Homework 2

EET 340

1. Convert Decimal value to binary and then convert them to hexadecimal value

a. 4510

4510 = 001011012 = 0x2D16

b. 2210

2210 = 000101102 = 0x1616

2. What will be the value of X1 after running the following instruction: `LSL X1, X2, #2`. Assume that X2 = 4. (Show the steps of calculation).

If X2 = 4,

`LSL X1, X2, `*`#2   ; X2 = 4, X1 = X2*2^2, X1 = 4*4, X1 = 16`*

X1 = 1610

3. What will happen to X2 after running the following instruction: `LDUR X2, [X5, #0]`. Assume that X5 = 0X8000000000004000 and locations 0X8000000000004000 through 0X8000000000004007 contain 0X00, 0X00, 0X00, 0X00, 0X00, 0X00, 0X02, and 0X23, respectively.

X2 will load the value of array X5 at position 0, so X2 will have the value of 0x00.

4. Convert he following assembly instruction to 32 bit machine code and then change it to hexadecimal format.

a.

`LDUR X10, [X5, `*`#16]`*

1986 16 0 5 1010

11111000010 000010000 00 00101 010102

1111 1000 0100 0001 0000 0000 1010 1010

0xF84100AA16

b.

`SUB X12, X14, X15`

1624 15 0 14 1210

11001011000 01111 000000 01110 011002

1100 1011 0000 1111 0000 0001 1100 1100

0xCD0F01CC16

   c.

```
LSR X11, X19, #2
```

1690 0 2 19 1110

11010011010 00000 000010 10011 010112

1101 0011 0100 0000 0000 1010 0110 1011

0xD3400A6B16

5. Convert C++ code snipped to LEGv8 assembly code. The following variables x, y, and z are associated with registers X19, X20, and X21 respectively, and base address of the array A is in X22. Comment the code.

```
x = x + y;
z = x + 4;
A[8] = A[3] + z;
                ; x = X19
                ; y = X20
                ; z = X21
                ; A[0] = X22
ADD X19, X19, X20    ; x += y
ADD X21, X19, #4     ; z = x + 4
LDUR X9, [X22, #24] ; A[3]
ADD X9, X9, X21      ; A[3] + z
STUR X9, [X22, #64] ; A[8] = A[3] + z
```

6. Convert C++ code snipped to LEGv8 assembly code. The following variables x, y, and z are associated with registers X19, X20, and X21, respectively, and base address of array d is in X22. Comment the code.

   a.

```
if (x > y){
    z = y + 4;
}
else{
    z = y - 16;
}

        ; x = X19
        ; y = x20
        ; z = x21
        ; d[0] = x22
CMP X19, X20
B.LT else
ADD X21, X20, #4
```

```
B exit
else: SUB X21, X20, #16
exit:
```

b.

```
for (i = 0; i < x; i++){
    y = d[i] + z;
}
```

```
                ; x = X19
                ; y = X20
                ; z = x21
                ; d[0] = x22
ADDI X9, XZR, #0     ; i = 0
loop: CMP X9, X19    ; Comparing i and x
B.GT exit           ; i < x
LSL X10, X9, #3      ; d[i] = x22 + i * 8
ADD X20, X10, X21    ; X20 = X10 + x
STUR X20, [X22, #0]
ADDI X9, X9, #1      ; i++
B loop
exit:
```