 设备：实现微机的输入输出功能

I/O 接口：I/O 设备与 CPU 的桥梁

操作系统：管理系统所有的软硬件资源

2．答：微型计算机的基本工作过程是执行程序的过程，也就是 CPU 自动从程序存放的第 1 个存储单元起，逐步取出指令、分析指令，并根据指令规定的操作类型和操作对象，执行指令规定的相关操作。如此重复，周而复始，直至执行完程序的所有指令，从而实现程序的基本功能。

3．答：微型计算机系统的硬件主要由运算器、控制器、存储器、输入设备和输出设备组成。“存储程序控制”的概念可简要地概括为以下几点：

① 计算机（指硬件）应由运算器、存储器、控制器和输入/输出设备五大基本部件组成。

② 在计算机内部采用二进制来表示程序和数据。

③ 将编好的程序和原始数据事先存入存储器中，然后再启动计算机工作，使计算机在不需要人工干预的情况下，自动、高速的从存储器中取出指令加以执行，这就是存储程序的基本含义。

④ 五大部件以运算器为中心进行组织。

4．答：采用二进制形式表示数据和指令。指令由操作码和地址码组成。

将程序和数据存放在存储器中，计算机在工作时从存储器取出指令加以执行，自动完成计算任务。这就是“存储程序”和“程序控制”（简称存储程序控制）的概念。

指令的执行是顺序的，即一般按照指令在存储器中存放的顺序执行，程序分支由转移指令实现。 计算机由存储器、运算器、控制器、输入设备和输出设备五大基本部件组成，并规定了 5 部分的基本功能。

5．答：将运算器与控制器集成在一起，称为微处理器。微处理器是微处理器的核心。微型计算机是由微处理器、存储器、输入/输出接口电路和系统总线构成的裸机系统。微型计算机系统是以微型计算机为主机，配上系统软件和外设之后而构成的计算机系统。三者之间是有很大不同的，微处理器是微型计算机的一个组成部分，而微型计算机又是微型计算机系统的一个组成部分。

1.3 计算机中数的表示及运算

一、选择题

1．D　　2．C　　3．C　　4．B　　5．C　　6．C

二、填空题

1．1000 1100、1111 0011、1111 0100

2．39.A、01010111.011000100101

3．25.5、00100101.0101B

4．－128~+127

5．01111011　　10000011

6．11101100

7．11100111、-103

8．阶码、尾数

9．128、8

10．线性

三、计算题

1．（1）57.625D ；　39.AH

（2）50.8125D ；　32.DH

(3) 11.84375D ; B.D8H

(4) 45.4375D ; 2D.7H

2. (1) 111 1011.0010 0001B ; 123.12890625D ; 123.12890625 BCD

(2) 1 0010 0111.0001 11B ; 295.06640625D ; 295.06640625 BCD

(3) 101 1010 0001.0100 0001B ; 1697.25390625D ; 1697.25390625 BCD

(4) 10 1101 1111 0011.01B ; 11763.25D ; 11763.25 BCD

3. (1) [96]原=0110 0000 ; [96]反=0110 0000 ; [96]补=0110 0000

(2) [31]原=0001 1111 ; [31]反=0001 1111 ; [31]补=0001 1111

(3) [-42]原=1010 1010 ; [-42]反=1101 0101 ; [-42]补=1101 0110

(4) [-115]原=1111 0011 ; [-115]反=1000 1100 ; [-115]补=1000 1101

4. (1) -110D

(2) -115D

(3) -78D

(4) +19494D

1.4 微型计算机组成电路

一、选择题

1.C 2.C 3.C 4.D 5.C 6.D 7. C

二、填空题

1.多个部件之间公用的、信息交换的、芯片、插件或系统之间的

2.内部总线、系统总线和外部总线、系统总线外部总线

3.可同时传送的二进制数据的位数、在单位时间内总线上可传送的数据总量

4.高速图形接口局部总线标准_、高速视频或高品质画面的显示

5.支持即插即用的新型串行、使用方便、速度加快、连接灵活、独立供电、支持多媒体

6.新型的高速串行、超过100Kbit/s的硬盘和视频设备

三、简答题

1. 答：微机主板常用总线有系统总线、I/O总线、ISA总线、IPCI总线、AGP总线、IEEE1394总线、USB总线等类型。

2. 答：1)具备即插即用特性，为USB接口设计的驱动程序和应用程序可自动启动、成本低，节省空间，为开放性的不具备专利版权的理想工业标准。

2)可动态连接和重新配置外设，支持热插拔功能；

3)允许多台设备同时工作；

4)可以向USB总线上的设备供电，总线上的设备可以自备电源；

5)通讯协议支持等时数据传输和异步消息传输的混合模式；

6)支持实时语音、音频、和视频数据传输。

3. 答：微型计算机总线的主要职能是负责计算机各模块间的数据传输，对总线性能的衡量也是围绕这一性能而进行的。性能中最重要的是数据传输率，另外，可操作性、兼容性和性能价格比也是很重要的技术特征。具体来说，总线的主要性能指标有以下几项：

(1)总线宽度：以位数表示。

(2)标准传输率Mb/s：是总线工作频率与总线宽度的字节数之积。

(3)时钟同步/异步：总线中与时钟同步工作的称为同步总线；与时钟不同步工作的称为异步总线。这取决于数据传输时源模块与目标模块间的协议约定。(4)信号线数：这是地址总线、数据总线和控制总线线数的总和。信号线数和系统的复杂程度成正比关系。

(5)负载能力：以系统中可以连接的扩展电路板数表示。

(6)总线控制方法：包括突发传输、并发工作、自动配置、仲裁方式、逻辑方式、中断方式

等项内容。
(7)扩展板尺寸：这项指标对电路板生产厂家很重要。
(8)其他指标：电源是 5V 还是 3V，能否扩展 64 位宽度等。
任何系统的研制和外围模块的开发，都必须服从其采用的总线规范。
4.答：PCI 总线的优点：
（1）高性能，低成本；
（2）通用性强，使用方便；
（3）PCI 总线与处理器无关，具有 32 位和 64 位数据总线，采用集中式总线仲裁、支持多处理器系统，通过桥（Bridge）电路兼容 ISA/EISA 总线，具有即插即用的自动配置能力等一系列优势；
（4）可靠性高、数据完整性好。

第 2 章 十六位微处理器

2.1 8088/8086 微处理器

一、选择题
1. D 2.B 3. A 4. B 5.B 6. C 7.B 8. C 9. D 10. A
11. B 12. D 13. A 14. A 15. D
二、填空题
1. EU、BIU、16、20
2. CS、IP
3. 10230H、1000H、0230H
4. 9、6、3
5.分段、物理、相对段基地址的偏移量、程序
6. 16、16、20、1M
三、判断题
1.×
2.√
3.×
四、简述题
1. 答：按功能可分成两大部分：执行单元（EU）和总线接口单元（BIU）。
（1）执行单元(EU)：由 8 个通道寄存器，、1 个标志寄存器、算术逻辑运算单元 ALU 及 EU 控制单元组成。EU 从 BIU 指令队列寄存器中获得指令和待处理数据进行操作。将指令代码译码后，发出相应的控制信息，将数据在 ALU 中进行运行，运算结果的特征保留在标志寄存器 FLAG 中。
（2）总线接口单元（BIU)：总线借口单元 BIU 包括 4 个段寄存器、1 个指令指针寄存器、1 个内部寄存器、1 个先入先出的 6 个字节（8088 是 4 个字节）的指令队列、总线控制逻辑电路及 20 条地址线。当 EU 从指令队列中取走指令，指令队列出现空字节时，BIU 即从内存中去除后续的指令代码放入队列中。当 EU 需要数据时，BIU 根据 EU 给出的地址，从指定的内存单元或外设中取出数据提供给 EU。运算结束后，将运算结果送入指定的内存单元或外设。如果指令队列的所有字节全空，EU 停止执行。直到指令队列中有指令，并把指令传到 EU 单元，EU 开始操作。一般情况下，程序是顺序执行的。当遇到跳转指令时，BIU 就使指令队列复位，从新 地址中取出指令并立即送给 EU 去执行。
2. 答：作用是：在执行指令的同时从内存中取了一条指令或下几条指令，取来的指令放在指令队列中这样它就不需要象以往的计算机那样让 CPU 轮番进行取指和执行的工作，从而提

