Assignment # 01

Q#1 Decimal:

(a) 00110101

$$(00110101)_2 = 0\times2^7+0\times2^6+1\times2^5+1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0$$
$$= 0+0+32+16+0+4+0+1$$
$$= (53)_{10}$$

(b) 10010110

$$(10010110)_2 = 1\times2^7+0\times2^6+0\times2^5+1\times2^4+0\times2^3+1\times2^2+1\times2^1+0\times2^0$$
$$=128+0+0+16+0+4+2+0$$
$$=(150)_{10}$$

(c) 11001100

$$(11001100)_2 = 1\times2^7+1\times2^6+0\times2^5+0\times2^4+1\times2^3+1\times2^2+0\times2^1+0\times2^0$$
$$=128+64+0+0+8+4+0+0$$
$$=(204)_{10}$$

Q#2 Sum:

(a) 10101111 + 11011011

$$\begin{array}{r} 10101111 \\ +\ 11011011 \\ \hline 110001010 \\ \hline \end{array}$$

(b) 10010111 + 11111111

$$\begin{array}{r} 1\,0\,0\,1\,0\,1\,1\,1 \\ +\;1\,1\,1\,1\,1\,1\,1\,1 \\ \hline 1\,1\,0\,0\,1\,0\,1\,1\,0 \\ \hline \end{array}$$

(c) 01110101 + 10101100

$$\begin{array}{r} 0\,1\,1\,1\,0\,1\,0\,1 \\ +\;1\,0\,1\,0\,1\,1\,0\,0 \\ \hline 1\,0\,0\,1\,0\,0\,0\,0\,1 \\ \hline \end{array}$$

Q#3 Subtract:

$$\begin{array}{r} 0\,0\,0\,0\,\cancel{1}\,\overset{10}{\cancel{1}}\,10\;1 \\ +\;0\,0\,0\,0\,0\,1\,1\,1 \\ \hline 0\,0\,0\,0\,0\,1\,1\,0 \\ \hline \end{array}$$

Q#4

(a) WORD	8-Bits
(b) DOUBLEWORD	32-Bits
(c) QUADWORD	64-Bits
(d) DOUBLE QUADWORD	128-Bits

Q#5 (a) 4095

$$\begin{array}{r|l} 2 & 4095 \\ \hline 2 & 2047-1 \\ \hline 2 & 1023-1 \\ \hline 2 & 511-1 \\ \hline 2 & 255-1 \\ \hline 2 & 127-1 \\ \hline 2 & 63-1 \\ \hline 2 & 31-1 \\ \hline 2 & 15-1 \\ \hline 2 & 7-1 \\ \hline 2 & 3-1 \\ \hline & 1-1 \end{array}$$

$= (1111\ 1111\ 1111)_2$

Hence 12 bits are required to represent $(4095)_{10}$.

(b) 65534

```
2 | 65534
2 | 32767 − 0
2 | 16383 − 1
2 |  8191 − 1
2 |  4095 − 1
2 |  2047 − 1
2 |  1023 − 1
2 |   511 − 1
2 |   255 − 1
2 |   127 − 1
2 |    63 − 1
2 |    31 − 1
2 |    15 − 1
2 |     7 − 1
2 |     3 − 1
  |     1 − 1
```

= (1111 1111 1111 1110)

Hence minimum of 16 bits are required to represent $(65534)_{10}$.

