CS 382
Fall 2024

Computer Architecture
and Organization

Lab 1

1 Task: Display a Binary Integer

In this lab, we're going to manipulate binary numbers in a C program. More specifically, we're going to write a C program that can print out the binary pattern of any 32-bit integer numbers.

We have provided a start code for you:

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>

void display(int8_t bit) {
    putchar(bit + 48);
}

void display_32(int32_t num) {
    /* Your code here */
}

int main(int argc, char const *argv[]) {
    display_32(382);
    return 0;
}
```

where `display_32()` is the function you need to complete. Here we use `int32_t` and `int8_t` as substitutes for `int` and `char` type, to make sure they have the same number of bits across different machines. The general algorithm would be extracting every bit of the number using bit-wise operations and shifting, while calling `display()` to print out **one bit**.

An example of output from the code above would be:

```
00000000000000000000000101111110
```

Notice two things:

- You need to output all 32 bits with leading zeros;
- MSB is the leftmost bit while LSB the rightmost, so you need to print out MSB first, and LSB last.

Requirements

- Your code must be able to compile successfully and executed without segmentation fault or any other type errors;
- Write your name and honor code pledge at the top of your code as comments;
- You must not change anything provided in the starter code, except in `main()` where you can write your own tests;

- You must not use division or multiplication in any part of your code (addition and subtraction are allowed, though); only use shifting (`<<` and/or `>>`) and bit-wise operators (`&` and/or `|`) to extract individual bits;
- You can create any functions that can help you, but you must call `display()` and `display_32()` functions. Also, you wouldn't need to include any more header files;
- All 32 bits must be printed out;
- MSB is the leftmost bit, while LSB the rightmost.

2 Grading

The lab will be graded based on a total of 10 points.

- **-10:** the code does not compile, or executes with run-time error;
- **-10:** if used multiplication and/or division and/or modulo operators;
- **-5:** `display()` and/or `display_32()` are not used;
- **-5:** included other header files, and/or the starter code was changed (except `main()`);
- **-5:** no display of binary number and/or the result is incorrect;
- **-3:** negative numbers are not displayed correctly;
- **-3:** leading zeros are not printed out;
- **-3:** the binary number is printed in the reverse order (*i.e.,* MSB is the *right*-most);
- **-1:** no pledge and/or name in C file.

Earlybird Extra Credit: 2% of extra credit will be given if the lab is finished by Thursday 11:59PM EST (1 day before the lab deadline). For specific policy, see syllabus.

Attendance: check off at the end of the lab to get attendance credit.

Deliverable

Submit a single `.c` file on Canvas.