高 CPU 的利用率。
3．答：8086 的存储器空间最大可以为 220（1MB）；8086 计算机引入了分段管理机制，当 CPU 寻址某个存储单元时，先将段寄存器内的内容左移 4 位，然后加上指令中提供的 16 位偏移地址形成 20 位物理地址，即在 8086 系统中，物理地址=段地址×10H＋偏移地址。
4． 答：指令的物理地址为 21F00H；CS 值和 IP 值不是唯一的，例如：CS=2100H，IP=0F00H。
5． 答：偏移地址为 54100H。（物理地址=段地址×16+偏移地址）
6．答：物理地址：在处理器地址总线上输出的地址称为物理地址。每个存储单元有一个唯一的物理地址。 逻辑地址：在处理器内部、程序员编程时采用逻辑地址，采用“段地址：偏移地址“形式。某个存储单元可以有多个逻辑地址，即处于不同起点的逻辑段中，但其物理地址是唯一的。
逻辑地址转换成物理地址：逻辑地址由处理器在输出之前转换为物理地址。将逻辑地址中的段地址左移二进制 4 位（对应 16 进制是一位，即乘以 16），加上偏移地址就得到 20 位物理地址。
① FFFFH:0＝FFFF0H
② 40H:17H＝00417H
③ 2000H:4500H＝24500H
④ B821H:4567H＝BC777H
7．答：状态标志位有 6 个：ZF、SF、CF、OF、AF、PF。其意思是用来反映指令执行的特征，通常是由 CPU 根据指令执行结果自动设置的；控制标志位有 3 个：DF、IF、TF。它是由程序通过执行特定的指令来设置的，以控制指令的操作方式。
8．答：执行部件有 8 个 16 位寄存器，AX、BX、CX、DX、SP、BP、DI、SI，AX、BX、CX、DX 一般作为通用数据寄存器。SP 为堆栈指针存器，BP、DI、SI 在间接寻址时作为地址寄存器或变址寄存器。总线接口部件设有段寄存器 CS、DS、SS、ES 和指令指针寄存器 IP。段寄存器存放段地址，与偏移地址共同形成存储器的物理地址。IP 的内容为下一条将要执行指令的偏移地址，与 CS 共同形成下一条指令的物理地址。
9．答：① 由于 8088 只能传输 8 位数据，所以 8088 只有 8 个地址/数据复用引脚；而 8086 是按 16 位传输数据的，所以有 16 个地址/数据复用引脚；
②8086 和 8088 的控制线引脚定义中第 28 和 34 腿也不一样，在最小模式时，8088 和 8086 的第 28 引脚（M/IO）的控制信号高低电平相反，而 8086 的第 34 腿为 BHE/S7，BHE 用来区分是传送字节、还是字，8088 的第 34 腿为 SS0，用来指出状态信息，不能复用。

2.2 8088/8086CPU 引脚

一、选择题
1．B 2．B 3．B 4．A 5．C
二、填空题
1．Ready、T3
2．INTR、NMI、屏蔽
3．地、多处理器和单处理器系统
4．CPU 与其他部件之间传送数据、地址和控制信息、数据、地址、控制
5． 、A0、A19～A1
6． I/O 读
7． 0，0，1，1
8．0，1
9．1、0 、0
10．0FFFFH、0、0、0

三、简答题

1.答：8086 执行了一个总线周期是指完成一次读或者写的操作，基本总线周期由 T1~T4 的 4 个时钟周期组成，也是典型的读存储器总线周期。在一个典型的读存储器总线周期中，地址信号、ALE 信号在 T1 时刻有效， 信号、数据信号在 T2~T3 有效。

2．答：数据与地址引脚

3. 答：INTR 是可屏蔽请求信号， 中断响应信号，NMI 是不可屏蔽中断请求信号，ALE 是地址锁存允许信号，HOLD 总线请求信号，HLDA 总线请求响应信号。

4. 答: RESET：复位输入信号，高电平有效。该引脚有效时，将迫使处理器回到其初始状态；转为无效时，CPU 重新开始工作。

HOLD：总线请求，是一个高电平有效的输入信号。该引脚有效时，表示其他总线主控设备向处理器申请使用原来由处理器控制的总线。

NMI：不可屏蔽中断请求，是一个利用上升沿有效的输入信号。该引脚信号有效时，表示外界向 CPU 申请不可屏蔽中断。

INTR：可屏蔽中断请求，是一个高电平有效的输入信号。该引脚信号有效时，表示中断请求设备向处理器申请可屏蔽中断。

5．答：考虑到芯片成本，8086/8088 采用 40 条引线的封装结构。40 条引线引出 8086/8088 的所有信号是不够用的，采用地址/数据线复用引线方法可以解决这一矛盾，从逻辑角度，地址与数据信号不会同时出现，二者可以分时复用同一组引线。

2.3 8088/8086CPU 两种工作模式的配置

一、选择题

1. C　2. C　3. B　4. A　5.D

二、简答题

1. 答：8086 微处理器有最大和最小工作模式。在最小模式下：8086 CPU 直接产生全部总线控制信号（DT/R，DEN，ALE，M/IO）和命令输出信号（RD，WR，INTA）并提出请求访问总线的逻辑信号 HOLD，HLDA。在最大工作模式下，必须配置 8288 总线控制器，并且根据 8086 提供的状态信号 S2，S1，S0，输出读写控制命令，可以提供灵活多变的系统配置，以实现最佳的系统性能。

2．答：8284A 是一个专用的时钟发生器，产生 4.77MHz 的标准时钟信号 CLK。此时钟信号作为系统时钟，并经 CLK 引脚直接送到 8086，作为微处理器的时钟信号。同时 8284A 还对复位和就绪信号实现内部的时钟同步，然后再输出，实施对 8086 的控制。所以，8086/8088 系统用的时钟发生器产生恒定的时钟信号 CLK，复位信号 RESET，准备就绪信号 READY。

2.4 8088/8086CPU 主要操作

一、选择题

1. C　2. B　3. B

二、判断题

1.×

2.×

3. √

三、简答题

1．答：时钟周期是指 CPU 基本时间计量单位，常用一个 T 状态表示，总线周期是指一次总线操作时间。在 T3 之后加入等待周期。

2．答：当 8088 进行读写存储器或 I/O 接口时，如果存储器或 I/O 接口无法满足 CPU 的读写时序（来不及提供或读取数据时），需要 CPU 插入等待状态 Tw。在读写总线周期的 T3 和 T4 之间插入 Tw。

3．答：8086/8088 通过利用 ALE 信号的是否有效来解决地址线和数据线的复用问题。ALE 作为最小模式的地址锁存允许信号输出端，在任何总线周期的 T1 状态，ALE 输出有效电平，以表示当前在地址/数据复用总线上输出的是地址信息。

2.5 8088/8086 系统的中断操作

一、选择题

1. D　2. C　3. C　4. B　5. C　6. A　7. C　8. C

二、填空题

1. Iret

2. 1K、7856H:3412H

1. 中断系统

2. 引起中断的设备或事件、内部中断和外部中断

3. 除法出错、运算溢出和程序调试中设置断点

4. 256、0~255、中断、4、中断入口段地址、中断入口偏移地址

5. 中断服务程序的入口地址中断向量表

三、简答题

1．答：有 8 级；按照产生中断的方法可分为硬件中断和软件中断。

2．答：中断就是 CPU 在执行当前程序时由于内外部事件引起 CPU 暂时停止当前正在执行的程序而转向执行请求 CPU 暂时停止的内外部事件的服务程序，该程序处理完后又返回继续执行被停止的程序；中断向量是中断处理子程序的入口地址；地址范围是 00000H-003FFH。

3．答：中断向量表的功能是当中断源发出中断请求时，即可查找该表，找出其中断向量，就可转入相应的中断服务子程序。1AH 在中断向量表的位置是 1AH×4=68H 在中断向量表 0000H：0068H 处；20H 在中断向量表的位置是 80H 在中断向量表 0000H：0080H 处。

