

Microprocessors and Assembly Language - Coding Questions

Write an assembly language program that adds two 8-bit numbers stored in registers and stores the result in a third register. Ensure proper usage of flags and comments. (3 marks) .1

Write a program in assembly language that moves data from memory to registers and then performs a bitwise AND operation. Explain how the operation affects the processor flags. (3 marks) .2

السؤال الأول 1

- استدعاء قيمة المتغير number1 من جزء data وتخزينها في المسجل al
- استدعاء قيمة المتغير number2 من جزء data وتخزينها في المسجل bl
- تنفيذ عملية add ولا يوجد تأثيرات على Carry Flag قيمته صفر

emulator: mycode8.exe

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers H L

AX	07	0B
BX	00	00
CX	00	61
DX	00	00
CS	0715	
IP	0008	
SS	0710	
SP	0040	
BP	0000	
SI	0000	
DI	0000	
DS	0714	
ES	0700	

0715:0008

07150:	B8 184	MOV AX, 00714h
07151:	14 020	MOV DS, AX
07152:	07 007	MOV AL, [00000h]
07153:	8E 142	MOV BL, [00001h]
07154:	D8 216	ADD AL, BL
07155:	A0 160	MOV [00002h], AL
07156:	00 000	NOP
07157:	00 000	NOP
07158:	8A 138	NOP
07159:	1E 030	NOP
0715A:	01 001	NOP
0715B:	00 000	NOP
0715C:	02 002	NOP
0715D:	C3 195	NOP
0715E:	A2 162	NOP
0715F:	02 002	NOP
07160:	00 000	NOP
07161:	90 144	NOP
07162:	90 144	NOP
07163:	90 144	NOP
07164:	90 144	NOP

original source code

```

01 .model small
02 .stack 64
03
04 .data
05
06
07 number1 db 11 ;Defining and initializing variables
08 number2 db 10
09 result db ?
10
11
12 .code ;start of code
13 main proc ; scop of main
14 mov ax,@data ;acsses of data value during invoknd of varial
15 mov ds,ax
16
17 mov al,number1 ;copy of number1 value into register
18 mov bl,number2 ;.....2.....
19 add al,bl ;add bl value into al register
20 mov result,al ; storing of al register into result
21
22
23
24 main endp
25 end main
26
27

```

flags

CF	0
ZF	0
SF	0
OF	0
PF	0
AF	0
IF	1
DF	0

analyse

emulator: mycode8.exe

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers H L

AX	07	0B
BX	00	0A
CX	00	61
DX	00	00
CS	0715	
IP	000C	
SS	0710	
SP	0040	
BP	0000	
SI	0000	
DI	0000	
DS	0714	
ES	0700	

0715:000C

07150:	B8 184	MOV AX, 00714h
07151:	14 020	MOV DS, AX
07152:	07 007	MOV AL, [00000h]
07153:	8E 142	MOV BL, [00001h]
07154:	D8 216	ADD AL, BL
07155:	A0 160	MOV [00002h], AL
07156:	00 000	NOP
07157:	00 000	NOP
07158:	8A 138	NOP
07159:	1E 030	NOP
0715A:	01 001	NOP
0715B:	00 000	NOP
0715C:	02 002	NOP
0715D:	C3 195	NOP
0715E:	A2 162	NOP
0715F:	02 002	NOP
07160:	00 000	NOP
07161:	90 144	NOP
07162:	90 144	NOP
07163:	90 144	NOP
07164:	90 144	NOP

