**BÀI TẬP CẤU TRÚC MÁY TÍNH VÀ HỢP NGỮ**

1. Dòng sau đây trình bày nội dung cùa một phần bộ nhớ trong trình debug:

DS:00 00 07 00 EC FF 00 00 46 41

Giả sử rằng 4 vị trí thấp nhất được trình bày trên mã hóa 2 số nguyên có dấu kích thước từ (word) dạng bù 2, tìm giá trị thập phân của 2 số nguyên, giải thích?  
***Giải:***

**DS:0000 07 00 EC FF 00 00 46 41**

thấp cao thấp cao

***2 số đó là 0007 và FFEC***

***0007 = 7***

***FFEC =***

**1111 1111 1110 1100**

**Bù 1: -000 0000 0001 0011**

**+ 1**

**-------------------------------**

**-000 0000 0001 0100 = - 20**

1. Giả sử rằng 2 lệnh PUSH BX, PUSH CX được thực hiện (theo thứ tự) ngay trước khi debug hiện thị nội dung sau:

AX= 3520 BX =00FF CX =1234 DX= FFE2 SP= FF00 BP =0000 SI=0000 DI=0000 DS=1000 ES =0FF0 SS =12EB CS= 1F76 IP= 010A NV UP EI PL NZ NA PO NC 1F76:010A 58 POP AX

1. Cho biết nội dung mới của thanh ghi CS, IP, SS, SP ngay sau khi lệnh kế được thực hiện

***CS mớii= CS cũ=1F76h***

***IP mới= PI cũ + độ dài lệnh=010Ah+010Bh***

***SS mới=SS cũ=12Ebh***

***SP mới=SP cũ+2=FF00h+2=FF02h***

1. Vẽ sơ đồ trình bày rằng làm thế nào đỉnh stack thay đổi như kết quả của việc thực thi lệnh kế, trình bày rõ địa chỉ segment: offset và nội dung (dạng số Hex) cho mỗi vị trí nhớ có liên quan, bao gồm cả đỉnh stack cũ và mới trong sơ đồ.

Satck phụ thuộc và segment SS:SP

12EB:FF00 34 <-đỉnh Stack hiện tại

12EB:FF01 12

12EB:FF02 FF<-đỉnh stack tương lai

12EB:FF03 00

12EB:FF04 <-đỉnh Stack quá khứ

12EB:FF05

Khi dùng PUSH BX giảm đi 2 (từ 04 xuống 02), dùng PUSD CX giảm đi 2 (từ 02 xuống 00)

Dùng POP AX ra đỉnh stack tương lai

1. Tìm và mô tả lỗi trong mỗi lệnh assembly 80x86 (có thể không có , có thể có 1, 2 ,3 lỗi) và hãy sửa lại các lỗi cho đúng, có thể sử dung, 2, … lệnh nếu cần

Giả sử cho các khai báo sau:

.data

mybyte db 41

theirbyte db ?

Mychars db “hubba dubbah”

1. MOV AX,12H ***-> Đúng***
2. MOV DS, 0123H

***-> SAI vì không thể mov trực tiếp tham số vào thanh ghi phân đoạn…***

***Sửa MOV BX, 0123H***

***MOV DS, BX***

1. ADD mybyte, 7200h ***-> sai Không thể cộng hai byte vào biến 1 byte***

Sửa ADD Word PTR mybyte,7200H

1. XCHG mybyte, mychars **-> không thể hoán chuyển trực tiếp từ Mem sang mem**
2. Cho AX = 012316 và BL = 1016. Tính nội dung mới của AX sau khi thực hiện lệnh DIV(MOD : chia het) BL?

***0123H =0000000100100011=291***

***10H = 000000000001000=16***

***DIV BL; AX:BL=0123h:10h***

***Thuong AL=12h,AX=0312H***

***Du AH=03h***

5. Cho biết kết quả sau khi thực hiện chuỗi lệnh sau (AH, AL = ?), giả sử rằng lúc đầu AL chứa 32H (là mã ASCII của số 2), BL chứa 3416 (mã ASCII của số 4) và AH = 0

***a. ADD AL, BL***

***AL=32H = 0000 0000 0011 0010 = 50***

***BL=34H= 0000 0000 0011 0100=52***

***ADD AL, BL = 0000 0000 0110 0110 (AL =66H)***

***b.AAA***

***AL=66h=0110 0110<9***

***🡺 AL=0000 0110= 06🡺 AX=0006h***

1. Nội dung ban đầu của AX, BL, ô nhớ SUM (kích thước từ) và cờ CF tương ứng là 1234H , ABH, 00CDH, 0H. Mô tả kết quả việc thực hiện chuỗi lệnh sau:

AX=1234h

BL=ABH

Ô nhớ SUM=00CDH

Cờ CF=0h

a.ADD AX,[SUM]= 1234+00CD=1301h

b. ADC BL, 05H= AB+05+0(cờ CF=0h)=B0h

C. INC Word PTR [SUM]

=> Chú ý:

Cộng 2 số hex 1234h + 00CDh

4 + D = 17 -> 17 -16 =1 ghi 1, nhớ 1

3 + C + 1(nhớ)= 16 -> 16 -16 =0 ghi 0, nhớ 1

2 + 1 = 3

1+ 0 = 1

* Kết quả 1301h

Trừ 2 số Hex : 1234h = 00CDh(mượn 16) nợ 1

4 – D = 4+ 16 – 13 = 7 (nợ 1)

3 – C= 3 + 16 – 12 - 1 = 6 (nợ 1)

2 – 0 -1 = 1

1 – 0= 1

* Kết quả 1167H

**Giải**

**AX =1234H**

**BL = ABH**

**SUM = 00CDH**

**CF=0H**

1. **ADD AX, [SUM]**

**AX = AX + [SUM] = 1234H + 00CDH = 1301H**

1. **ADC BL, 05H (cộng toán hạng nguồn , toán hạng đích và cờ nhớ)**

**BL = BL + 05H + CF =ABH + 05H + 0h= B0H**

**c. INC Word PTR [SUM] -> tăng SUM lên 1 -> SUM = 00CEH**

7. Cho biết nội dung của thanh ghi AX(Hex) sau khi thực hiện đoạn chương trình sau:

MOV AL, 05H ;AL= 05h

MOV BH, 09H ;BH =09h

MUL BH AX = AL \* BH = 002Dh =45 (nhân số không dấu, do 8 bit nên AX = AL \* thn8)

AAM; 45 /10 = 4 dư 5 -> AH =04h, AL =05h -> AX = 0405H

ADD AX, 3030H ;-> AX= 3435H

8. Bảng sau đây trình bày nội dung của 4 vị trí nhớ liện tục của data segment của chương trình của ngôn ngữ assembly viết trên 80x86 (lưu trữ theo kiểu little endian)

Địa chỉ Nội dung

1200:0400 00

1200:0401 01

1200:0402 FF

1200: 0403 FC

1. Cho biết nội dung (dạng số HEx) của các thanh ghi AX và BX ngay sau khi các lệnh sau dc thực hiện:

MOV AX,[0400] ->AX =0100H

MOV BX,[0402] ->BX = FCFFH

1. Giả sử nội dung của các thanh ghi AX và BX trong câu a mô tả các số nguyên có dạng bù 2, tính giá trị thập phân của chúng

***AX = 0100H = 0000 0001 0000 000 -> 256***

***BX =FCFFH = 1111 1100 1111 1111***

***Bù 1: -000 0011 0000 0000***

***+1= -000 0011 0000 0001 = -769***

9. Giải thích hoạt động được thực hiện bỡi mỗi lệnh sau:

a. MOV AX, 0110H **-> AX= 0110H**

b. MOV DI, AX **->DI = 0110H**

c. MOV BL, AL **->** **BL = 10H**

d. MOV [0110H], AX **-> [0111H, 0110H] = AX -> [0111H ]= 01h, [0110H]= 10h**

e. MOV [BX +DI], AX **-> [BX+DI+1, BX+DI] =AX -> [BX +0111H] =01h, [BX+1010h] =10h**

f. MOV [DI]+4, AX **-> [DI]+4 = AX -> [0115H] = 01H , [0114H] =10H**

g. MOV [BX][DI]+4, AX->**[BX+DI+4] = 0110H**

10.Viết một chuỗi lệnh khởi động thanh ghi ES với giá trị tức thời 1010H

***MOV AX, 1010H***

***MOV ES, AX***

11. Viết một lệnh lưu nội dung của thanh ghi ES vào bộ nhớ ở địa chỉ DS: 1000h

**MOV DS, [1000H]**

12. Tại sao lệnh MOV CL, AX bị lỗi khi hợp dịch

**Vì không thể chuyển biến 2 byte vào biến 1 byte**

13. Mô tả hoạt động được thực hiện bỗi mỗi lệnh sau:

a. XCHG AX, BX -> Đổi AX <-> BX

b. XCHG BX, DI -> Đổi BX <-> DI

c. XCHG [DATA], AX -> Đổi [DATA+1, DATA] <-> AX

d. XCHG [BX+DI ], AX -> Đổi [BX+DI+1, BX+DI] <->AX

14. Nếu thanh ghi BX = 0010H, DI = 0010H, DS = 1075H, vị trí ô nhớ nào bị trao đổi với AX khi mỗi lệnh trong bài 6 được thực thi