4．答：3 种，软件查询确定优先级，硬件优先级排队电路确定优先级，中断屏蔽接口电路。

5．答：CPU 响应可屏蔽中断的条件是：

（1）CPU 必须处于开中断状态 IF=1

（2）CPU 现行指令执行结束

（3）没有其他优先级高的中断请求。（没有内部中断，没有非屏蔽中断，没有总线请求。

6．答：中断请求：外设通过硬件信号的形式、向处理器引脚发送有效请求信号。

中断响应：在满足一定条件时，处理器进入中断响应总线周期。

关中断：处理器在响应中断后会自动关闭中断。

断点保护：处理器在响应中断后将自动保护断点地址。

中断源识别：处理器识别出当前究竟是哪个中断源提出了请求，并明确与之相应的中断服务程序所在主存位置。

现场保护：对处理器执行程序有影响的工作环境（主要是寄存器）进行保护。

中断服务：处理器执行相应的中断服务程序，进行数据传送等处理工作。

恢复现场：完成中断服务后，恢复处理器原来的工作环境。

开中断：处理器允许新的可屏蔽中断。

中断返回：处理器执行中断返回指令，程序返回断点继续执行原来的程序。

7．答：在中断响应过程中，CPU 向 8259A 的 INTR 引脚发 2 个负脉冲。作用：第一个负脉冲通知 8259A ，CPU 允许中断请求，要求送中断类型；第二个负脉冲，8259 传输中断类型码。

第 3 章 存储器系统

3.1 存储器的概述

一、选择题

1. D　2. D　3. C　4. C　5. C

二、填空题

1. 二进制信息总量、二进制信息、越强

2. 通过指令可随机地对存储单元进行访问、静态 RAM、动态 RAM、动态 RAM

3. 高速小容量、CPU、主存、CPU 正在使用的指令和数据、提高 CPU 访问存储器的存取速度，减少处理器的等待时间

三、判断题

1. （ × ）

2. （ √ ）

3. （ × ）

4. （ × ）

四、简答题

1. 答：1) 存储容量。存储器可以存储的二进制信息总量称为存储容量。存储容量有两种表示方法：

⑴ 位表示方法。以存储器中的存储地址总数与存储字位数的乘积表示。如 1K×4 位，表示该芯片有 1K 个单元（1K＝1024），每个存储单元的长度为 4 个二进制位。

⑵ 字节表示方法。以存储器中的单元总数表示（一个存储单元由 8 个二进制位组成，称为一个字节，用 B 表示）。如 128B，表示该芯片有 128 个单元。

2）存储速度。存储器的存储速度可以用两个时间参数表示，一个是“存取时间”，定义为从启动一次存储器操作到完成该操作所经历的时间；另一个是“存储周期”，定义为启动两次独立的存储器操作之间所需的最小时间间隔。

3) 可靠性。存储器的可靠性用平均故障间隔时间 MTBF 来衡量。MTBF 越长，可靠性越高。

4) 性能/价格比。这是一个综合性指标，性能主要包括上述三项指标：存储容量、存储速度和可靠性，对不同用途的存储器有不同的要求。

2. 答：RAM 有两种，（1）SRAM(静态 RAM)，它采用触发器电路构成一个二进制位信息的存储单元，这种触发器一般由 6 个晶体管组成，它读出采用单边读出的原理，写入采用双边写入原理；（2） DRAM（动态 RAM），它集成度高，内部存储单元按矩阵形式排列成存储体，通常采用行，列地址复合选择寻址法。

ROM 有 5 种，固定掩摸编程 ROM、可编程 PROM、紫外光擦除可编程 EPROM、电可檫除的可编程 EEPROM 和闪速存储器。

3.2 8086/8088 系统存储器的组织

一、填空题

1. D7~D0

2. ，IO/

3. 7FH

4. 奇、偶

5. 1、2

二、判断题并改正

1. ×，每个段小于等于为 64KB。

2. √。

3. √。

4. ×，内存直接与 CPU 进行程序和数据的交换。

5. √。

6. ×，可以通过紫外线擦除写入数据。
7. √。
8. ×，微处理器进行读操作，就是把数据从主存或外设读到微处理器。
9. ×，EPROM 只能通过紫外线向内部写入数据。
10. √
三、简答题
1．存储示意图：
地址　　存储空间
00130H　0DAH
00131H　31H

00134H　7FH
00135H　5EH
2. 答：30022H 字节单元的内容为 ABH；30024H 字节单元的内容为 EFH。30021H 字单元的内容为 AB34H；30022H 字单元的内容为 CDABH。

3. 答：3017H:000AH、3015H:002AH 和 3010H:007AH 的存储单元的物理地址都是 3017AH。
4．答：8086 是一个 16 位的结构，采用分段管理办法可形成超过 16 位的存储器物理地址，扩大对存储器的寻址范围（1MB，20 位地址）。若不用分段方法，16 位地址只能寻址 64KB 空间。
5．答:8086 微处理器 CPU 中寄存器都是 16 位，16 位的地址只能访问大小为 64KB 以内的内存。8086 系统的物理地址由 20 根地址线形成，怎样用 16 位数据处理能力实现 20 位地址的寻址呢？要做到对 20 位地址空间进行访问，就需要两部分地址，在 8086 系统中，就是由段地址和偏移地址组成的。而这两个地址都是 16 位，将这两个地址采用相加的方式组成 20 位地址去访问存储器。
在 8086 系统的地址形成中，当段地址确定后，该段的寻址范围就已经确定，其容量不大于 64KB。同时，通过修改段寄存器的内容，可达到逻辑段在整个 1MB 空间中浮动。各个逻辑段之间可以紧密相连，可以中间有间隔，也可以相互重叠。
采用段基址和偏移地址方式组成物理地址的优点是：满足对 8086 系统的 1MB 存储空间的访问，同时在大部分指令中只要提供 16 位的偏移地址即可。

3.3 存储芯片与 CPU 的连接

一、选择题
1．B　　2．A　　3．A
二、填空题
1．16、11、3
2．15、16、8
三、简答题
1．答：①全译码方式：存储器芯片中的每一个存储单元对应一个唯一的地址。译码需要的器件多；②部分译码方式：存储器芯片中的一个存储单元有多个地址。译码简单；线选：存储器芯片中的一个存储单元有多个地址。地址有可能不连续。不需要译码。
2．答：（1）位扩展：当存储器的容量要求与芯片的容量相同，但位数不同，就需要进行位上扩展。
（2）字扩展：当存储器的位数与芯片的相同，但是容量不足时，就需要在字上扩展。

（3）字位同时扩展：是指在内存容量和数据位长宽两个方向上同时扩展。
当需要组成的内存容量为M×N时，若已有芯片为m×n，所需芯片数＝（M／m）×（N／n）。
3．答：需要16片芯片；其中A11-A0参与片内寻址；A12做芯片组的片选信号。
4．答：512K×8需要19条地址线，8条数据线。1 K × 4需要10条地址线，4条数据线。16K × 8需要14条地址线，8条数据线。64K × 1需要14条地址线，1条数据线。
5．答：最后一个单元的地址是：4800H+213-1
四、应用题
1．答：第一组的基本地址是:0000H-03FFH
第二组的基本地址是:4000H-43FFH
第三组的基本地址是:8000H-83FFH
第四组的基本地址是:0C000H-0C3FFH
每一组都有重叠区,每组重叠区15个
2．解：⑴（1M×16）/（128K×16）=8
若每个模块板为128K×16位，共需8个模块板；
⑵（128K×16）/（16KB）=16
每个模块板内共有16个SRAM芯片；
⑶ 主存共需8×16=128个RAM芯片；将CPU发出地址的高3位A19、A18、A17经3：8线译码器译码后选择各模块板。
3．答：（1）EPROM单片容量为4K×8位，SRAM单片容量为2K×8位。
（2）74LS138的作用是译码。
（3）ERROM2的地址范围为：F9000H～F9FFFH。
解释：A19、A18、A17、A16、A15 均为1；由于ERROM2由Y1连接，译码器输入端A14、A13、A12为001，A11-A0为片内地址，所以20位地址是：
11111 001 000000000000----11111 001 111111111111
4．答：A19 A18 A17 A16 A15 A14 A13 A12 - A0
0 …… 0
第一组范围： 0 0 1 1 1 1 0
1 …… 1
即为：3C000H—3EFFFH
A19 A18 A17 A16 A15 A14 A13 A12 - A0
0 …… 0
第二组范围： 0 0 1 1 0 0 1
1 …… 1
即为：32000H—33FFFH
5．答：分配给32K×8ROM芯片的地址为：F8000H－FFFFFH；分配给8K×8ROM的地址为下述4组地址之一：
E0000H~E1FFFH E2000H~E3FFFH
E4000H~E5FFFH E6000H~E7FFFH
6．答 ：（1）8K×8bit
（2）因为有两根地址线没有用到，所以有4种地址范围：
DA000H~DBFFFH； DE000H~DFFFFH； FA000H~FBFFFH； FE000H~FFFFFH。
7．参考答案：连接图：

