NYCU-EE IC LAB – Spring 2023

Lab01 Practice

Simulation flow

Objective

We will guide you to simulate RTL design, synthesize design and simulate gate-level design.

Reference data

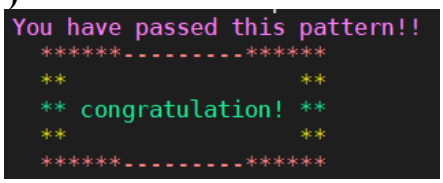
1. Use the lecture document that was on the IC Lab course page
2. Ask the TA if you have any questions.

Data Preparation

1. Extract test data from TA's directory:
   % tar xvf ~iclabta01/Lab01.tar

Simulation Flow:

1. Enter Lab01/Practice/01_RTL folder by command
   "cd Lab01/Practice/01_RTL"
2. Simulate RTL design by command "./01_run"
3. **Check the simulation result, which represents whether your design functions correctly or not. (You will see congratulation message like this on the terminal if you pass the pattern.)**



4. Use command "nWave &" to open nWave and check the waveform to debug.
   After finishing RTL design, enter "Lab01/Practice/02_SYN folder" by command "cd ../02_SYN"
5. Synthesize design by command "./01_run_dc"
6. After 02_SYN simulation done the files of"CORE_SYN.v","CORE_SYN.sdf" would be created under the folder " Lab01/Practice/02_SYN/Netlist"
7. Enter the"Lab01/Practice/02_SYN/Report" folder. **Check timing and area report to ensure your design meets the spec.**

```
Operating Conditions: slow   Library: slow
Wire Load Model Mode: top

  Startpoint: in_n0[1] (input port)
  Endpoint: out_n[4] (output port)
  Path Group: default
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  input external delay                     0.00       0.00 r
  in_n0[1] (in)                            0.00       0.00 r
  U59/Y (NAND2XL)                          0.13       0.13 f
  U70/Y (INVXL)                            0.14       0.27 r
  U76/Y (NAND2XL)                          0.11       0.38 f
  U78/Y (NAND2XL)                          0.11       0.49 r
  U79/Y (OAI21XL)                          0.12       0.61 f
  U60/Y (NOR2X1)                           0.14       0.74 r
  U62/S (ADDFXL)                           0.99       1.73 f
  U86/Y (NAND2XL)                          0.22       1.95 r
  U87/Y (MXI2XL)                           0.22       2.17 f
  U88/Y (AOI2BB2XL)                        0.30       2.48 f
  U89/Y (AOI21XL)                          0.17       2.65 r
  U90/Y (INVXL)                            0.04       2.69 f
  out_n[4] (out)                           0.00       2.69 f
  data arrival time                                   2.69

  max_delay                               30.00      30.00
  output external delay                    0.00      30.00
  data required time                                 30.00
  -----------------------------------------------------------
  data required time                                 30.00
  data arrival time                                  -2.69
  -----------------------------------------------------------
  slack (MET)                                        27.31
```

(Be careful!! In the CORE.timing file, slack should be "MET")

8. **Enter 02_SYN and check the syn.log file to see if there are latches in the design. (Be careful!! You can't have latch in your design.)**

```
============================================================================
|    Register Name    | Type  | Width | Bus | MB | AR | AS | SR | SS | ST |
============================================================================
|      out_n_reg      | Latch |   7   |  Y  | N  | N  | N  | -  | -  | -  |
============================================================================
```

Forbidden !!

9. Enter Lab01/Practice/03_GATE_SIM folder
10. Simulate gate-level design by command "./01_run"
11. **Check the gate level simulation result, which represents whether your synthesized design functions are correct or not.**
12. Use command "nWave &" to open nWave and check the waveform
13. Clean the log files and dump files by command "./09_clean_up"(Recommended!)

Design

A simple calculator with addition, subtraction. A module FA is provided, you need to use it to build the two mentioned operations.

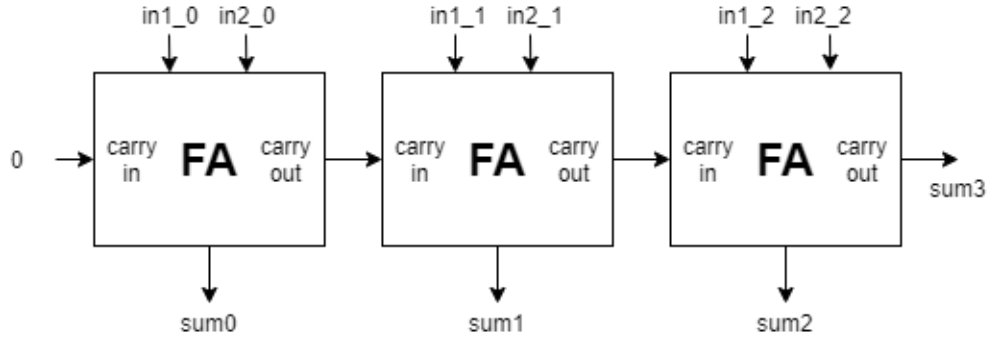
INPUT 3 bits: in_n0, in_n1.
INPUT 1 bits : opt
OUTPUT 4 bits : out_n

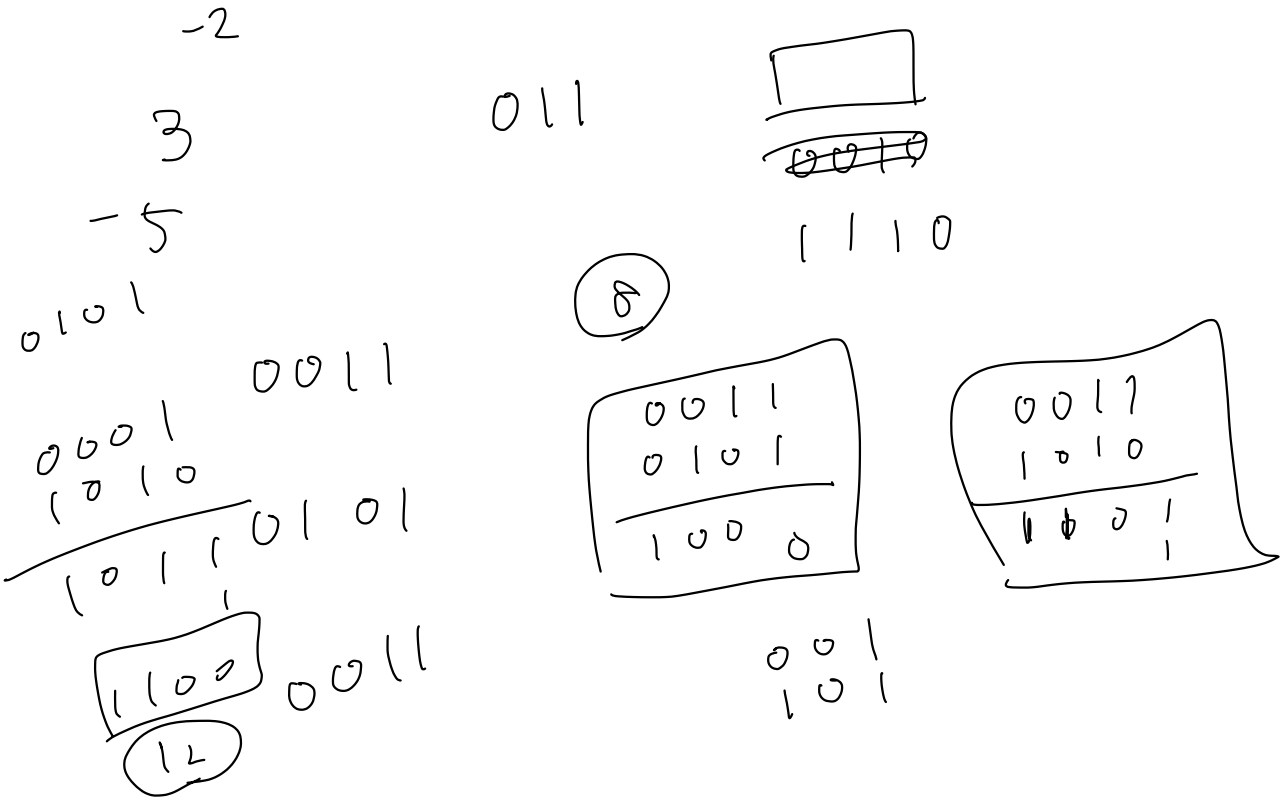
opt=0 → addition: in_n0 + in_n1
opt=1 → subtraction: in_n0 – in_n1

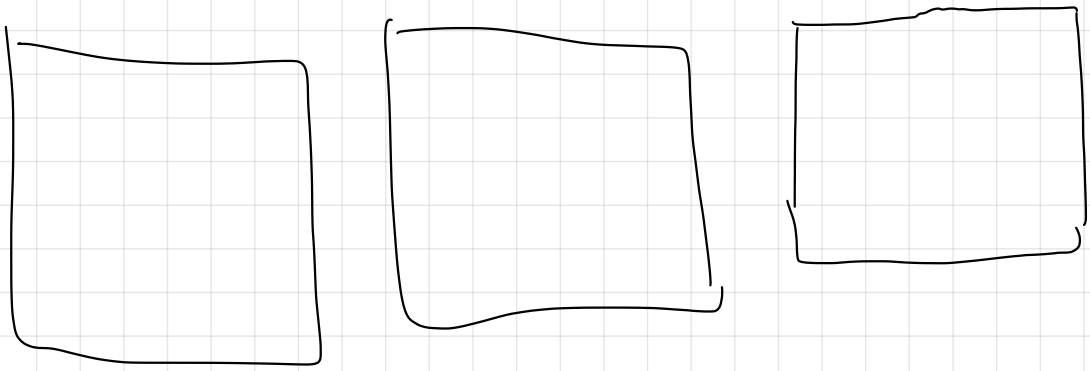
You should use the module FA provided in CORE.v to design your calculator. There's no area constraint for this practice, so feel free to use any number of FAs. However, using the fewest number of FAs to finish the task is recommended.

Hints

1. **You are not allowed to use arithmetic operation +, -, * in this practice. Instead, you should use gate-level, for example, the module provided by TA.**
2. **Example for adding two 3-bits variables:**







$$\begin{array}{r} 001 \\ 010 \\ \hline 011 \end{array}$$

$$\begin{array}{r} 001 \\ 010 \\ \hline 011 \\ 101 \\ \hline \end{array}$$