Ô nhớ BX bị thay đổi khi thực hiện các lệnh bài 6:

AX=0010

BX=0100

DS=1000

15. Giã sử rằng AX= 0010h, BX = 0100h, DS = 1000h, điều gì xảy ra nếu lệnh XLAT được thực thi?

**XLAT thực thi : MOV AL,[DS:BX +AL]= MOV AL, [0110]**

**AL= Nội dung của ô nhớ [0110]**

16. Viết một lệnh nạp AX từ địa chỉ 0200h và DS từ đĩa chỉ 0202h

LDS AX,[200]

17. Viết một lệnh cộng toán dạng tức thời 111Fh và cờ CF với nội dung của thanh ghi dữ liệu DX

ADC DX,111F

18. Viết một lệnh trừ nội dung của word trong ô nhớ được chỉ bởi thanh ghi nền BX và cờ CF khỏi thanh ghi AX

SBB AX,[BX]

19. Giả sử AX = 0123H, BL = 10H, nội dung mới của AX sẽ là bao nhiêu sau khi thực hiện lệnh DIV BL.

***0123H =0000000100100011=291***

***10H = 000000000001000=16***

***291 / 16 = 18 dư 3***

***-> Thương cất vào AL =12h (=18)***

***Dư cất vào AH = 03h***

***->AX = 0312H***

20. Viết 1 lệnh khi thực hiện che tất cả trừ bit 7 của nội dung thanh ghi DX.

XOR DX,00H

21. Viết 1 lệnh khi thực hiện che tất cả trừ bit 7 của từ dữ liệu được lưu trữ ở địa chỉ DS:0100H.

AND word PTR [100h],00h

22. Xác định quan hệ giữa nội dung cũ và mới của AX sau khi thực hiện các lệnh sau:

NOT AX -> BÙ 1

ADD AX, 1 -> CỘNG 1

-> Lấy số bù 2, 2 số đối nhau

23. Viết một lệnh đảo mức logic của bit trọng số cao nhất của giá trị trong byte cao của thanh ghi AX

XOR AX,8000H;

24. Giả sử các thanh ghi của 8088 và bộ nhớ như sau:

AX = 0000H BX= 0010H CX=0105H DX=1111H

SI= 0100H DI=0200H DS:100H=0FH DS:200H:22H

DS: 201H =44H DS:210H =55H DS:211H=AAH DS:220H=AAH

DS:221H=55H DS:400H =AAH DS:401H=55H CF =0

Tính kết quả tạo ra trong toán hạng đích sau khi thực hiện các lệnh sau:

1. SHL DX, CL

CL = 05H;DX=1111H=0001 0001 0001 0001;DICH TRAI 5 BIT;0010 0010 0010 0000=2220H

2220H

1. SHL Byte PTR [0400H],CL

CL = 05H;Byte PTR [0400] = AAH = 1010 1010DICH TRAI 5 BIT;0100 0000 = 40H

40H

1. SHR Byte PTR [DI], 1

Byte PTR [DI = 0200] = 22H=0010 0010;DICH TRAI 1 BIT;0100 0100 = 11H

11H

1. SHR Byte PTR [DI+BX],CL 02H
2. SAR Word PTR [BX+DI],1

Word PTR[BX+DI = 0210] = 55H = 0101 0101;DICH PHAI 1 BIT;0010 1010 = 002AH

002AH

Word PTR[BX+DI + 1 = 0211] = AAH = 1010 1010;DICH PHAI 1 BIT; 0101 0101 = 0022H;

0022H

1. SAR Word PTR [BX][DI]+10h,CL

CL=05H;Word PTR [BX][DI]+10h = 0010H+0200H+10H=0220H;AAH=1010 1010;DICH PHAI 5 BIT;0000 0101 = 0005H;

0005H

**25**.Cho biết kết quả của chương trình sau (AX=?)

Mem DW 8 ;Mem = 8

MOV AX, Mem ; AX = 8

SHL AX,1 ; AX = 2\*8

MOV BX, AX; BX = 2\*8

SHL AX,1; AX = 2\*2\*8

SHL AX,1 ; 2\*2\*2\*8

ADD AX, BX ; 0050H

**26**.Cho biết nội dung của biến Dem sau khi thực hiện đoạn chương trình sau

Mem DB ‘ACDE’

DB 50 DUP(0)

DB ‘ES0123SFS’

Đem DW 0

MOV CX, 63

MOV BX,OFFSET Mem

MOV Đem, 0

CONT:

CMP BYTE PTR[BX], ‘S’

JNZ Notmatch

INC Đem

Notmatch :

INC BX

LOOP CONT

**27**.AX=? (dạng DEC)

MOV AX,FFECH ; AX=-20

MOV BH,10 ; BH=10

IDIV BH ;AX:BH=-20:10→THƯƠNG AL=-2, DƯ AH=0→ AX=254 (AX=00FE)

IF(AL == 2)🡺AX = 0002H

IF(AL == -2) DOI LAI : -2 = 1000 0010

1111 1101

+ 1

1111 1110

* AL = FE => AX = 00FE => AX = 15\*16^1 + 14\*16^0

**28.**AX=? (DẠNG HEX)

MOV AX,3433H ; AX=3433H

ADD AL,AH ;AL=AL+AH=34H+33H=67H

AAA ;AH=00,AL=07→AX=0007H

**30.** MOV AL,-24 ;AL= -24=E8H

MOV AH,8 ;AH=8=08H

MOV BL,16 ;BL=16=10H

DIV BL ;AX:BL=08E8:10→Thương AL=8E, Dư AH=08h → AX=088EH

**BÀI TẬP GIẢI MÃ ĐỊA CHỈ**

1. Dùng EPROM 2764 (8k\*8) để ghép thành bộ nhớ 64k cho 8088 bắt đầu từ địa chỉ F0000h. Dủng mạch giải mã 74138 để thực hiện
2. Dùng EPROM 27128 (16k\*8) để ghép thành bộ nhớ 64k cho 8088 bắt đầu từ địa chỉ F0000h.
3. Ghép nối 8088 với SRAM 62256 (32k\*8) để được bộ nhớ 256kb, bắt đầu từ đại chỉ 00000h
4. Thiết kế mạch giải mả địa chỉ cho CPU có 16 bit địa chỉ, 8 bit dữ liệu:
5. Bảng bộ nhớ và I/O như sau:

0000H – 5FFFH :EPROM 24KB

6000H – 7FFFH: Ngoại vi 8KB

8000H- FFFFH: SRAM 32KB

1. 0000H – FFFFH : RAM 64KB được tạo từ các RAM 256 x 8
2. Thiết kế mạch giải địa chỉ cho CPU có 20 bit địa chỉ, 8 bit dữ liệu để có thế truy cập RAM 64 KB với vùng địa chỉ là 00000h – 0FFFFh. Biết RAM có số hiệu là HM62864 với các đường tín hiệu.:

* /CSI, CS2 là 2 đường chọn chip
* /0E cho phép xuất
* /WE cho phép ghi
* 16 đường địa chỉ A15….A0
* 8 đường dữ liệu I/O7….I/O0

Và CPU có các đường điều khiển bộ nhớ là /MEMW và /MEMR

1. Tưởng tượng dùng 1 DRAM 256 Mbit được tổ chức như 32 triệu địa chỉ các từ 8 bit để xây dựng 3 hệ thống nhớ. Xét hệ thống nhớ được xây dựng dùng các chip này như sau:
2. Xét hệ thống nhớ 256 MByte được tổ chức như 64 triệu địa chỉ các từ 32 bit. Tính:

* Số chip cần thiết cho 1 bank
* Số bank cho 1 hệ thống nhớ
* Kiểu giải mã được yêu cầu
* Số chip DRAM được yêu cầu

1. Xét hệ thống nhớ 8 GByte được tổ chức thành 1 triệu địa chỉ, mỗi địa chỉ là các từ 64 bit

* Số chip cần thiết cho 1 bank
* Số bank cho 1 hệ thống nhớ
* Kiểu giải mã được yêu cầu
* Số chip DRAM được yêu cầu

1. Thiết kế 1 hệ thống nhớ được tổ chức như 512k địa chỉ từ 16 bit dung các chip nhớ được tổ chức như 512K địa chỉ các từ 4 bit. Hỏi:

* Tổng số bit hệ thống nhớ lưu trữ
* Bao nhiêu chip DRAM nếu dung các chip có dung lượng là 64Kx4
* Trình bày thiết kế và vẽ hình

1. Cho chip SRAM 16 MBit được tổ chức thàng 2 triệu địa chỉ các từ 8 bit.
2. Giả sử các tế bào SRAM và chip SRAM là hình vuông. Dung tiếp cận tổ chức được trình bày cho DRAM trình bày những câu hỏi:

* Số cột
* Số từ trên 1 cột
* Loại giải mả cột được yêu cầu
* Số MUX được yêu cầu

1. Xét hệ thống nhớ SRAM 8 triệu địa chỉ từ 128 bit. Tính:

* Số chip cần thiết cho 1 bank
* Số bank cho 1 hệ thống nhớ
* Kiểu giải mã được yêu cầu
* Số chip DRAM được yêu cầu