

课后题答案+解析：

检测点1.1

```
3 2:1024 0 1023
4
5 3:2^13 2^10
6
7 4:2^30 2^20 2^10
8
9 5:64 1 16 4
10
11 6:1 1 2 2 4
12
13 7:512 256
14
15 8:二进制
16
17 题解：
18
19 1:cpu的寻址能力是8KB，地址总线宽度为10根的寻址能力是1KB=1024b=2^10,所以2^x=8KB=8*1024=2^13
20
21 所以x=13。
22
23 2:1B是一个存储单元，1KB=1024B，等于1024个存储单元，由于存储单元的范围是从(00000 00000 ~ 11111 11111)转化成十进制就是0~1023，
24
25 的编号是0到1023。
26
27 3:1B=8bit,1KB=2^10B=8*1024bit=2^13。
28
29 4:1GB=1024*1KB=10241MB=1024*1024*1KB=1024*1024*1024*1B
30
31 1GB=2^30B,1MB=2^20B,1KB=2^10B。
32
33 5:10根地址总线的寻找能力是1KB，那么16是2^6KB,20是1024KB=1MB，24是2^4*1MB=16MB,32是1024*4*1MB=4GB。
34
35 6:一根数据线，传输1bit，八根传输8bit，8bit=1B,那么答案分别是1，1，2，2，4。
36
37 7:8086每次读取2B(字节),1024B/2B=512次,80386每次读取4字节，那么答案:1024B/4B=256次。
```

检测点2.1

```
1 (1) 写出每条汇编指令执行后相关寄存器的中的值
2 mov ax,62627 AX=F4A3H
3 mov ah,31H AX=31A3H
4 mov al,23H AX=3123H
5 add ax,ax AX=6246H
6 mov bx,826CH BX=826CH
7 mov cx,ax CX=6246H
8 mov ax,bx AX=826CH
9 add ax,bx AX=04D8H
10 mov al,bh AX=0482H
11 mov ah,bl AX=6C82H
12 add ah,ah AX=D882H
13 add al,6 AX=D888H
14 add al,al AX=D810H
15 mov ax,cx AX=6246H
16 (2)利用已写学习的汇编指令编写出2^4
17 mov ax,2 AX=2
18 add ax,ax AX=4
19 add ax,ax AX=8
20 add ax,ax AX=16
```

检测点2.2

```
1 (1) 00010H,1000FH
2
3 (2) 1001H,2000H
4
5 题解：
6 (1)物理地址=给定段地址*16+(0~FFFF)
7 所以范围:00010H+0H~00010H+FFFFH=(00010H~1000FH)
8 (2)设给定段地址为XH:
9 最大段地址:X*16=2000H
10 X=2000H
11 最小段地址:X*16+FFFF=2000H
12 X*16=10001H
13 但是X右移一位为1000H<1000.1H
14 所以X=1001H
15
16 思考
17 所以当X<1001H或者X>2000H时，cpu无论怎么变化偏移地址都无法寻到20000H单元
```

检测点2.3

```
1 下面的三条指令执行后，cpu几次修改IP?都是在什么时候，最后IP的值是多少？
2 mov ax,bx
3 sub ax,ax
4 jmp ax
5 答案：（1）4次
6 （2）
7 第一次读取mov ax,bx
8 第二次读取sub ax,ax
9 第三次读取jmp ax
10 第四次执行jmp ax
11 （3）
12 最后 IP的值是0，
13 假设CS=a,IP=b
14 那么执行完第一条指令后
15 CS=a,IP=b+2
16 ax=bx
17 执行完第二条指令
18 CS=a,IP=b+4
19 ax=0
20 执行完第三条指令
21 首先CS=a,IP=b+4
22 此时第三条指令读取完毕
23 然后执行第三条指令
24 IP=ax=0
25 CS=a,IP=0
26 最后IP是0
27
```

检测点3.1

```
1 (1)
2 AX=2662H
3 BX=E626H
4 AX=E626H
5 AX=2662H
6 BX=D6E6H
7 AX=FD48H
8 AX=2C14H
9 AX=0000H
10 AX=00E6H
11 BX=0000H
12 BX=0026H
13 AX=000CH
14 (2)
15 CS=2000H,IP=0,DS=1000H
16 AX=0,BX=0
17
18 mov ax,6622H
19 CS=2000H IP=3 DS=1000H AX=6622H BX=0000H
20 jmp 0FF0:0100
21 CS=1000H IP=0 DS=1000H AX=6622H BX=0000H
22 mov ax,2000H
23 CS=1000H IP=3 DS=1000H AX=2000H BX=0000H
24 mov ds,ax
25 CS=1000H IP=5 DS=2000H AX=2000H BX=0000H
26 mov ax,[0008]
27 CS=1000H IP=8 DS=2000H AX=C3B9H BX=0000H
28 mov ax,[0002]
29 CS=1000H IP=000B DS=2000H AX=EA66H BX=0000H
30 (3)
31 数据和程序在计算机中都是以二进制的形式存放的，
32 在区别程序和数据时，关键是看段地址，如果段地址是ds
33 段，说明该内存存放的是数据，如果段地址是cs段，说明该内存
34 存放的是指令。
35
36
```

