Homework 2
EET 340
Introduction to Computer Organization and Architecture

INSTRUCTIONS: Show the detailed steps of your calculation. The homework solution can either be typed in word or handwritten. However, convert the word or scanned (handwritten) documents to PDF and submit to blackboard. Please comment your assembly code.

1. Convert Decimal value to binary and then convert them to hexadecimal value: (10 Points)

a. 45_{10}

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

32 + 0 + 8 + 4 + 0 + 1
0010 = 2, 1101 = D

$\boxed{101101_2 = 2D_{16}}$

b. 22_{10}

16 + 0 + 4 + 2 + 0
0001 = 1, 0110 = 6

$\boxed{10110_2 = 16_{16}}$

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) (10 points)

X1 = 4 * 2^2 = 16
X2 = 0X0000 0000 0000 0004

4 = 0000 0100
1^{ST} SHIFT = 0000 1000
2^{ND} SHIFT = 0001 0000

X1 = 0001 0000

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. (10 points)

Instruction is to LoaD Unscaled Register X2 (64 bit) with the contents of the memory pointed at by X5 + 0 (i * 8 bit memory contents). The 8 memory slots from 4000 to 4007 contain 0, 0, 0, 0, 0, 2, 35 in decimal which is being loaded to X5.

$\boxed{X2 = 0000\ 0000\ 0000\ 0223_{16}}$

4. Convert following assembly instruction to 32 bit machine code and then change it to Hexadecimal format. (25 Points)

```
a. LDUR X10, [X5, #16]
b. SUB X12, X14, X15
c. LSR X11, X19, #2
```

a. D-Format

OPCODE (11 bit)	ADDRESS (9 bit)	OP2 (2 bit)	Rn (5 bit)	Rt (5 bit)
1986_{10}	16_{10}	0_{10}	5_{10}	10_{10}
$1111\ 1000\ 010_2$	$0\ 0001\ 0000_2$	00_2	$00\ 101_2$	$0\ 1010_2$
				F 8 4 1 0 0 A A_{16}

b. R-Format

OPCODE (11 bit)	Rm (5 bit)	SHAMT (6 bit)	Rn (5 bit)	Rd (5 bit)
1624_{10}	15_{10}	0_{10}	14_{10}	12_{10}
$0110\ 0101\ 100_2$	$0\ 1111_2$	$0000\ 00_2$	$01\ 110_2$	0 1100
				6 5 8 F 0 1 C C_{16}

c. I-Format

OPCODE (11 bit)	Rm (5 bit)	SHAMT (6 bit)	Rn (5 bit)	Rd (5 bit)
1690_{10}	0_{10}	2_{10}	19_{10}	11_{10}
$1101\ 0011\ 010_2$	$0\ 0000_2$	$0000\ 10_2$	$10\ 011_2$	$0\ 1011_2$
				$D3400A6B_{16}$

5. Convert C++ code snippet 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. (15 Points)

```
x = x + y;
z = x + 4;
A[8] = A[3] + z;
```

```
ADD X19, X19, X20     // x = x + y

ADDI X21, X19, #4     // z = x + 4

LDUR X9, [X22, #24]  // X9 = A[3]
ADD X9, X9, X21       // X9 = A[3] +
z
STUR X9, [X22, #64]  // A[8] = X9
```

6. Convert C++ code snippet 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 d is in X22. Comment the code. (30 Points)

```
a. if (x > y)  z = y + 4;
     else z = y - 16;
```

```
CMP X19, X20          //compare x and y
B.LT L1               // if x is less than y, branch to L1
ADDI X21,X20, #4      // z = y + 4
```

```
L1: SUBI X21, X20, #16        // z = y - 16
```

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

```
LOOP:
        MOVI X9, #0           // i = 0
        CMP X9, X19           // compare i and x
        B.GE EXIT             // branch to exit if i is greater than x
        LSL X10, X9, #3       // X10 = X9 * $2^3$ = i * 8
        ADD X10, X22, X10  // X10 = X22 + (i*8) = d base + memory index
        STUR X10, [X22, #0] // indexing is stored in d[i] array
        ADD X20, X10, X21  // y = d[i] + z
        B LOOP
EXIT:
```