2.16KB

3.所用 2 片 RAM 的地址范围为：8000H~8FFFH 和 C000H~FFFFH

4. ① 确定读写信号片选是否正常；

② 地址线和数据线是否正常；

③ 用示波器检查 RAM 各点是否正常。

第 4 章 8088/8086 指令系统

4.1 8088/8086 指令系统的寻址方式

一、简答题

1. 答：数据操作数的寻址方式有七种，分别为：立即寻址，寄存器寻址，直接寻址，寄存器间接寻址，寄存器相对基址变址和相对基址变址寻址。其中寄存器寻址的指令执行速度最快。

2.（1）答：寻址方式为直接寻址；PA＝60064H

（2）答：寻址方式为直接寻址； PA＝60005H

（3）答：寻址方式为寄存器间接寻址；PA＝60100H

（4）答：寻址方式为寄存器间接寻址；PA＝60300H

（5）答：寻址方式为寄存器间接寻址；PA＝50400H

（6）答：寻址方式为寄存器间接寻址；PA＝61200H

（7）答：寻址方式为寄存器相对寻址；PA＝61410H

（8）答：寻址方式为寄存器相对寻址；PA＝60305H

（9）答：寻址方式为基址变址寻址；PA＝60400H

（10）答：寻址方式为相对基址变址寻址；PA=61505H

3.（1）答：有效地址为 EA＝C237H

（2）答：有效地址为 EA＝637DH

（3）答：有效地址为 EA＝125B4H

（4）答：有效地址为 EA＝8E18H

（5）答：有效地址为 EA＝1504FH

4.答：

		源操作数	目的操作数
(1)	MOV AX,100H	立即数	寄存器
(2)	MOV CX,AX	寄存器	寄存器
(3)	ADD [SI],1000	立即数	寄存器间接
(4)	SUB BX,[SI+100]	变址	寄存器
(5)	MOV [BX+300],AX	寄存器	变址
(6)	AND BP,[DI]	寄存器间接	寄存器

4.2 汇编指令

一、选择题

1. A 2. D 3. C 4. D 5. C 6. A 7. B 8. C 9. C 10. B

11.B 12. A 13.A 14.C 15.C

二、简答题

1. 答：（1）立即数不能传送到 DS

（2）栈操作，操作数类型必须为字类型

（3）SP 寄存器不能做间址寄存器

（4）I/O 指令的直接寻址，地址只能为 8 位

（5）正确

（6）移位超过一位时，应采用 CL 寄存器间址

（7） 16 进制数以字母开头时应在前面加“0”
（8）加法指令 ADD，缺少一个操作数
（9） I/O 指令操作数只能用 AX、AL 提供
（10）符号扩展指令为隐含操作数
（11）加法指令应为双操作数指令，立即数不能做目的操作数
（12）源操作数形式错误，SI 和 DI 不能同时做为间址寄存器
（13）AX 不能做为间址寄存器
（14）I/O 指令操作数只能用 AX、AL 提供
（15）移位指令，位移位数大于 1 时，应用 CL 来指明
（16）立即数不能做为目的操作数
（17）源、目的操作数的类型不一致
（18）传送指令中，两个操作数不能同时为存储器操作数
（19）交换指令的操作数不能使用立即数
（20）传送指令的源操作数，不能使用立即数
（21）源操作数不能为立即数，必须是存储器操作数
（22）I/O 指令端口地址表示错误，只能用 8 位立即数或 DX 间址
（23）源操作数不能为寄存器操作数
（24）乘法指令的目的操作数是隐含的，不能出现在指令中
（25）BX 和 BP 不能同时为间址寄存器
（26）比较指令的两个操作数不能同时为存储器操作数
（27）I/O 指令中，源操作数只能使用 AL 或 AX 寄存器
（28） “与”指令中，目的操作数不能使用立即数
（29）移位指令，移位位移不能用 CX 寄存器指明
（30）正确
（31）CS 段寄存器不能做为目的操作数
（32）源、目的操作数的类型不一致
（33）没有指定存储器操作数类型

2．答： (SP)＝1FFEH，(AX)＝5000H， (BX)＝5000H

3．答：（1）(AL)=7AH ，OF=0 SF=0 ZF=0 AF=0 PF=0 CF=0
(2) (AL)=0DCH ，OF=0 SF=1 ZF=0 AF=1 PF=0 CF=1
（3）(AL)=0B8H ，OF=0 SF=1 ZF=0 AF=0 PF=1 CF=1
（4）(AL)=23H ，OF=0 SF=0 ZF=0 AF=1 PF=0 CF=1
(5) (AL)=0E9H ，OF=1 SF=1 ZF=0 AF=0 PF=0 CF=1

4．答：(1) 未指明数据类型，改为：MOV BYTE PTR[1200],23H
(2) 立即数不能作为目标操作数，改为：MOV [1020H],CX
(3) 两操作数不能均是内存单元
(4) IP 不能在指令中出现
(5) 操作数必须是 16 位，改为：PUSH AX
(6) CX 不能作为端口地址的间接访问寄存器，改为：OUT DX,AL
(7) 直接端口地址写法错误，改为：IN AL,80H
(8) 两操作数的数据类型不一致，改为：MOV CX,3300H

5.答：(1) (AX)=1200H
(2) AX)=0100H
(3) (AX)=4C2AH

(4) (AX)=3412H

(5) (AX)=4C2AH

(6) (AX)=7856H

(7) (AX)=65B7H

6．答：(1) ADD DX, BX

(2) ADD AL, [BX][SI]

(3) ADD [BX+0B2H], CX

(4) ADD WORD PTR [0524H], 2A59H

(5) ADD AL, 0B5H

7．答:不正确。 正确的方法是使用两条指令实现：

```
MOV AL, [3000H]
MOV [2000H)，AL
```

8．三种方法是：

(1) 使用乘法指令：

```
MOV BL,10
MUL BI,
```

(2) 使用移位指令:

```
SHL AL,1
MOV BL，AL
SHL AL,2
ADD AL,BL
```

(3)使用加法指令:

```
ADD AL，AL
MOV BL,AL
ADD AL,AL
ADD AL,AL
ADD AL,BL
```

9．答：该段程序实现了字节数据 X 乘 10，结果在 AX 中。

10．答：程序段如下

```
（1）AND AL，0F0H
（2）OR BL，0FH
（3）XOR CL，0FH
（4） TEST DL，03H
JZ NEXT
MOV BL，1
JMP NEXT1
NEXT： MOV BL，0
NEXT1：HLT
```

11．答：程序段如下

```
MOV AX，[3000H]
ADD [2000H]，AX
MOV AX，[3002H]
ADC [2002H]，AX
```

12．答： (2000H)=39H ，(2001H)=00H 。将(2000H),(2001H)两相邻单元中存放的未组合型

BCD 码压缩成组合型 BCD 码，并存入(2000H)单元，0?(2001H)
13．答：
（1）（CL）= F6H
（2）（1E4F6H）= 78H
（3）（BX）=0056H ，（AX）=1E40H
14．答：MOV BX, AX
AND BX , 8000H
MOV CL, 4
SHL AX, CL
AND AX,7FFFH
OR AX, BX
15．答：
TEST BX,0010 0000 0000 0000B
JZ ZERO
MOV AL,01H
HLT
ZERO: MOV AL,0
HLT
16．答:将 DX: AX 中的双字左移 4 位（乘 16）

第 5 章 汇编语言伪指令

5.1 汇编语言语句的格式

一、选择题
1．B 2．B 3.C 4.B 5．C 6.A 7．D 8．C 9．B
二、填空题
1．伪指令
2．操作码、操作数（地址码）

三、分析题，
1．改正后:
```
STAKSG SEGMENT
      DB    100 DUP(?)
STAKSG  ENDS
DTSEG       SEGMENT
DATA1       DB  ?
DTSEG       ENDS
CDSEG   SEGMENT
MAIN        PROC    FAR
ASSUME CS: CDSEG, DS: DTSEG, SS: STAKSG
START:  MOV AX, DTSEG
MOV DS, AX
MOV AL, 34H
ADD AL, 4FH
```

```
MOV DATA1, AL
MOV AH, 4CH
INT     21H
MAIN    ENDP
CDSEG   ENDS
        END
```

2. 程序框图（流程图）如下：

四、简答题

1. 答：助记符：人们采用便于记忆、并能描述指令功能的符号来表示机器指令操作码，该符号称为指令助记符。

汇编语言：用助记符表示的指令以及使用它们编写程序的规则就形成汇编语言。

汇编语言程序：用汇编语言书写的程序就是汇编语言程序，或称汇编语言源程序。

汇编程序：汇编语言源程序要翻译成机器语言程序才可以由处理器执行。这个翻译的过程称为“汇编”，完成汇编工作的程序就是汇编程序（MASM.EXE）。

2. 参考答案：

```
DATA1 SEGMENT
      …
DATA1 ENDS
STACK1 SEGMENT
      …
STACK1 ENDS
COSEG SEGMENT
      ASSUME DS:DATA1,SS:STACK1,CS:COSEG
      START:MOV AX,DATA1
            MOV DS,AX
MOV AX,STACK1
            MOV SS,AX
…
                MOV AH，4CH
                INT 21H
 COSEG ENDS
            END
```

5.2 汇编语言伪指令

一、选择题

1. D　2. D　3. B　4. D　5. A　6. C

二、分析题

1. 答：X 不能多次赋值，Y 可以多次赋值。

2. 答：（1）指出程序段的功能将两个字节合并为一个字节，即两个非压缩 BCD 码合并为一个压缩 BCD 码 ；

（2）程序执行后，AL=　68H　　。

3. 答：4 字节

6 字节

4×2×5=40 字节
5×3×4=60 字节
4. 答：（1）正确。
（2）错误。宏不能精简目标代码。
（3）错误。高级语言程序经编译或解释后直接转换为目标代码。
（4）　正确。
5. 答：（1）错误。K1 是符号常数，在此处相当于立即数 100，故不能做目的操作数。
（2）正确。
（3）正确。
（4）错误。A1、A2 都是字节变量，相当于两个存储器单元，故不能同时出现在一条指令中直接进行比较。
（5）错误。用 EQU 定义的符号不能重新赋值。
6. 答: (AX)=40

第 6 章 汇编语言程序设计

6.1 汇编语言程序结构

一、程序分析题
1. 答 ：

2. 答 ：该程序段的功能：键盘输入小写字符，在屏幕上显示相应的大写字符。键盘接收字符，屏幕显示字符，返回 DOS 系统。
3.答：如果地址为 5FH 的外设输入到 AL 中的数据的最高位=1，则（AH）= 0，否则，（AH）= 0FFH。
6.2 三种程序结构编程
一、选择题
1．A
二、简答题
答：汇编语言程序的开发有 4 个步骤：
编辑：用文本编辑器形成一个以 ASM 为扩展名的源程序文件。
汇编：用汇编程序将 ASM 文件转换为 OBJ 模块文件。
连接：用连接程序将一个或多个目标文件链接成一个 EXE 或 COM 可执行文件。
调试：用调试程序排除错误，生成正确的可执行文件。
三、编程题
1.参考答案 ：

```
DATA    SEGMENT
  X       DB  15
  S       DB  ?
DATA    ENDS
CODE    SEGMENT
          ASSUME  DS:DATA, CS:CODE
START:  MOV      AX,DATA
          MOV      DS,AX
          MOV      AL,X
          TEST      AL,80H
```

```
          JZ        LL0
          SAL       AL,1      ;2X
          JMP       LL10
LL0:      CMP       AL,10
          JG        LL1
          MOV       BL,AL
          SAL       AL,1        ;3X
          ADD       AL,BL
          JMP       LL10
LL1:      SAL       AL,1        ;4X
          SAL       AL,1
LL10:     MOV       S,AL
          MOV       AH,4CH
          INT       21H         ;返回 DOS
CODE      ENDS
          END       START     ;汇编结束
```

2. 参考答案：

```
  STACK  SEGMENT STACK       ;定义堆栈段
           DB  100 DUP(?)          ;开辟 100 个存储单元
  STACK  ENDS                    ;堆栈段结束
  CODE   SEGMENT                ;定义代码段
           ASSUME CS:CODE,SS:STACK
  START:  MOV AH,01H             ;1 号调用,从键盘输入一字符存入 AL
           INT  21H                ;输入并回显
           MOV  BL,AL              ;保存从键盘输入字符
           CMP  AL,' a'                ;与“a”的 ASCII 码比较
           JB  L3                   ;低于“a”的 ASCII 码,转 L3（退出)。
           CMP  AL,' z'                ;与“z”的 ASCII 码比较
           JA  L3                   ;高于“a”的 ASCII 码,转 L3（退出)。
           SUB   AL,20H             ;将 AL 中字符的 ASCII 码减去 20H 变成大写字母。
           MOV  BL,AL              ;保存结果
           MOV  DL, '-'                ;显示字符 '-' 送 DL
           MOV  AH,02H             ;2 号调用,在显示器上显示字符 '-'。
           INT   21H                ;显示 '-'
           MOV  DL,BL              ;结果（大写字母）送 DL。
           MOV AH,02H             ;2 号调用,在显示器上显示 DL 中的内容（大写字母)。
           INT  21H                ;显示大写字母
           MOV  DL,20H            ;空格送 DL。()
           MOV AH,02H             ;2 号调用,在显示器上显示 DL 中的内容（大写字母)。
           INT  21H                ;显示空格(使两结果用空格隔开)
           JMP  START             ;不是“回车符”,转 START（继续）
    L3:    MOV AH,4CH              ;是“回车符”,设置返回 DOS 功能码。
           INT 21H                 ;返回 DOS
```

```
   CODE  ENDS                  ;代码段结束
           END START            ;程序汇编结束
```

3. 参考答案：程序段如下

```
       MOV  BX，2000H
       MOV  CX，9
MOV  AL，[BX]
  N2：INC  BX
       CMP  AL，[BX]
       JGE   N1
       MOV  AL，[BX]
  N1：LOOP  N2
       MOV  [2000H]，AL
       HLT
```

4. 参考答案：

```
DATA    SEGMENT
        DATA1   DB  00H
        DATA2   DB  00H
DATA    ENDS
CODE    SENGMENT
ASSUME  CS:CODE,DS:DATA
START:  MOV AX, DATA
        MOV  DS,AX
        MOV AH, 1
        INT     21H         ；输入一个十六进制数 0～9，A～F
        MOV DATA1,  AL
        CMP AL, 40H
        JB      L1
        SUB     AL, 31H     ；10H～15H，A~F 的数转换成 BCD 码
        JMP     L2
L1: SUB     AL, 30H     ；00H～09H；0~9 转换成 BCD 码
L2: MOV BL, AL
        AND AL, 0F0H
        MOV CL, 4
        SHR     AL, CL
        ADD AL,     30H；BCD 码的高位转换成字符
        MOV DL,     AL
        MOV AH,  2
        INT     21H         ；显示高位
        AND BL, 0FH
        ADD BL, 30H ；BCD 码的低位转换成字符
        MOV DL,     BL
        MOV AH,  2
        INT     21H     ；显示低位
```

```
        MOV AH, 4CH
        INT     21H
CODE    ENDS
    END  START
```

4. 参考答案：

```
DATA  SEGMENT
        DA1    DD  12345678H
        COUNT   DB      00H
DATA  ENDS
CODE    SENGMENT
ASSUME  CS:CODE,DS:DATA
START:  MOV AX, DATA
            MOV DS, AX
            MOV   SI,   OFFSET  DA1
            MOV     BL, 00H
            MOV BH, 8
            MOV CX, 4
A2: MOV AL, BYTE  PTR [SI]
         A1:    ROR AL, 1
            JNC     L1
            INC     BL
L1: DEC BH
            JNZ   A1
            INC     SI
            LOOP    A2
            MOV COUNT,  BL
            MOV AH, 4CH
            INT     21H
CODE    ENDS
            END     START
```

6. 参考答案：

```
Code segment
    Assume  CS:Code
abc:    mov     ah,1
    int     21h
    cmp     al,' a'
    jb      stop
    cmp     al,' z'
    ja      stop
    sub     al,20h
    mov     dl,al
    mov     ah,2
    int     21h
    jmp     abc
```

```
stop: mov ah,4ch
    int 21h
code  ends
    end
```