检测点3.2

```
1 (1)
2 mov ax,2000
3 mov ss,ax
4 mov sp,0010
5 (2)
6 mov ax,1000
7 mov ss,ax
8 mov sp,0000
9
```

检测点 6.1

```
1 (1)
2 mov cs:[bx],ax
3 (2)
4 cs
5 32
6 pop cs:[bx]
```

检测点 9.1

```
1 (1)
2 1:db 0,0,0,0
3 2:dw 2 dup(0)
4 (2)
5 0
6 cs
7 (3)
8 0006H
9 00BEH
10 注意:理解 word ptr 和 dword ptr的概念
```

检测点 9.2

```
1 (1)
2 mov ch,0
3 mov cl,[bx]
4 jcxz ok
5 inc bx
6
7 解释：我们要确保cx的高位为0
8 mov ch,0
9 mov cl,[bx]
10 如果当前的cx==0，那么执行jcxz jmp short ok
```

检测点 9.3

```
1 (1)
2 inc cx
3
4 解释：不能让cx=0,不然在loop指令下无法退出，如果cx=0，我们让cx+1，
5 此时就可以退出了
```

检测点 10.1

```
1 (1)
2 1000h,0
3 先把1000h推入栈，在把0
4 retf 执行后，cs=1000,ip=0
5
6
```

检测点 10.2

```
1 (1)6
2 解释，
3 执行完call s 指令被读取完后,ip->6,然后执行call s指令，将 当前ip的值
4 压入栈，转跳到标号 s ，执行pop ax,此时ax=6
5
6
```

检测点 10.3

```
1 1000:0 mov ax,0
2 1000:3 call far ptr s
3 1000:8 inc ax
4 1000:9 s:pop ax
5 add ax,ax
6 pop bx
7 add ax,bx
8 (1)1010
9 ax=0
10 push 1000
11 push 0008
12 jmp 1000:9
13 pop ax,8 ax=1008
14 add ax,ax ax=0010
15 pop bx bx=1000
16 add ax,bx ax=1010
```

```
1 (1) 3
2
3 assume cs:code
4
5 stack segment
6
7 dw 8 dup(0)
8
9 stack ends
10
11 code segment
12
13 start:
14
15 mov ax,stack ;ax=栈段地址
16
17 mov ss,ax ; ss栈段地址=ax
18
19 mov sp,16 ; 栈顶偏移地址sp=16=10H
20
21 mov ds,ax ;段地址 ds=ax 和栈段一致
22
23 mov ax,0
24
25 call word ptr ds:[0EH] ; push ip ,ip= inc ax处的ip
26
27 ; 此时 ss:0eh: inc ax处的ip
28
29 ; ds=ss:[0]=ss:0eh
30
31 inc ax ;执行下面语句
32
33 inc ax
34
35 inc ax ax=3
36
37 mov ax,4c00H
38
39 int 21H
40
41 code ends
42
43 end start
44 (2)
45 assume cs:code
46 data segment
47 dw 8 dup (0)
48 data ends
49
50 code segment
51
52 start: mov ax,data
53 mov ss,ax
54 mov sp,16
55 mov word ptr ss:[0],offset s
56 mov ss:[2],cs
57 call dword ptr ss:[0]
58 nop
59 s: mov ax,offset s
60 sub ax,ss:[0CH]
61 mov bx,cs
62 sub bx,ss:[0EH]
63 mov ax,4c00H
64 int 21H
65
66 code ends
67
68 end start
69 解释：
70 首先 我们让栈地址[0],[2]存放cs,和标号处的ip,然后执行call dword ptr
71 ss:[0],此时 push nop 处的cs和ip,sub ax,ss:[0H],因为栈里存的是一条空语句地址。
72 所以ax=1,因为cs相同所以 bx=0
```

检测点 11.1

```
1 (1)
2 ZF PF SF
3 sub al,al 1 1 0 结果0b 1 1 0 为0 0个1 非负
4 mov al,1 1 1 0 传送指令不影响标志寄存器的状态
5 push ax 1 1 0 传送指令不影响标志寄存器的状态
6 pop bx 1 1 0 传送指令不影响标志寄存器的状态
7 add al,bl 0 0 0 结果10b 不为0 1个1 非负
8 add al,10 0 1 0 结果110b 2个1 非负
9 mul al 0 1 0 结果1001000b 2个1 非负
10 (2)
11 CF OF ZF PF SF
12 sub al,al 0 0 1 1 0
13 mov al,10h 0 0 1 1 0
14 add al,90h 0 0 0 1 1
15 mov al,80h 0 0 0 1 1
16 add al,80h 1 1 1 1 0
17 mov al,0fch 1 1 1 1 0
18 add al,05h 1 0 0 0 0
19 mov al,7dh 1 0 0 0 0
20 add al,0bh 0 1 0 1 1
```

检测点 11.3

```
1 (1)
2 jb s0
3 ja s0
4 (2)
5 jna s0
6 jnb s0
```

检测点 11.4

```
1 (1)
2 45h
3 解释：通过操作标志寄存器的值为000000001000101b
4 最后ax的值是45H
```

检测点 12.1

```
1 (1)0070:018BH
2 (2)4N,4N+2
3 解释：
4 可以查看段地址 0070=3*4+2位置
5 偏移地址：018B=3*4位置
6
```