

COMP LAB 2

190911112

ADIT LUHADIA

IT A

1. Multiply two 16 bit unsigned numbers by repetitive addition

DATA SEGMENT

MULTIPLICAND DW 1012H

MULTIPLIER DW 22H

RESULT DD ?

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS: DATA

START:

MOV AX, DATA

MOV DS, AX

MOV AX, 0

MOV DX, 0

MOV CX, MULTIPLIER

REPEAT_ADD::TO REPEATEDLY ADD THE NUMBER

ADD AX, MULTIPLICAND

ADC DX, 0

LOOP REPEAT_ADD

LEA SI, RESULT

MOV [SI], AX

MOV [SI+2], DX

MOV AH, 4CH

INT 21H

CODE ENDS

END START

The screenshot shows the DOSBox 0.74-3 interface. The title bar reads "DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...". The menu bar includes File, View, Run, Breakpoints, Data, Options, Window, and Help. The status bar at the bottom shows function key shortcuts: F1-Help, F2-Bkpt, F3-Mod, F4-Here, F5-Zoom, F6-Next, F7-Trace, F8-Step, F9-Run, F10-Menu. The main window is divided into several panes. The top-left pane shows assembly code for an 80486 CPU, with the instruction "int 21" at address 0023 highlighted. The top-right pane displays the current state of registers: ax=4C64, bx=0000, cx=0000, dx=0002, si=0004, di=0000, bp=0000, sp=FFFE, c=0, z=0, s=0, o=0, p=0, a=0, i=1, d=0. The bottom-left pane shows memory dumps for DS (starting at 0000) and SS (starting at 0000). The bottom-right pane shows the current instruction pointer (ip) at 0023 and the stack pointer (sp) at FFFE.

2. Convert four digit BCD number to HEX.

DATA SEGMENT

BCD DW 7865H

TEMP DW 4 DUP(?)

HEX DW ?

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS: DATA

START:

MOV AX, DATA

MOV DS, AX

MOV BX, BCD

MOV SI, 0

MOV CL, 04

GET_EACH_DIGIT: ;GETTING EACH INDIVIDUAL DIGIT AND STORING IN TEMP

MOV AX, BX

AND AX, 0FH

MOV TEMP[SI], AX

INC SI

INC SI

SHR BX, CL

CMP BX, 0

JNZ GET_EACH_DIGIT

LEA SI, TEMP

MOV BX, [SI]

MOV DI, 03H

MOV CX, 0AH

;MULTIPLYING EACH INDIVIDUAL DIGIT BY 1, 0AH, 0AH^2 AND 0AH^3

MULTIPLY_AND_ADD_DIGITS:

INC SI

INC SI

MOV AX, [SI]

MUL CX

ADD BX, AX

MOV AX, 0AH

MUL CX

MOV CX, AX

DEC DI

JNZ MULTIPLY_AND_ADD_DIGITS

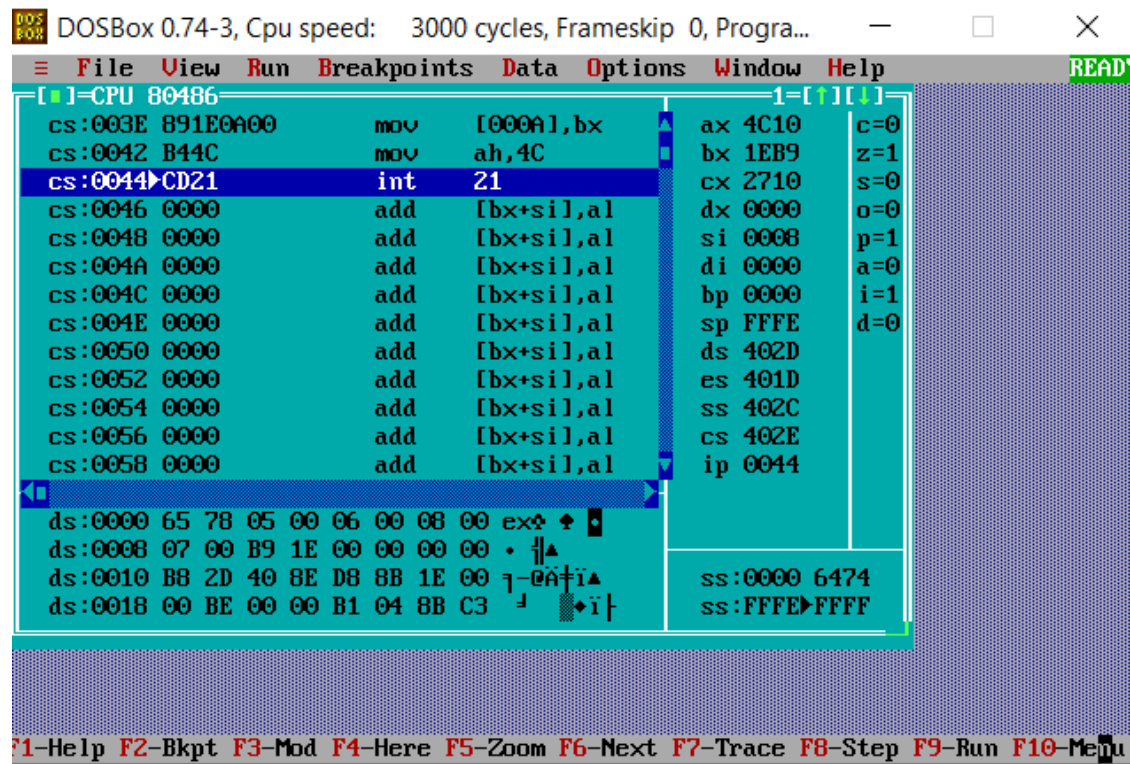
MOV HEX, BX

MOV AH, 4CH

INT 21H

CODE ENDS

END START



3. Convert two digit number from HEX to BCD.

DATA SEGMENT

HEX DB 0B8H

TEMP DB 3 DUP(0)

BCD DW ?

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS: DATA

START:

MOV AX, DATA

MOV DS, AX

MOV AL, HEX

MOV CL, 0AH

LEA SI, TEMP

REPEAT_DIV::REPEATEDLY DEVIDE THE NUMBER BY 0AH

MOV AH, 0

DIV CL

MOV[SI], AH

INC SI

CMP AL, 0

JNZ REPEAT_DIV

LEA SI, TEMP

MOV CL, 04

MOV AL, [SI+1]

ROR AL, CL

ADD AL, [SI]

MOV AH, [SI+2]