7. 参考答案：

```
START: IN   AL,71H      ;将 71H 端口的字节读入 AL
       CLC              ;清除 CF
       CMP  AL,10
       JC   LP1         ;小于 10 转 LP1
       CMP  AL,20
       JC   LP2         ;小于 20 大于等于 10 转 LP2
       MOV  BL,0FFH
LP3:   MOV  AL,BL
       OUT  73H,AL      ;将 AL 的 0FFH 输出到 73H 端口
       HLT
LP1:   MOV  BL,00
       JMP  LP3
LP2:   MOV  BL,88H
       JMP  LP3
```

第 7 章 输入输出系统

7.1 输入输出接口概念

一、选择题

1. A　2. A　3. A　4. A　5.A

二、填空题

1.外设与 CPU 通信的控制部件， CPU 与外设间传递信息的

2.总线和外部设备、信息转换和数据传递、数据信息、控制信息、状态信息

3. I/O 端口与内存统一编址和 I/O 端口独立编址

4. CPU 与外设、 CPU 小批量慢速数据输入/输出设备传送

5.内存与外设、DMA 控制器

三、简答题

1. 答：位于主机与外设之间，用来协助完成数据传送和控制任务的逻辑电路称为接口电路，接口电路对输入/输出过程起缓冲和联络作用。接口的功能是有，数据缓冲功能，联络功能，寻址功能，预处理功能，中断管理功能。

2. 答：I/O 端口和存储器统一编址；I/O 端口单独编址。8086 系统采用 I/O 端口单独编址方式。

3.答：因为许多接口设备中，在工作原理，驱动方式，信息格式以及工作速度方面彼此相差很大，因此为了进行速度和工作方式的匹配，并协助完成二者之间数据传送控制任务。

4. 答：数据输入寄存器，数据输入寄存器，状态寄存器和控制寄存器。数据端口能对传送数据提供缓冲，隔离，寄存的作用 ；状态寄存器用来保存外设或接口的状态；控制寄存器用来寄存 CPU 通过数据总线发来的命令。

5. 答：输入/输出指令和访问存储器的指令明显区分开，使程序清晰，可读性好；而且 I/O 指令长度短，执行的速度快，也不占用内存空间，I/O 地址译码电路较简单。不足之处是 CPU 指令系统中必须有专门的 IN 和 OUT 指令，这些指令的功能没有访问存储器的指令的功能强；I/O 端口数目有限。另外，CPU 要能提供区分存储器读/写和 I/O 读/写的控制信号。

6．答：外设接口译码电路各接口芯片的地址如下表：

7.2 数据传送控制方式

一、选择题

1． C 2．A 3．D

二、简答题

1.答：程序控制方式：特点是依靠程序的控制来实现主机和外设的数据传送,可分为无条件传送方式和查询方式.

中断控制方式：每次输入和输出一个数据，CPU 都要检查外设的状态。

直接存储器存取控制方式： cpu 不参加数据传送，而是由 DMA 控制器来实现内存与外设，外设与外设之间的直接传递。

通道方式：可以实现对外围设备的统一管理和外围设备与内存之间的数据传送。

外围处理机方式：由 PPU 独立于主机工作，减少了 CPU 控制外设的负担。

2.答：查询传送方式 CPU 通过程序不断查询相应设备的状态，状态不符合要求，则 CPU 需要等待；只有当状态信号符合要求时，CPU 才能进行相应的操作。中断方式提高了计算机系统中信息处理的并行和处理器效率，中断可以实现同步操作，实时处理等功能。

3.解：一次 DMA 传送过程由传送前的预处理、数据传送、传送结束 3 个阶段组成。

预处理是由 CPU 完成的。当 CPU 执行到读写 I/O 设备调用语句时，启动 DMA 传送过程，向 DAM 卡送入设备识别信号、启动设备，测试设备运行状态，送入内存地址初值，传送数据个数，DMA 的功能控制信号等，之后，CPU 继续执行原来程序。

数据传送在 DMA 卡控制下自动完成。DMA 卡向 CPU 发出请求总线使用权的信号，若总线空闲，总线控制器将送响应回答信号给 DMA 卡，DMA 卡取得总线使用权，清“0”DMA 请求触发器以撤消请求总线的信号，并启动数据传送过程。DMA 在传送过程中还要完成对内存地址计数器和数据数量计数器的计数操作，并通过检查数据数量计数器是否为 0，决定要启动下一次传送，还是结束本批全部数据的传送过程。

传送结束处理，是由数据数量计数器的值为 0 引发出来的。当数据数量计数器的值为 0 时，DMA 将向 CPU 发出中断请求信号，CPU 响应这一请求后，转入中断服务程序；检查是否结束数据传送。

4.答：异步通信的帧格式是用一个起始位表示传送字符的开始，用 1-2 个停止位表示字符结束。起始位与停止位之间是数据位，数据位后是校验位，数据的最底位紧跟起始位,其他各位顺序传送;同步通信的帧格式是在每组字符之前必须加上一个或多个同步字符做为一个信息帧的起始位。

5. 答：波特率是单位时间内通信系统所传送的信息量。 需要多长时间=1024/（1200/10）=8.53s

6．答：

```
MOV     SI,2000H            ；初始化数据区地址
MOV     AX,1000H
MOV     DS, AX
        LL: MOV     DX,0FFE2H
            IN      AX,DX           ;读状态
TEST    AL,01H          ；测试是否满足就绪？
JZ      LL              ；不满足，继续读状态
MOV     DX,0FFE0H       ；就绪，准备输入数据
IN      AX,DX           ；从输入接口读取数据
```

```
MOV [SI]，AX          ；存数据
INC          SI
LOOP     LL                ；数据没有输入完，继续
```

7.3 并行通讯和并行接口芯片 8255

一、选择题

1. A　2. B　3. A　4. D　5. D

二、简答题

1. 答：主要特点：数据并行传输，传输速度快，但距离较近。

主要功能：并行传输数据，在主机与外设之间起到数据缓冲和匹配的作用。

2. 答：8255A 有 3 种工作方式：称为方式 0，方式 1 和方式 2。

其中 A 口可以工作在 3 种方式中的任一种；B 口只能工作在方式 0 和方式 1；C 口通常作为控制信号使用，配合 A 口和 B 口的工作。

①方式 0：基本的输入输出方式，A 口、B 口和 C 口(C 口分为 2 个 4 位使用)都可提供简单的输入和输出操作，对每个口不需要固定的应答式联络信号。在程序中可直接使用输入指令(IN)和输出(OUT)指令对各口进行读写。方式 0 一般用于无条件传送的场合，不需要应答式联络信号，外设总是处于准备好的状态。

②方式 1：选通输入/输出方式，当 A 口和 B 口进行输入输出时，必须利用 C 口提供的选通和应答信号。而且这些信号与 C 口中的某些位之间有着固定的对应关系，这种关系是硬件本身决定的不是软件可以改变的。由于工作在方式 1 时，要由 C 口中的固定位作为选通和应答等控制信号，因此称方式 1 为选通的输入/输出方式。

③方式 2：带选通的双向传输方式，8255A 可以向外设发送数据，同时 CPU 通过这 8 位数据线又接收外设的数据。因此称为双向的传输方式。只能适用于 A 口。一个 8 位的双向口(A 口)和 1 个 5 位的控制口(C 口)。A 口的输入和输出都可以被锁存。5 位的控制口用于传送 8 位双向口的控制和状态信息。

三、应用编程题

1. 答：由题知应为 10111001H=B9H

```
MOV  AL，B9H
MOV  DX，006CH
OUT   DX，AL
```

2.答：

```
MOV  DX，00C0H      ；端口地址
MOV  AL，00001101   ；对 PC6 置 1
OUT   DX，AL
MOV  AL，00001000   ；对 PC4 置 0
OUT   DX，AL
```

3.答：

```
（1）MOV   AL, 10011001B  ; A 和 B 组方式 0,A 和 C 口输入口,B 口作为输出口
          MOV   DX, P1
          OUT   DX, AL
（2）MOV    AL, 11000100B  ; A 组置成方式 2,B 组置成方式 1,B 口作为输出口
          MOV   DX, P1
          OUT   DX, AL
（3）MOV    AL, 10110110B  ; A 口方式 1 输入,PC6 和 PC7 输出,B 口方式 1 输入
          MOV   DX, P1
          OUT   DX, AL
```

4. 答：

```
MOV  AL，  80H
```

```
       OUT   8BH，AL
       MOV   AL，0DH
       OUT   8BH，AL
MOV   AL，06H
OUT   8BH，AL
```

