Jetic Gū
Columbia College

1. Handwritten submissions and proprietary formats (e.g. Pages or MS Word) **will not be graded**.
2. Late submission and resubmission policies are stated on the course webpage.
3. Mathematical expressions must be written entirely using LaTeX, otherwise **50%-100%** of marks will be deducted.
4. Circuits must be **tested** using switches/probs against a truth table. Untested circuits will receive 0.

Submission File structure:

```
submission.zip
    — answer.pdf
    — c1-1.cct
    — c1-2.cct
    — c2.cct
    — c3.cct
    — c4.cct
```

The files `circuit1-1`, `circuit1-2`, `circuit3` are 1pt each, `circuit2` 2pt, `circuit3` 3pt, `answer.pdf` 2pt.

Lab 4

5. (PDF) Datapath conceptual question: assume a datapath with a 4-bit register array (4 GPRs inside) that can perform certain functions. The datapath takes input $\texttt{OP}_{2:0}$ from the control unit for function selection, $\texttt{rd}_{1:0}$ and $\texttt{rs}_{1:0}$ for register selection, $\texttt{in}_{3:0}$ for value input.

$\texttt{OP}_{2:0}$	**Register Operation**
000	No change
001	Clear a single register to 0, selected by $\texttt{rd}_{1:0}$
010	Perform register transferring, assign the value of register at address $\texttt{rs}_{1:0}$ to register at $\texttt{rd}_{1:0}$.
011	Load value from $\texttt{in}_{3:0}$ into a single register, selected by $\texttt{rd}_{1:0}$
110	Perform addition of 2 register values, selected by $\texttt{rd}_{1:0}$ and $\texttt{rs}_{1:0}$, store the output to register with address $\texttt{rd}_{1:0}$. (Use the adder-subtractor functional block)
111	Perform subtraction of 2 register values, selected by $\texttt{rd}_{1:0}$ and $\texttt{rs}_{1:0}$, store the output to register with address $\texttt{rd}_{1:0}$. (Use the adder-subtractor functional block)

Write down the sequence for all necessary inputs for computing 4 + 5 - 7. You will need to load number 4, 5, 7 into the datapath, then perform the necessary calculation, and finally store the result in register number 0. (2pt)

Hint: here's a sample for loading value 3 into register number 0, and 2 into register 1 (one line per):

$\texttt{OP}_{2:0}$ = 011, $\texttt{rd}_{1:0}$ = 00, $\texttt{in}_{3:0}$ = 0011

$\texttt{OP}_{2:0}$ = 011, $\texttt{rd}_{1:0}$ = 01, $\texttt{in}_{3:0}$ = 0010

6. Register design:
   A. Draw the circuit diagram of a D Flip-Flop with EN, using the `D flip-flop wo/SQ` component in the system library. Save it as a component in your library, as well as in a circuit file (`c1-1.cct`).
   Requirement: your CCT file must show your component being tested using switches and probs.
   B. Draw the circuit diagram of a 4bit Register using the above D Flip-Flop with EN, your register must have $D_3D_2D_1D_0$, EN, C, and R as input ports, and $Q_3Q_2Q_1Q_0$ as output (`c1-2.cct`).
   Requirement: your CCT file must show your component being tested using switches, probs, and HEX Keyboard and Display.
7. Register array: draw the circuit diagram of a Register array with 4 registers, that meets the following specification (`c2.cct`):
   A. The register array will have one 4bit `rd_in` bus providing new values to be stored, 2bit `rd` bus specifying the register to take in new values;
   B. one 4bit `rs_out` bus outputting values from the register array, selected by the 2bit `rs` bus;
   C. a single `Clear` switch that can clear all registers to 0; and
   D. a single `CLK` switch simulating the clock unit.
   E. you should use your own register in Q1, 2-to-4 decoder, 4channel 4bit multiplexer.
   Requirement: your CCT file must show your component being tested using switches, probs, and HEX Keyboard and Display.
8. Datapath functional block: implement a 4bit Bitwise NOT component (`c3.cct`).
9. Final assembly:
   A. Copy your design from `c2.cct`, name it `c4.cct`.
   B. Overall Inputs:
      I. `func_in`, a hex keyboard
      II. `mode`, a switch for functional block mode
      III. `OP`, a hex keyboard, using least significant 2bits for function selection
      IV. `rd`, a hex keyboard, using least significant 2bits

V. `rs`, a hex keyboard, using least significant 2bits

VI. `rt`, a hex keyboard, using least significant 2bits

VII. `CLK`, a switch for simulating clock

VIII. `Clear`, a switch for clearing all registers

C. You should have 4 functional blocks, selected by 2bit input bus `op`:

I. Function 0: register assignment, takes input from a HEX keyboard (`func_in`);

II. Function 1: register transferring, takes input from the register output bus (`rs_out`);

III. Function 2: Bitwise NOT, takes input from the register output bus (`rs_out`), outputs its bitwise complement.

IV. Function 3: Adder-Subtract, takes input from the register output bus (`rs_out`), and another register (`rt_out`), specified by 2bit `rt` bus. There should also be a mode switch input, selecting between performing addition and subtraction.

D. The output from the functional block selected by `op` will be fed back into the register array on `rd_in`, replacing the keyboard in `c2.cct`.