CS-382 Computer Architecture and Organization Fall 2022

Lab 2 · Binary Numbers in C

Lecturer: Shudong Hao *Date: See Canvas*

In this lab, we're going to manipulate binary numbers in 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>

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: 1) you need to output all 32 bits with leading zeros, and 2) MSB is the leftmost bit while LSB the rightmost, so you need to print out MSB first, and LSB last.

Requirements

Code that doesn't compile will receive zero credits – no exception!

- Write your name and honor code pledge at the top of your code as comments;
- Do not change anything provided in the starter code, except in `main()` where you can write your own tests;
- Do 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'd have to use the `display()` and `display_32()` functions. Also, you wouldn't need to include any more header files;
- All 32 bits (with leading zeros) should be printed out;
- MSB is the leftmost bit, while LSB the rightmost.

Deliverable

Submit a single `.c` file on Canvas.