2.1

1. ＭＯＶ ＤＩ， ［ＥＡＸ］

源操作数：寄存器间接寻址 目的操作数：寄存器寻址

（4）ＳＵＢ ＤＸ， ５［ＢＸ ＋ＤＩ］

源操作数：基址变址寻址 目的操作数：寄存器寻址

（5）ＭＯＶ ［ＥＤＩ\*４ ＋２］， ＡＸ

源操作数：寄存器寻址 目的操作数：变址寻址

（６）ＭＯＶ ＢＨ， ＤＳ： ［１０］

源操作数：直接寻址 目的操作数：寄存器寻址

（ＥＡＸ） ＝０ＡＡＨ　　　（ＥＢＸ） ＝０ＢＢＨ　　　　　（ＥＣＸ） ＝０ＣＣＨ　　　　　（ＥＤＸ） ＝０ＤＤＨ

（ＥＢＰ） ＝５０００Ｈ （ＥＳＩ） ＝４０００Ｈ （ＥＤＩ） ＝６０００Ｈ （ＤＳ） ＝１０００Ｈ （ＥＳ） ＝２０００Ｈ （ＳＳ） ＝３０００Ｈ

以下存储单元的内容均是按字类型说明的：

（３５０００Ｈ） ＝１００Ｈ （１００ＢＢＨ） ＝２００Ｈ （１６０００Ｈ） ＝３００Ｈ （２４０００Ｈ） ＝４００Ｈ

（３５５００Ｈ） ＝３５５Ｈ （１３Ｄ００Ｈ） ＝０Ｆ１３ＤＨ （１７０００Ｈ） ＝０Ｆ１３ＤＨ （１００Ｂ３Ｈ） ＝０Ｆ８ＢＢＨ

（２７０００Ｈ） ＝２７０Ｈ （３９０００Ｈ） ＝３９Ｈ （１６０ＢＢＨ） ＝０ （１００ＡＡＨ） ＝２

（１Ａ０００Ｈ） ＝１ＡＨ

试判断下列各指令是否正确， 对于正确的指令，请填写表 ２．１ 中的后面各项。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 指 令 | | 目的操作数地址 | 本指令执行前后目的操作数的值 | | 是否  正确 |
| 执行前 | 执行后 |
| ~~MOV~~ | ~~AX, 3~~ |  |  |  |  |
| SUB | WORD PTR [AX], 3 |  |  |  | 错误 |
| ADD | WORD PTR [EAX], 3 | 100AAH | 2H | 5H | 正确 |
| ~~ADD~~ | ~~BH, 2~~ |  |  |  |  |
| ~~SUB~~ | ~~EBP, 2~~ |  |  |  |  |
| SUB | [CX], DX |  |  |  | 错误 |
| ~~ADD~~ | ~~DI, 2~~ |  |  |  |  |
| ~~SUB~~ | ~~SI, 10~~ |  |  |  |  |
| MOV | [BX], BX | 100BBH | 200H | 2BBH | 正确 |
| ~~MOV~~ | ~~[DI], DX~~ |  |  |  |  |
| MOV | ES： [DX] , BX |  |  |  | 错误 |
| MOV | ES： [SI], DS |  |  |  | 错误 |
| ~~MOV~~ | ~~DX, SS~~ |  |  |  |  |
| ~~MOV~~ | ~~EAX, EIP~~ |  |  |  |  |
| ~~SUB~~ | ~~2[DX], AX~~ |  |  |  |  |
| ADD | 500H[BP], CX | 35500H | 355H | 3CCH | 正确 |
| ~~SUB~~ | ~~[SI-300H], AX~~ |  |  |  |  |
| ~~MOV~~ | ~~[AX+2], BX~~ |  |  |  |  |
| MOV | [DI + 1000H], SI | 17000H | 0F13DH | 4000H | 正确 |
| ~~MOV~~ | ~~[CX-100H], AX~~ |  |  |  |  |
| ~~MOV~~ | ~~[DX+60], AX~~ |  |  |  |  |
| MOV | -8[BX], CX | 100B3H | 0F8BBH | 0CCH | 正确 |
| ~~MOV~~ | ~~ES： 1000[DI], BP~~ |  |  |  |  |
| MOV | [BP + SI], DX | 39000H | 39H | 0DDH | 正确 |
| ~~MOV~~ | ~~[DI+SI], DX~~ |  |  |  |  |
| MOV | [EDI + ESI], DX |  |  |  | 错误 |
| MOV | [BX + DI] , 10H | 160BBH | 0 | 10H | 正确 |
| ~~MOV~~ | ~~[BX + DI] , DX~~ |  |  |  |  |

2.4..386

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

DATA SEGMENT USE16

BUF1 DB 0,1,2,3,4,5,6,7,8,9

BUF2 DB 10 DUP(0)

BUF3 DB 10 DUP(0)

BUF4 DB 10 DUP(0)

DATA ENDS

CODE SEGMENT USE16

ASSUME CS:CODE,DS:DATA,SS:STACK

START: MOV AX,DATA

MOV DS,AX

MOV ESI,0

MOV ECX,10

LOOPA: MOV AL,BUF1[ESI]

MOV BUF2[ESI],AL

INC AL

MOV BUF3[ESI],AL

ADD AL,3

MOV BUF4[ESI],AL

INC ESI

DEC ECX

JNZ LOOPA

MOV AH,4CH

INT 21H

CODE ENDS

END START

3.1

