

CS/CE 4341 - Assignment#3

Due Date: 4/20/25, 11:59 pm

1- Add an Overflow output to the 32-bit ALU shown in Figure 5.15. The output is TRUE when the result of the adder overflows. Otherwise, it is FALSE. Write a Boolean equation for the Overflow output. Sketch the Overflow circuit. Please also add a Zero output to the 32-bit ALU. The output is TRUE when $Y = 0$. Design the modified ALU in an HDL.

2- Design an 8-bit left shifter using only 24 2:1 multiplexers. The shifter accepts an 8-bit input, A, and a 3-bit shift amount, shamt2:0. It produces an 8-bit output, Y. Sketch the schematic

3- A zero extension unit extends an unsigned number from M to N bits ($N > M$) by putting zeros in the upper bits of the output. Sketch a circuit for a zero extension unit with a 4-bit input and an 8-bit output. Write the HDL for your design.

4- Design a multiplier that handles two's complement numbers.

5- Consider the following MIPS assembly language snippet. The numbers to the left of each instruction indicate the instruction address.

a) Translate the instruction sequence into machine code. Write the machine code instructions in hexadecimal.

b) List the addressing mode used at each line of code.

```

0x00400028      add  $a0, $a1, $0
0x0040002c      jal  f2
0x00400030  f1:   jr   $ra
0x00400034  f2:   sw   $s0, 0($s2)
0x00400038      bne  $a0, $0, else
0x0040003c      j    f1
0x00400040  else: addi $a0, $a0, -1
0x00400044      j    f2
  
```

6- Translate the following program to MIPS assembly program (Please explain each instruction in your code by a comment and submit a .asm file)

```

#include <stdio.h>

int main()
{
    //Initialize array
    int arr[] = {5, 2, 8, 7, 1};
    int temp = 0;

    //Calculate length of array arr
    int length = sizeof(arr)/sizeof(arr[0]);

    //Displaying elements of original array
    printf("Elements of original array: \n");
    for (int i = 0; i < length; i++) {
        printf("%d ", arr[i]);
    }

    //Sort the array in ascending order
    for (int i = 0; i < length; i++) {
        for (int j = i+1; j < length; j++) {
            if(arr[i] > arr[j]) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }

    printf("\n");

    //Displaying elements of array after sorting
    printf("Elements of array sorted in ascending order: \n");
    for (int i = 0; i < length; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
  
```