5. 答：(1)MOV   DX,203H

```
MOV      AL,10111000B
OUT      DX,AL
(2) MOV      DX,202H
IN       AL,DX
MOV      AH,AL
TEST     AL,80H
JNZ      NEXT1
MOV      DX,203H
MOV      AL,00000011B  ；对 PC1 置位
OUT      DX，AL
NEXT1：MOV       AL，AH
TEST     AL，40H
JZ       NEXT2
MOV      AL，00000000B ；对 PC0 复位
MOV      DX，203H
OUT      DX，AL
NEXT2：……
```

6. 答：（1）PORTA   EQU   60H

```
PORTB    EQU   62H
PORTC    EQU   64H
PCTRL    EQU   66H
（2） TAB1   DB   0C0H,…   ;字型表
…
MI1     PROC
      MOV     AL,88H
      OUT     PCTRL,AL      ;初始化 8255
      IN      AL,PORTC      ;读开关状态
      MOV     CL,4
      SHR     AL,CL          ;转成低 4 位
      LEA     BX, TAB1
      XLAT    TAB1        ;查表
      OUT          PORTA,AL      ;显示输出
      RET
MI1   ENDP
```

7. 答：（1） 控制字为=1 001 0 1 00=94H

```
MOV  AL, 94H
OUT 83H, AL
```

（2）端口地址的译码电路，如图所示。

8．答：地址：208H~20BH
控制字=1 00 1 0 0 00 =90H，A 口发生 0 输入，B 口方式 0 输出
程序段如下：

```
MOV  DX , 20BH
MOV  AL, 90H
OUT  DX, AL
MOV  DX, 108H
IN    AL, DX            ;读 A 口
INC   DX
OUT  DX, AL            ;写 B 口
```

9. 答：控制字=1 000 0 1 0 0=84H，B 口方式 1 输出
允许 B 口中断，可以查询中断指示状态：PC2 置 1，则置位字=0 000 010 1B=05H

```
MOV  AL, 84H
OUT  93H, AL  ;写控制字
MOV  AL, 05H
OUT  93H, AL   ;写中断允许
LEA  SI , Buf
MOV  CX, 1000
L1:  MOV  AL,  [SI]      ;将数据传送至 AL
     OUT  91H,  AL       ;从 A 口输出数据，  使  变为低电平
L2:  IN   AL,  92H       ;从 C 口读取状态字
                ;打印机取走数据后，  变为低电平，并将  变为高电平
    AND  AL,  01H        ;判断 B 口是否提出中断申请
     JZ   L2            ;如果没有，继续检测状态字
    INC  SI              ;
    DEC  CX
    JNZ  L1
    HLT
```

10．分析：由于 8255A 的 A 口以方式 1 工作，因此将 8255A 的 PA7～PA0 与打印机的数据线 D7～D0 连接，PC7 作为 输出信号与打印机的数据选通信号 引脚相连，PC6 作为 输入信号与打印机的应答信号 相连，PC4 用来查询打印机的忙信号 BUSY 的状态。在这里应该注意，当 CPU 输出数据时，8255A 产生一个低电平有效的 输出信号，当 8255A 接收到一个响应信号 时，才能恢复为高电平。另一方面，打印机需要一个数据选通信号 才能接收数据，而 是一个低脉冲信号，因此直接将 与 相连，将会因为互相等待而产生“死锁”。采用单稳态电路 74LS123 即可满足 8255A 和打印机双方的时序要求，因为单稳态电路只要输入一个下降沿信号就可以输出一个低脉冲信号。

打印机的工作原理是：当数据选通信号 （负脉冲）有效时，数据线 D7～D0 上的数据被锁存到打印机内部的数据缓冲区中，同时将忙信号 BUSY 置 1，表示打印机正在处理输入的数据，等到输入的数据处理完毕，撤消忙信号，将 BUSY 清 0，同时送出应答信号 ，表示一个字符已经输出完毕。
打印驱动程序编制如下：

```
DATA    SEGMENT
         BUFFER  DB  100H DUP(?)
```

```
DATA    ENDS
CODE    SEGMENT
        ASSUME    CS:CODE, DS:DATA
START:  MOV  AX, DATA
    MOV  DS, AX
        MOV  AL, 0A8H           ; A 口方式 1 输出, PC4 输入
    MOV  DX, 0FFE3H         ; 控制口地址
    OUT  DX, AL             ; 控制字写入控制口
    MOV  CX, 100H           ; 传送字节数送 CX 寄存器
    MOV  SI, OFFSET BUFFER ; 数据缓冲区首地址送 SI 寄存器
L1:  MOV  DX, 0FFE2H         ; C 口地址
    IN   AL, DX             ; 读 C 口内容, 查询 BUSY 信号
  AND  AL, 10H            ; 保留 PC4 状态, 判断 BUSY=1?
  JNZ  L1                  ; BUSY＝1, 打印机处于忙状态, 应该继续查询
  MOV  AL, [SI]            ; BUSY＝0, 打印机处于空闲状态, 可以输出数据
  MOV  DX, 0FFE0H         ; A 口地址
    OUT  DX, AL             ; 输出数据
    INC  SI                 ; 修改数据缓冲区地址
    LOOP  L1                ; 数据未传送完毕, 继续传送
    MOV  AX,                ; 数据传送完毕, 返回 DOS
    INT  21H
CODE  ENDS
        ENDS  START
```

7.4 串行通讯和串行接口芯片 8251

一、填空题

1. 低
2. 5-8
3. 移位
4. 同步和异步
5. 远
6. 相同

二、简答题

1.答：传输方式可分为单工方式、半双工方式、全双工方式。

（1） 对传输速率有严格要求。

（2） 采用单条传输线来传输数据，减小了传输成本，增加了收发双方的复杂性。

（3） 传输过程中，由于引起误码，需差错控制。

2. 答：串行通信中：数据传送方式是串行的（一位一位传送），数据传送速度较慢，但成本低，适用于远距离传送。

并行通信中：数据传送方式是并行的（数位一起传送），数据传送速度较高，但成本较高，适用于近距离通信。

3. 答：同步通信是用时钟信号加载传输信号的，因些收发时钟频率=收发波特率；异步通信情况下，接收时钟频率=n×接收波特率，其中 n=1,16,64；发送时钟频率可以等于波特率，也可以为 n×发送波特率，但考虑到时钟与接收时钟一致，故发送时钟频率=n×发波特率，其中 n=1,16,64。

4．答：起始位置和停止位置的作用是使收发双方在随机传送的字符与字符间实现同步。接收端在检测到起始位时，便知道字符已经到达，应开始接收字符；当检测到停止位时，则知道字符传送已经结束。

5．答：每帧占1?7?1?1=10位，波特率为4800 bit/s,故每分钟能传送的最大字符数为28800（个）（4800*60/10）

6．答:每个字符需要的发送位数是12位（数据位8位，校验位1位，停止位2位，起始位1位)。每秒发送100个字符共1200位。因此波特率为1200波特，位周期= ≈833μs。

第8章 控制器芯片

8.1 中断控制器8259

一、选择题

1．D　2．A　3．A　4．A　5．B

二、简答题

1．答：8259A通过级联的方式由9片构成最多64级优先权的中断源。

2．答：8259A的内部结构有数据总线缓冲器，读写逻辑电路，级联缓冲比较器，中断请求寄存器（IRR)，中断屏蔽寄存器（IMR)，中断服务寄存器（ISR)，优先权判别器（PR)，控制逻辑。

3．答：8259A有2种中断结束方式：中断自动结束方式，中断非自动结束方式（一般中断和特殊中断)；中断自动结束方式只适合有一块8259A，并且各中断不发生嵌套的情况。中断非自动结束方式只能适合与全嵌套方式下不能用与循环优先级方式。

4．答：有4种，普通全嵌套方式，特殊全嵌套方式，自动循环方式，优先级特殊循环方式

5．答：8259A的初始化命令字ICW1，ICW2，ICW3，ICW4；操作命令字OCW1，OCW2，OCW3；ICW2，ICW3，ICW4，OCW1写入奇地址，ICW1，OCW2，OCW3为偶地址。

