

21BDS0340

Abhinav Dinesh Srivatsa

Microprocessors and Microcontrollers Lab

Lab Task – I

Question 1

Code:

```
; addition
MOV A, #10H ; A = 10H
MOV R0, #20H ; A = 20H
ADD A, R0 ; A = A + R0 = 30H
MOV R0, A ; addition result stored in R0

; subtraction
MOV A, #20H ; A = 20H
MOV R1, #54H ; R1 = 54H
SUBB A, R1 ; A = A - R1 = -34H = CCH in 2's complement
MOV R1, A ; subtraction result stored in R1

; multiplication
MOV A, #0CFH ; A = CFH
MOV B, #10H ; R2 = 10H
MUL AB ; AB = A x B = CF0H
MOV R2, A ; MSB's of AB stored in R2
MOV R3, B ; LSB's of AB stored in R3

; division
MOV A, #54H ; A = 54H
MOV B, #20H ; B = 20H
DIV AB ; AB = A / B = Quotient - 2H, Remainder - 14H
MOV R4, A ; R4 = Quotient of A / B
MOV R5, B ; R5 = Remainder of A / B

END
```

Output:

Register	Value
[-] Regs	
r0	0x30
r1	0xcc
r2	0xf0
r3	0x0c
r4	0x02
r5	0x14
r6	0x00
r7	0x00
[-] Sys	
a	0x02
b	0x14
sp	0x07
sp_max	0x07
PC \$	C:0x001E
auxr1	0x00
dptra	0x0000
states	28
sec	0.00001680
psw	0x41

Manual Calculations:

Addition:

$$\begin{aligned} 10H + 20H &= 0001\ 0000 + 0010\ 0000 \\ &= 0011\ 0000 \\ &= \underline{30H} \end{aligned}$$

Subtraction:

$$\begin{aligned} 20H - 54H &= 0010\ 0000 - 0101\ 0100 \\ &= 0010\ 0000 + 10101011 + 1 \\ &= 1100\ 1100 \\ &= \underline{CCH} \end{aligned}$$

Multiplication:

$$\begin{aligned} CFH \times 10H &= 11001111 \times 0001\ 0000 \\ &= 0000\ 1100\ 1111\ 0000 \\ &= 0CF0 \\ &= \underline{0CF0H} \end{aligned}$$

Division:

$$54H / 20H = 01010100 / 00100000$$

$$\begin{array}{r} 10 \\ \overline{) 1010100} \\ \underline{100000} \\ 010100 \\ \underline{000000} \\ 10100 \end{array}$$

$$= \text{Quotient} = 2H$$

$$\text{Remainder} = 14H$$

Question 2

; registration number = 21BDS0340

; grouped as 02 1B D0 03 40 (S swapped for 0)

MOV A, #02H ; A = 02H

ADD A, #1BH ; A = 1DH

ADD A, #0D0H ; A = EDH

ADD A, #03H ; A = F0H

ADD A, #40H ; A = 130H, but too many bits, A = 30H, CY = 1

END

Output:

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x30
b	0x00
sp	0x07
sp_max	0x07
PC \$	C:0x000A
auxr1	0x00
dpnr	0x0000
states	5
sec	0.00000300
psw	0x80
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	1

Manual Calculations:

02, 1B, D0, 03, 40

$$02 + 1B = 1D$$

$$1D + D0 = ED$$

$$ED + 03 = F0$$

$$F0 + 40 = \underline{130H}$$

Question 3

Code:

```
; registration number = 21BDS0340 (ignoring S)
; values = 2, 1, B, D, 0, 3, 4, 0
; moving values into bank 0
```

```
MOV R0, #02H
```

```
MOV R1, #01H
```

```
MOV R2, #0BH
```

```
MOV R3, #0DH
```

```
MOV R4, #00H
```

```
MOV R5, #03H
```

```
MOV R6, #04H
```

```
MOV R7, #00H
```

```
; pushing into stack, bank 1
```

```
PUSH 00H
```

```
PUSH 01H
```

```
PUSH 02H
```

```
PUSH 03H
```

```
PUSH 04H
```

```
PUSH 05H
```

```
PUSH 06H
```

```
PUSH 07H
```

```
END
```

Output:

sp	0x07
sp_max	0x07

After 1st push: R0

sp	0x08
sp_max	0x08

After 2nd push: R1

sp	0x09
sp_max	0x09

After 3rd push: R2

sp	0x0a
sp_max	0x0a

After 4th push: R3

sp	0x0b
sp_max	0x0b

After 5th push: R4

sp	0x0c
sp_max	0x0c

After 6th push: R5

sp	0x0d
sp_max	0x0d

After 7th push: R7

sp	0x0e
sp_max	0x0e

After 8th push: R8

sp	0x0f
sp_max	0x0f

Question 4

Code:

```
; registration number = 21BDS0340 (ignoring S)  
; values = 2, 1, B, D, 0, 3, 4, 0  
; moving values into bank 1, stack
```

```
MOV 08H, #02H  
MOV 09H, #01H  
MOV 0AH, #0BH  
MOV 0BH, #0DH  
MOV 0CH, #00H  
MOV 0DH, #03H  
MOV 0EH, #04H  
MOV 0FH, #00H
```

```
; settings stack pointer to 0FH
```

```
MOV SP, #0FH
```

```
; popping values into bank 0, current bank
```

```
POP 07H  
POP 06H  
POP 05H  
POP 04H  
POP 03H  
POP 02H  
POP 01H  
POP 00H
```

END

Output:

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x0f
sp_max	0x0f

After 1st pop: into R7

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x0e
sp_max	0x0f

After 2nd pop: into R6

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x04
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x0d
sp_max	0x0f

After 3rd pop: into R5

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x03
r6	0x04
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x0c
sp_max	0x0f

After 4th pop: into R4

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x03
r6	0x04
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x0b
sp_max	0x0f

After 5th pop: into R3

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x0d
r4	0x00
r5	0x03
r6	0x04
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x0a
sp_max	0x0f

After 6th pop: into R2

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x0b
r3	0x0d
r4	0x00
r5	0x03
r6	0x04
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x09
sp_max	0x0f

After 7th pop: into R1

Register	Value
Regs	
r0	0x00
r1	0x01
r2	0x0b
r3	0x0d
r4	0x00
r5	0x03
r6	0x04
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x08
sp_max	0x0f

After 8th pop: into R0

Register	Value
Regs	
r0	0x02
r1	0x01
r2	0x0b
r3	0x0d
r4	0x00
r5	0x03
r6	0x04
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x07
sp_max	0x0f

Question 5

Code:

```
; registration number = 21BDS0340 (ignoring S)
; values = 2, 1, B, D, 0, 3, 4, 0
; moving values into bank 0
```

```
MOV R0, #02H
MOV R1, #01H
MOV R2, #0BH
MOV R3, #0DH
MOV R4, #00H
MOV R5, #03H
MOV R6, #04H
MOV R7, #00H
```

```
; pushing into stack, bank 1
```

```
PUSH 00H
PUSH 01H
PUSH 02H
PUSH 03H
PUSH 04H
PUSH 05H
PUSH 06H
PUSH 07H
```

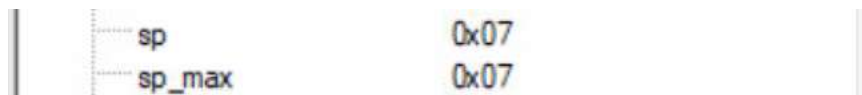
; popping from stack into R0-R7, bank 0

```
POP 00H
POP 01H
POP 02H
POP 03H
POP 04H
POP 05H
POP 06H
POP 07H
```

END

Output:

Push Operations:



After 1st push



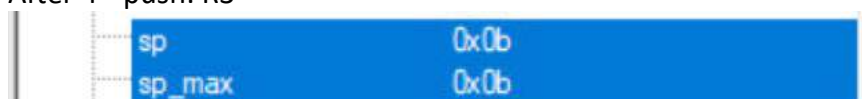
After 2nd push



After 3rd push: R2



After 4th push: R3



After 5th push: R4

sp	0x0c
sp_max	0x0c

After 6th push: R5

sp	0x0d
sp_max	0x0d

After 7th push: R7

sp	0x0e
sp_max	0x0e

After 8th push: R8

sp	0x0f
sp_max	0x0f

Pop Operations:

After 1st pop:

sp	0x0e
sp_max	0x0f

After 2nd pop:

sp	0x0d
sp_max	0x0f

After 3rd pop:

sp	0x0c
sp_max	0x0f

After 4th pop:

sp	0x0b
sp_max	0x0f

After 5th pop:

sp	0x0a
sp_max	0x0f

After 6th pop:

sp	0x09
sp_max	0x0f

After 7th pop:



After 8th pop:

