Jetic Gū

## Columbia College

Please remember to write your name and student number.
You must complete the following assignment and submit a PDF of relevant questions. Handwritten submissions and proprietary formats (e.g. Pages or MS Word) will not be accepted. You will also need to upload LogicWork circuit design file. Then upload a single ZIP file to Moodle.
Submission File structure:

```
submission.zip
    - answer.pdf
    - circuit1-1.cct
    - circuit1-2.cct
    - circuit2.cct
    - circuit3.cct
    - circuit4.cct
    - lib.clf
```

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

## Lab 4

1. (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 $\mathrm{Func}_{2: 0}$ from the control unit for function selection, reg $_{1: 0}$ and reg_21:0 for register selection, Func_in $3: 0$ for value input.

| Func2:0 | Register Operation |
| :---: | :---: |
| 000 | No change |
| 001 | Clear a single register to 0 , selected by reg ${ }_{1}: 0$ |
| 010 | Perform register transferring, assign the value of register at address reg_2 1:0 $^{\text {to }}$ register at regi:0. |
| 011 | Load value from Func_in ${ }_{3}$ :0 into a single register, selected by $\mathrm{reg}_{1}$ :0 |
| 110 | Perform addition of 2 register values, selected by reg $\mathrm{g}_{1: 0}$ and reg_2 $2_{1: 0}$, store the output to register with address $\mathrm{reg}_{1: 0}$. (Use the adder-subtractor functional block) |
| 111 | Perform subtraction of 2 register values, selected by reg 1 : and reg_ $21: 0$, store the output to register with address $\mathrm{reg}_{1: 0}$. (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 4 into register number 0 , and 5 into register 1 (one line per):

$$
\begin{aligned}
& \text { Func }_{2: 0}=011, \text { reg }_{1: 0}=00, \text { Func_in }_{3: 0}=0100 \\
& \text { Func }_{2: 0}=011, \text { reg }_{1: 0}=01, \text { Func_in } n_{3: 0}=0101
\end{aligned}
$$

2. Register design:
A. Draw the circuit diagram of a D Flip-Flop with EN, using the $D f l i p-f l o p$ wo/SQ component in the system library. Save it as a component in your library, as well as in a circuit file (circuit1-1.cct).
B. Draw the circuit diagram of a 4bit Register using the above D Flip-Flop with EN, your register must have $D_{3} D_{2} D_{1} D_{0}, E N, C$, and $R$ as input ports, and $Q_{3} Q_{2} Q_{1} Q_{0}$ as output (circuit1-2.cct).
3. Register array: draw the circuit diagram of a Register array with 4 registers, that meets the following specification (circuit2.cct):
A. the register array will have one 4bit data_in bus providing new values to be stored, 2bit reg_in bus specifying the register to take in new values;
B. one 4bit data_out bus outputting values from the register array, selected by the 2bit reg_out 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. I also recommend using the HEX display and keyboards to help run simulations.
4. Datapath functional block: implement a 4bit Bitwise NOT component (circuit3.cct).
5. Final assembly:
A. Copy your design from circuit2.cct, name it circuit4.cct.
B. Overall Inputs:
I. func_in, a hex keyboard
II. mode, a switch for functional block mode
III. func, a hex keyboard, using least significant 2 bits for function selection
IV. reg_in, a hex keyboard, using least significant 2bits
V. reg_out, a hex keyboard, using least significant 2bits
VI. reg_out_2, 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 func:
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 (data_out);
III. Function 2: Bitwise NOT, takes input from the register output bus (data_out), outputs its bitwise complement.
IV. Function 3: Adder-Subtract, takes input from the register output bus (data_out), and another register (data_out_2), specified by 2bit reg_out_2 bus. There should also be a mode switch input, selecting between performing addition and subtraction.
D. The output from the functional block selected by func will be fed back into the register array on data_in, replacing the keyboard in circuit2.cct.