(c) 42319

```
2 | 42319
2 | 21159 − 1
2 | 10579 − 1
2 |  5289 − 1
2 |  2644 − 1
2 |  1322 − 0
2 |   661 − 0
2 |   330 − 1
2 |   165 − 0
2 |    82 − 1
2 |    41 − 0
2 |    20 − 1
2 |    10 − 0
2 |     5 − 0
2 |     2 − 1
  |     1 − 0
```

$= (1010\ 0101\ 0100\ 1111)_2$

Hence minimum of 16 bits are required to represent $(42319)_{10}$

Q#6 (a) 0011 0101 1101 1010

$(0011)_2 = (3)_{10} = (3)_{16}$

$(0101)_2 = (5)_{10} = (5)_{16}$

$(1101)_2 = (8+4+1)_{10} = (13)_{10} = (D)_{16}$

$(1010)_2 = (8+2)_{10} = (10)_{10} = (A)_{16}$

$= (35DA)_{16}$

Scanned with CamScanner

(b) 1100 1110 1010 0011

$(1100)_2 = (8+4)_{10} = (12)_{10} = (C)_{16}$
$(1110)_2 = (8+4+2)_{10} = (14)_{10} = (E)_{16}$
$(1010)_2 = (8+2)_{10} = (10)_{10} = (A)_{16}$
$(0011)_2 = (1+2)_{10} = (3)_{10} = (3)_{16}$

$(CEA3)_{16}$

(C) 1111 1110 1101 1011

$(1111)_2 = (15)_{10} = (F)_{16}$
$(1110)_2 = (14)_{10} = (E)_{16}$
$(1101)_2 = (13)_{10} = (D)_{16}$
$(1011)_2 = (11)_{10} = (B)_{16}$

$(FEDB)_{16}$

Q#7 Status Flags:

- Carry flag
- Zero flag
- Sign flag
- Overflow flag
- Parity flag

Q#8

Carry Flag.

Q#9

Overflow Flag.

Q#10

Sign Flag

Q#11

Floating-point unit.

Q#12

```
 .386
 .model flat , stdcall
 .stack 4096
 ExitProcess PROTO, dwExitCode: DWORD
 .data
 .code
 main PROC

mov   EAX, 20
mov   EBX, 15
mov   ECX, 10
mov   EDX, 5
add   EAX, EBX
add   ECX, EDX
sub   EAX, ECX

INVOKE  ExitProcess , 0
main   ENDP
END  main
```

Q#13

(a) FFFF8002h (MSB > 7) –ve integer
(b) 00004321h (MSB< 7) +ve integer

Q#14

After the first line executes 32-bit register EAX stores 1002FFFFh. When we increment AX the value in AX changes from FFFFh to 0000h. So the final value is 10020000h.

Q#15

After the first line executes 32-bit register EAX stores 30020000h. When we decrement AX the value in AX changes from 0000h to FFFFh. So the final value is 3002FFFFh.

Q#16

The negative sign reverses the sign of a number to its 2-complement. After first line it stores 1002FFFFh in EAX register. When we neg AX then it converts FFFFh to its 2's complement which is 0001h. So the final value is 10020001h.

Q#17

The parity flag indicates whether or not an even number of 1 bits occur in the least significant byte of the destination operand. Since the value of AL register is 03h (03d) which represent 0011B. The least significant bit is 1, so parity flag will be set to 1.

Q#18

The value of EAX will be FFFFFFFFh. As it generates a negative result so the sign flag will be set to 1.

Q#19

AL is the valid range in the first expression. The sign flag is also triggered. Despite the result being positive in second line, the sign flag would still be 'on'. Thus, it tells you that the result must be an overflow and not valid.

Q#20

The value of EAX will be 00035678h.

Q#21

The value of EAX will be 12341237h.

Q#22

```
.386
.model flat, stdcall
.stack 4096
ExitProcess PROTO, dw ExitCode: DWORD
```

```
.data
bigEndian BYTE 12h,34h,56h,78h
littleEndian DWORD ?
.code
main PROC
mov AL,[bigEndian+3]
mov BYTE ptr [littleEndian],Al

mov AL,[bigEndian+2]
mov BYTE PTR [littleEndian+1],Al

mov AL,[BigEndian+1]
more BYTE PTR [littleEndian+2],AL

mov AL,[BigEndian]
more BYTE PTR [LittleEndian+3],AL

INVOKE ExitProcess,0
main ENDP
END main
```