6．答：中断请求寄存器IRR：保存8条外界中断请求信号IR0～IR7的请求状态。Di位为1表示IRi引脚有中断请求；为0表示该引脚无请求。

中断屏蔽寄存器IMR：保存对中断请求信号IR的屏蔽状态。Di位为1表示IRi中断被屏蔽（禁止)；为0表示允许该中断。

中断服务寄存器ISR：保存正在被8259A服务着的中断状态。Di位为1表示IRi中断正在服务中；为0表示没有被服务。

7．答：8259A中断控制器可以接受8个中断请求输入并将它们寄存。对8个请求输入进行优先级判断，裁决出最高优先级进行处理，它可以支持多种优先级处理方式。8259A可以对中断请求输入进行屏蔽，阻止对其进行处理。8259A支持多种中断结束方式。8259A与微处理器连接方便，可提供中断请求信号及发送中断类型码。8259A可以进行级连以便形成多于8级输入的中断控制系统。

三、应用题

1．答：？8259A占2个端口地址为20H，22H或24H，26H 其中ICW1的设置地址为20H或24H

？8255A占4个端口地址为80H，82H，84H，86H ，控制寄存器的地址为86H。

2．答：（1）MOV AL,0001011B

OUT 20H,AL

MOV AL,00110010B

OUT 21H,AL

MOV AL,00010011B

```
        OUT 21H,AL
```
（2）如果显示 E，则端口 A 送出的数据是 30H；
如果显示 0，则端口 A 送出的数据是 01H；
程序如下：
```
MOV AL,10000000B
        OUT 63H,AL
        MOV AL,30H
OUT 60H,AL
MOV AL,01H
OUT 60H,AL
```
3. 答：根据已知条件得到以下预置命令字为：
ICW1=00010011=13H ,A0=0 偶地址
ICW2=01001000=48H ,A0=1 奇地址
ICW4=00000011=03H ; A0=1 奇地址
初始化程序段为：
```
        MOV DX,120H
        MOV AL,13H
OUT  DX,AL    ;写入预置命令字 1
MOV DX,121H
        MOV AL,48H
OUT  DX,AL    ;写入预置命令字 2
MOV DX,121H
        MOV AL,03H
OUT  DX,AL    ;写入预置命令字 4
```

8.2 定时/计数控制器 8253

一、选择题
1. C 2. D 3. C 4. A 5. D 6. D 7. C 8. C 9. A 10. A
二、简答题
1．答：8253 有三个计数通道，每个计数通道有 3 条信号线：CLK：计数输入用于输入定时基准脉冲或计数脉冲.OUT:输出信号以相应的电平指示计数的完成或输出脉冲的波型.GATA:选通输入用于启动或禁止计数器的操作，以使计数器和计数输入信号同步。
2．答：CLK 为计数时钟输入引脚，为计数器提供计数脉冲；GATE 为门控信号输入引脚，用于启动或禁止计数器操作，如允许/禁止计数、启动/停止计数等；OUT 为输出信号引脚以相应的电平或脉冲波形来指示计数的完成、定时时间到。
三、编程题
1．答：
```
MOV AL,00011110H   ; 控制字
OUT 43H,AL
MOV AL,3000H       ;计数初值
OUT 40H,AL
MOV AL,01010110H   ; 计数器 1
OUT 43H,AL
MOV AL,100H
OUT 41H,AL
MOV AL,10011000H   ; 计数器 2
```

```
OUT 43H,AL
MOV AL,4030H
OUT 42H,AL
```
2. 答：(1) MOV AL, 50H
```
   OUT  07H, AL
   MOV  AL, 128
   OUT  05H, AL
(2)  MOV  AL, 33H
        OUT  07H, AL
        MOV  AX, 3000
OUT  04H, AL
MOV  AL, AH
OUT  04H,A;
```
（3） MOV AL, 0B4H
```
OUT  07H, AL
MOV  AL, 0F0H
OUT  06H,AL
MOV  AL, 02H
OUT  06H, AL
```
3. 答：编写 8253 的初始化程序

（1）确定端口地址：0310H、0312H、0314H、0316H

（2）确定工作方式：通道 0，方式 3

通道 1，方式 1

通道 2，方式 5

（3）确定计数值： 通道 0：N0=1MHz/2KHz=500

通道 1：N1＝480us/(1/1mhz)=480

通道 2：N2=26

（4）确定控制字： 通道 0：00110111B

通道 1：01110011B

通道 2：10011011B

对 3 个通道的初始化程序如下：

```
；通道 0 初始化程序
MOV DX, 316H
MOV AL,00110111B
OUT DX,AL
MOV DX,310H
MOV AL,00H
OUT DX, AL
MOV AL,05H
OUT DX,AL
；通道 1 的初始化程序
MOV DX, 316H
MOV AL, 001110011B
OUT DX, AL
```

```
MOV DX, 312H
MOV AL, 80H
OUT DX, AL
MOV AL, 04H
OUT DX, AL
；通道 2 初始化程序
MOV DX, 316H
MOV AL, 10011011B
OUT DX,  AL
MOV DX, 314H
MOV AL，26H
OUT DX，AL
```

4．计数器 0　　工作于方式 3　　45.454KHZ

5．答：（1） 8255 的端口地址为 80H， 82H， 84H， 86H

8253 的端口地址为 90H， 92H， 94H， 96H

8259 的端口地址为 A0H， A2H，

8251 的端口地址为 B0H， B2H，

(2) OUT 80H， AL

IN AL， 82H

6．答：(1) 程序对 8253 的通道 1 进行初始化。

(2)计数常数为 10000D，BCD 计数。

(3)工作在方式 3，方波速率发生器周期=10000?1μs=10000μS=10ms。

7．答：由于 A6 和 A2 不受约束，所以共有四组可选地址

（A6=0）： 3A0H~3A3H （A2=0） 3A4H~3A7H （A2=1）

（A6=1）： 3E0H~3E3H （A2=0） 3E4H~3E7H（A2=1）

选择 3A0H~3A3H 这组地址： 用计数器 0，工作方式 3

计数初值为：N=1MHZ/1KHZ=1000

8253 初始化程序段如下：

```
MOV  AL，36H  ；0011 0110B
MOV  DX，3A3H
OUT  DX，AL
MOV  AX，1000
MOV  DX，3A0H
OUT  DX，AL
MOV  AL，AH
OUT  DX，AL
```

8.3 DMA 控制器 8237

1．答：DMA 控制器能给出访问内存所需要的地址信息，并能自动修改地址指针，也能设定和修改传送的字节数，还能向存储器和外设发出相应的读/写控制信号。在 DMA 传送结束后，它能释放总线，把对总线的控制权又交还给 CPU。用 DMA 方式传输数据时，不需要进行保护和恢复断点及现场之类的额外操作。

2．答：8237A 取得总线控制权以后进行单字节的 DMA 传送，传送完一个字节以后修改字节计数器和地址寄存器，然后就将总线控制权放弃.若 I/O 的 DMA 请求信号 DREQ 继续有效，

8237A 再次请求总线使用权进行下一字节的传送。

第 9 章 转换器芯片

9.1 D/A 转换及转换器

一、选择题

1．B　　2．D

二、填空题

1.分辨率

2. ADC，0809

3. DAC，运算放大器

二、应用题

程序段如下：

```
              MOV    DX, PORT          ; DAC 端口地址
              MOV    AL, 00H           ; 初始值
REPEAT:  OUT    DX, AL            ; 输出, 完成 D/A 转换
              INC    AL                ; 增量
             JMP    REPEAT            ; 重复转换过程
```

9.2 A/D 转换及转换器

一、选择题

1．C　　2．B　　3．B　　4．A　　5．B

二、简答题

1. 答：ADC 把模拟量转换为数字量信号，分为四步来完成：采样、保持、量化、编码。转化过程可以用逐次逼近型电路、V/F 转换型电路和双积分型电路。

2.答：ADC 与微处理器接口的基本任务是:向 ADC 转发启动转换信号;向 CPU 提供转换结束信号,把转换好的数据送入微处理器。

3. 答：可采用直接与 CPU 的 INTR 引脚连接，或通过 8259A 接 CPU。

4．答：传感器：将各种现场的物理量测量出来并转换成电信号。

放大器：放大器把传感器输出的信号放大到 ADC 所需的量程范围。

低通滤波器：滤波器用于降低噪声、滤去高频干扰，以增加信噪比。

多路开关：对多个模拟信号分时地接通到 A/D 转换器上转换，达到共用 A/D 转换器以节省硬件的目的。

采样保持器：对高速变化的信号，使用采样保持器可保证 A/D 转换期间信号不变，保证转换精度。