Computer Architecture Assignment 1-2024UCS0087 Dashpreet Singh

11 August 2025 23:46

Q1: A = 1101 , B = 0110

$$A+B = \begin{array}{r} 1101 \\ +0110 \\ \hline 10011 \end{array} = 10011$$

→ here, the lower bits are 0011, with a carry out of 1.

Q2 →

CX =	1000	1101	0110	1001
DX =	0010	1111	1001	1011

CX − DX ⇒

0111	0122	2012	0122
0010	1111	1001	1011
0101	1101	1100	1110

since, CX > DX, no borrow was required ∴ CF = 0

OF = 1, as large −ve number + a smaller +ve results in a +ve

Q3 → AH = 0x55 = 01010101

BH = 0xAA = 10101010

AH ^ BH = 11111111

Zero Flag = 0 (result ≠ 0)

Parity Flag = 1 (even one)

Q4 → SI = 0x7D8F

DI = 0x7D00

CMP SI, DI

= SI − DI

= 0x8F

CF : 0 (No borrow)

OF : 0 (+ve operands give +ve ans.)

OF : 0 (+ve operands give +ve ans.)

Q5 → CL = 0111 1001

DL = 1000 0101

CL OR DL = 1111 1101

CL will have 111 1101

Zero Flag : 0

Sign Flag : 1

Q6: A = 1011 1101

B = 0110 0101

A - B - Borrow (1)

(1011 1101 - 1) - 0110 0101

= 1011 1100 - 0110 0101

= 0101 0111

CF : 0, Now there's no borrow

OF : 1 (-ve - +ve → +ve)

Q7: A = 1101 0110

B = 1010 1101

A + B + Carry (1)

= 1101 0110 + 1010 1101 + 1

= 11000 0100

8bits result : 1000 0100

Carry out = 1

Q8: AX = 0x1234

BX = 0x5678

```
  1 1 1 2 1
  1 2 3 4
  5 6 7 8
```

→ CF = 1

→ OF = 1

```
  9 1 A 0
  7 f 6 C x
  6 D 3 8 x x
  5 B 0 4 x x x
-----------------
→ 6 2 6 0 0 6 0   →  the number is 0x06260060
```

splitting it between AX and DX

AX = 0x0060

DX = 0x0626

Q9 → AH = 0xAB = 1010 1011

BH = 0x5C = 0101 1100

AH AND BH = 0000 1000

AH will have 0x08

ZF = 0

PF = 0 (odd)

Q10 AX = 0xF0F0 = 1111 0000 1111 0000

BX = 0x0F0F = 0000 1111 0000 1111

AX ^ BX = 1111 1111 1111 1111

AX will have 0xFFFF

ZF = 0

PF = 1 (8 ones → even)

Q11→ MOV AX, [SI]

SI = 0x 2000

Segment address = 0x 3000

Now, physical Address = (Segment << 4) + offset.

= 0x3000 << 4 + 0x 2000

= 0x 3000 x 0x10 + 0x2000

= 0x 30000 + 0x 2000

= 0x 3200

⇒ Physical address = 0x 3200

Q12 → Segment address = 0x 1000

Q12 → Segment address = 0x 1000
Offset address = 0x 3000
Physical address = 0x1000 x 0x 10 + 0x 3000
= 0x 13000

Q13 → Bx (offset) = 0x1234
Since, segment address is not explicitly given
∴ by default, DS segment register is used.
Physical address = (DS << 4) + offset
= 0x0000 x 0x 10 + 0x1234
= 0x 1234

Q14 → Segment address = 0x 2000
offset = Bx + 0x 10
= 0x 1010
Physical address = 0x 2000 << 4 + 0x 1010
= 0x 2000 x 0x10 + 0x1010
= 0x 21010

Q15 → Segment Address = 0x 2000
Effective Address = BP + 20H = 0x 3020
Physical Address = 0x 4000 x 0x10 + 0x 3020
= 0x 43020

Q16 →
```
MOV AL, [3000H]
INC AL
MOV [3000H], AL
```

Q17 → Let the memory locations of 400Ah and 4001h have some initial values

```
MOV AL, [4000H]
MOV BL, [4001H]
MOV [4000H], BL
MOV [4001H], AL
```

Q18 → Let the five continuous memory locations after 7000H have some values.

```
MOV  SI, 7000H
MOV  BL, 5
LOOP_SUM:
   ADD  AL, [SI]
   ADC  AL, 0
   INC  SI
   DEC  BL
   JNZ  LOOP_SUM
```

Q19 →

```
MOV  SI, 8000H
MOV  BL, 5
MOV  CL, 1
LOOP:
   MOV  [SI], CL
   INC  CL
   INC  SI
   DEC  BL
   JNZ  LOOP
```

Q20 →

1. Little Endian : 0x6665
   Big Endian : 0x6566
   diff : 6665 (5th carry)
   6566
   0 0ff
   = 255 → Satisfies

2. Little Endian : 0x0001
   Big Endian : 0x0100
   Here, Little Endian < Big Endian
   ∴ diff. would be -ve ⇒ fails.

3. Little Endian : 0x4243
   Big Endian : 0x4342
   Here, Little Endian < Big Endian
   ∴ diff. would be -ve ⇒ fails.

4. Little Endian : 0x0100
   Big Endian : 0x0001
   diff : 0100
   0001
   ff
   = 255 → Satisfies