original source code

```

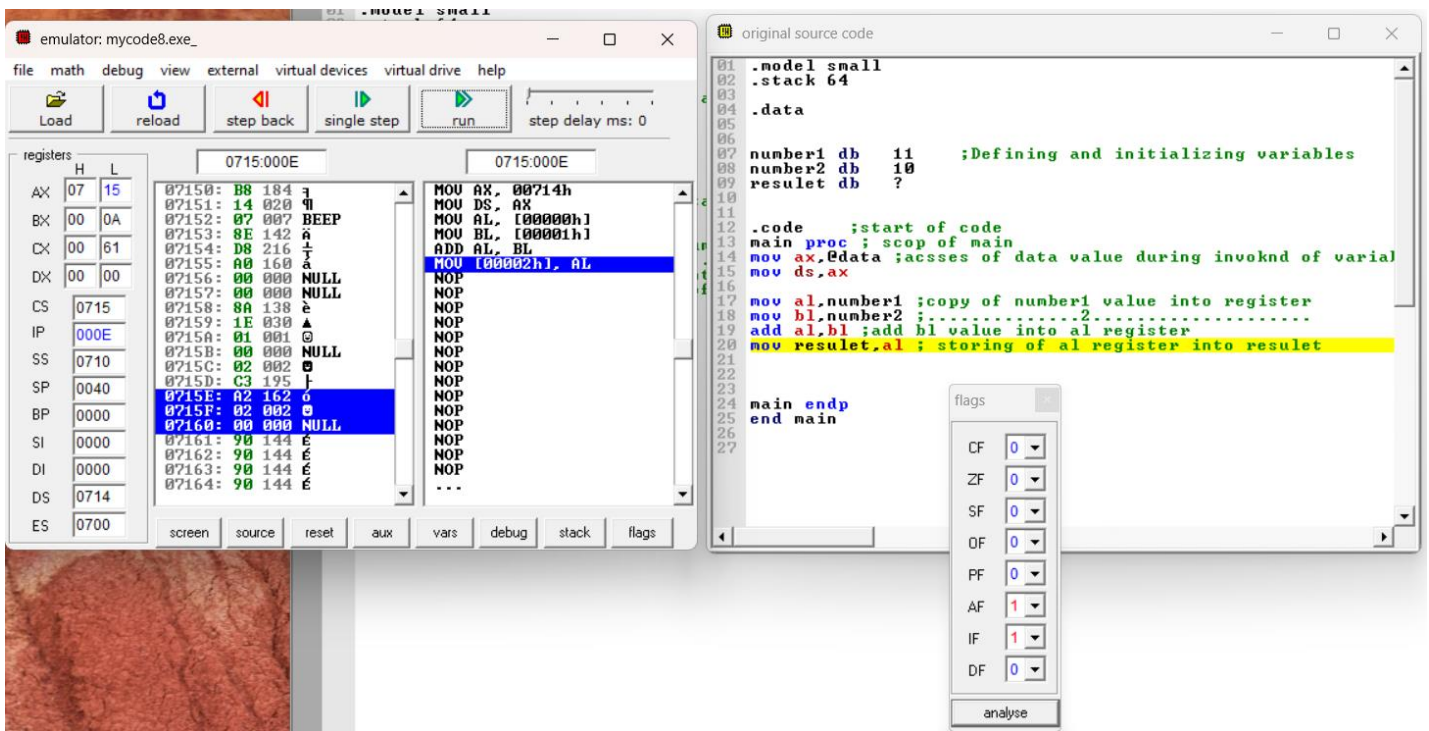
01 .model small
02 .stack 64
03
04 .data
05
06
07 number1 db 11 ;Defining and initializing variables
08 number2 db 10
09 result db ?
10
11
12 .code ;start of code
13 main proc ; scop of main
14 mov ax,@data ;acsses of data value during invoknd of varial
15 mov ds,ax
16
17 mov al,number1 ;copy of number1 value into register
18 mov bl,number2 ;.....2.....
19 add al,bl ;add bl value into al register
20 mov result,al ; storing of al register into result
21
22
23
24 main endp
25 end main
26
27

```

flags

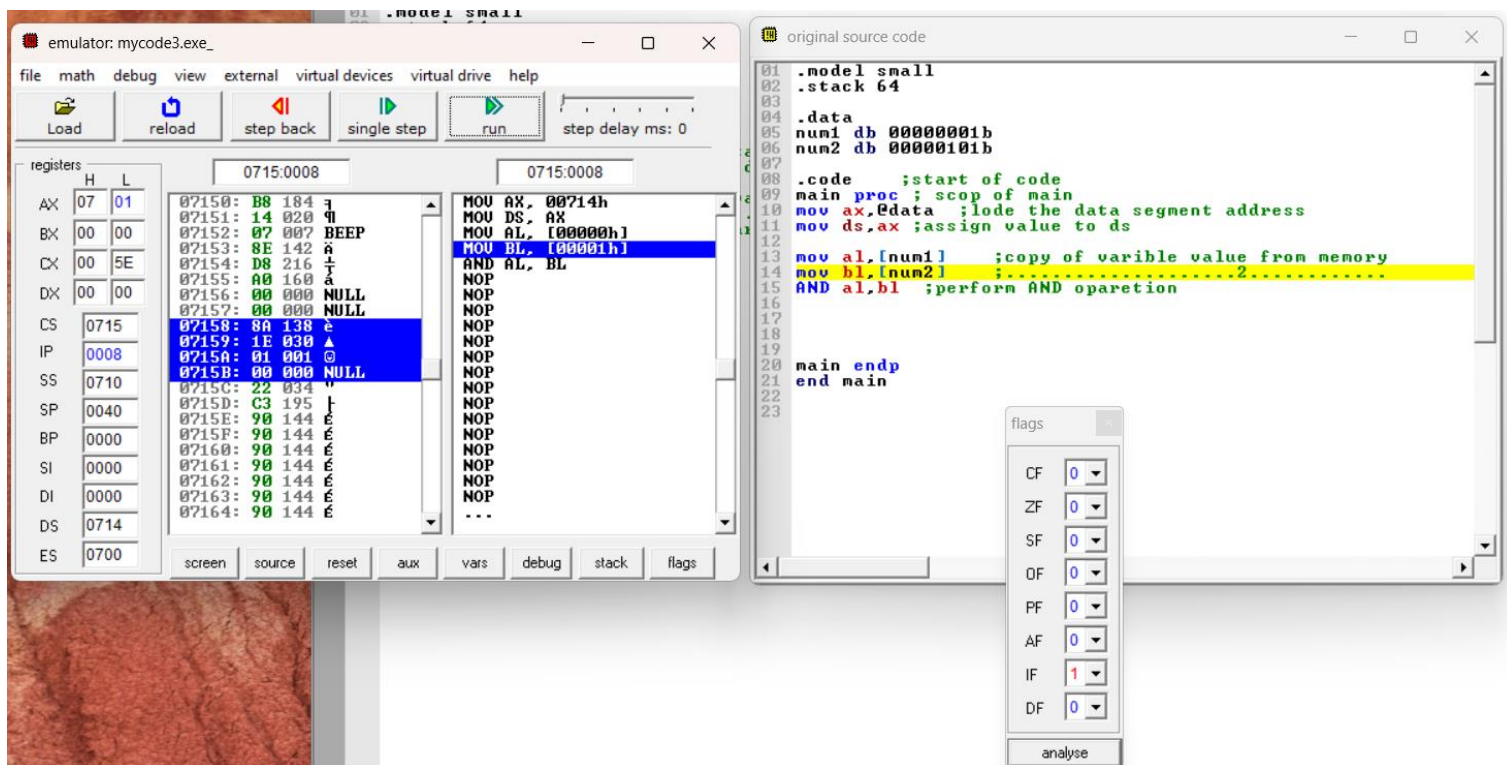
CF	0
ZF	0
SF	0
OF	0
PF	0
AF	0
IF	1
DF	0

analyse



السؤال الثاني2

- استدعاء قيمة المتغير num1 من جزء data وتخزينها في المسجل al
- استدعاء قيمة المتغير num2 من جزء data وتخزينها في المسجل bl
- تنفيذ عملية AND ولا يوجد تأثيرات على Carry Flag قيمته صفر



emulator: mycode3.exe_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	07	01
BX	00	05
CX	00	5E
DX	00	00
CS	0715	
IP	000C	
SS	0710	
SP	0040	
BP	0000	
SI	0000	
DI	0000	
DS	0714	
ES	0700	

0715:000C

07150:	B8 184	MOV AX, 00714h
07151:	14 020	MOV DS, AX
07152:	07 007	BEEP
07153:	8E 142	MOV AL, [00000h]
07154:	D8 216	MOV BL, [00001h]
07155:	A0 160	AND AL, BL
07156:	00 000	NOP
07157:	00 000	NOP
07158:	8A 138	NOP
07159:	1E 030	NOP
0715A:	01 001	NOP
0715B:	00 000	NOP
0715C:	22 034	NOP
0715D:	C3 195	NOP
0715E:	90 144	NOP
0715F:	90 144	NOP
07160:	90 144	NOP
07161:	90 144	NOP
07162:	90 144	NOP
07163:	90 144	NOP
07164:	90 144	NOP

screen source reset aux vars debug stack flags

original source code

```

01 .model small
02 .stack 64
03
04 .data
05 num1 db 00000001b
06 num2 db 00000101b
07
08 .code ;start of code
09 main proc ; scop of main
10 mov ax,0data ;lode the data segment address
11 mov ds,ax ;assign value to ds
12
13 mov al,[num1] ;copy of variable value from memory
14 mov bl,[num2] ;.....2.....
15 AND al,bl ;perform AND oparetion
16
17
18
19
20 main endp
21 end main
22
23

```

flags

CF	0
ZF	0
SF	0
OF	0
PF	0
AF	0
IF	1
DF	0

analyse

emulator: mycode3.exe_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	07	01
BX	00	05
CX	00	5E
DX	00	00
CS	0715	
IP	000E	
SS	0710	
SP	0040	
BP	0000	
SI	0000	
DI	0000	
DS	0714	
ES	0700	

0715:000E

07150:	B8 184	MOV AX, 00714h
07151:	14 020	MOV DS, AX
07152:	07 007	BEEP
07153:	8E 142	MOV AL, [00000h]
07154:	D8 216	MOV BL, [00001h]
07155:	A0 160	AND AL, BL
07156:	00 000	NOP
07157:	00 000	NOP
07158:	8A 138	NOP
07159:	1E 030	NOP
0715A:	01 001	NOP
0715B:	00 000	NOP
0715C:	22 034	NOP
0715D:	C3 195	NOP
0715E:	90 144	NOP
0715F:	90 144	NOP
07160:	90 144	NOP
07161:	90 144	NOP
07162:	90 144	NOP
07163:	90 144	NOP
07164:	90 144	NOP

screen source reset aux vars debug stack flags

original source code

```

01 .model small
02 .stack 64
03
04 .data
05 num1 db 00000001b
06 num2 db 00000101b
07
08 .code ;start of code
09 main proc ; scop of main
10 mov ax,0data ;lode the data segment address
11 mov ds,ax ;assign value to ds
12
13 mov al,[num1] ;copy of variable value from memory
14 mov bl,[num2] ;.....2.....
15 AND al,bl ;perform AND oparetion
16
17
18
19
20 main endp
21 end main
22
23

```

flags

CF	0
ZF	0
SF	0
OF	0
PF	0
AF	0
IF	1
DF	0

analyse