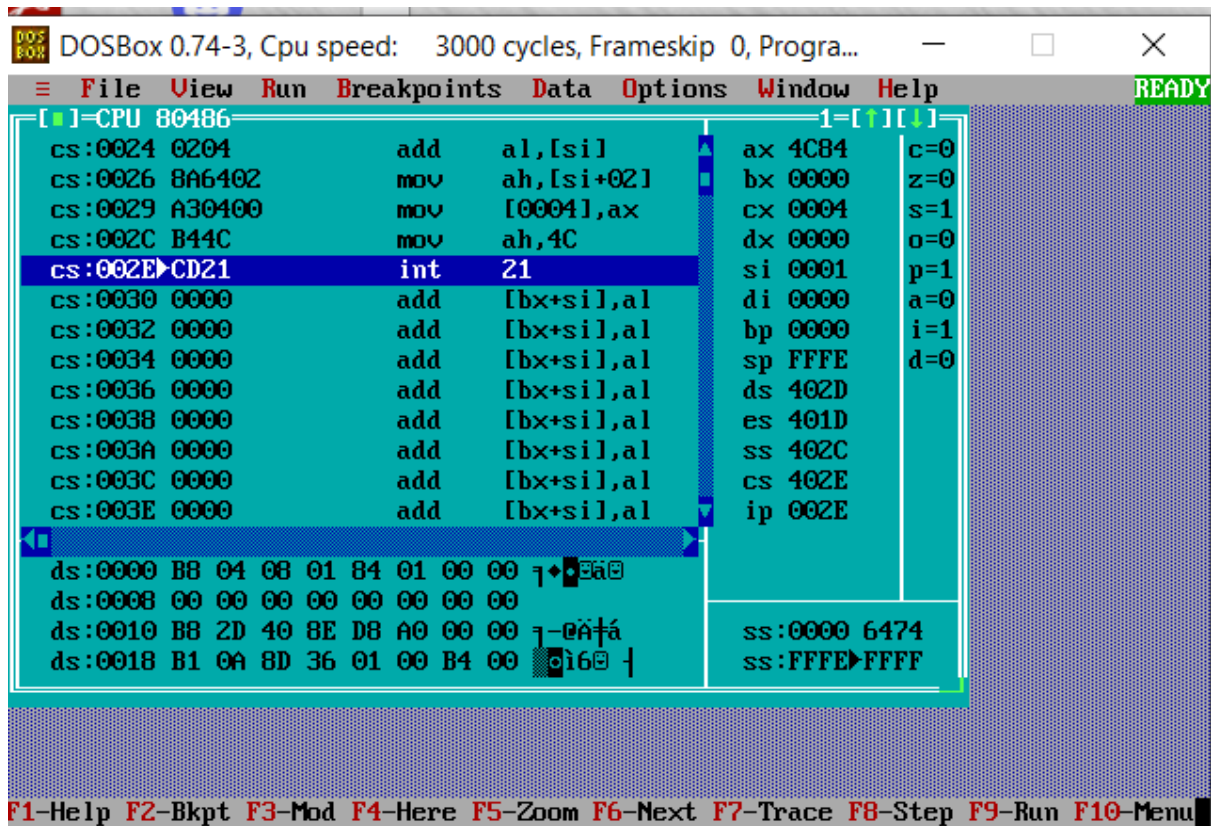
MOV BCD, AX

MOV AH, 4CH

INT 21H

CODE ENDS

END START



4. Convert four digit number from HEX to BCD.

DATA SEGMENT

HEX DW 0FFFFH

TEMP DB 5 DUP(0)

BCD DB 3 DUP(1)

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS: DATA

START:

MOV AX, DATA

MOV DS, AX

MOV AX, HEX

MOV CX, 000AH

LEA SI, TEMP

REPEAT_DIV::REPETEDLY DIVIDE NUMBER BY 000AH

MOV DX, 0

DIV CX

MOV[SI], DL

INC SI

CMP AX, 0

JNZ REPEAT_DIV

MOV BL, 02

LEA DI, BCD

MOV CL, 04

LEA SI, TEMP

PACK:

MOV AL, [SI+1]

ROR AL, CL

ADD AL, [SI]

INC SI

INC SI

MOV [DI], AL

INC DI

DEC BL

JNZ PACK

MOV AH, [SI]

MOV [DI], AH

MOV AH, 4CH

INT 21H

CODE ENDS

END START

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra... — □ ×

≡ File View Run Breakpoints Data Options Window Help READY

[]=CPU 80486 1=[↑][↓]

cs:0031 8805	mov	[di],al	ax 4C55	c=0
cs:0033 47	inc	di	bx 0000	z=1
cs:0034 FECB	dec	bl	cx 0004	s=0
cs:0036 75F0	jne	0028	dx 0006	o=0
cs:0038 8A24	mov	ah,[si]	si 0006	p=1
cs:003A 8825	mov	[di],ah	di 0009	a=0
cs:003C B44C	mov	ah,4C	bp 0000	i=1
cs:003E CD21	int	21	sp FFFE	d=0
cs:0040 0000	add	[bx+si],al	ds 402D	
cs:0042 0000	add	[bx+si],al	es 401D	
cs:0044 0000	add	[bx+si],al	ss 402C	
cs:0046 0000	add	[bx+si],al	cs 402E	
cs:0048 0000	add	[bx+si],al	ip 003E	

ds:0000 FF FF 05 03 05 05 06 35 40 40 45
ds:0008 55 06 00 00 00 00 00 00 U
ds:0010 B8 2D 40 8E D8 A1 00 00 7-0A+i
ds:0018 B9 0A 00 8D 36 02 00 BA 0 i6 ||

ss:0000 6474
ss:FFFE FFFF

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu