NYCU-EE IC LAB – Spring 2024

Online Test 2024/04/13

Design: Infix to prefix convertor and prefix evaluation

Data Preparation

Extract test data from TA's directory:

% tar xvf ~iclabTA01/OT_2024_Spring.tar

Design Description

Infix, prefix, postfix is important for computer operation. In this test, you will be given 10 operand signal, 9 operator signal and 1 option signal indicating whether to do prefix evaluation or convert infix expression into prefix expression. For more details, when the option signal = 0, the 10 operand signal and the 9 operator signal are given in prefix expression. You need to evaluate the result. When the option signal = 1, the 10 operand signal and 9 operator signal are given in infix expression. You need to convert them into prefix expression. The description of calculations is summarized in the following table：

Option signal	Detailed description
opt = 0	**Example 1** Input：/ * 2 4 * 3 8 (prefix expression) **Step 1**：3 * 8 = 24 **Step 2**：2 * 4 = 8 **Step 3**：8 / 24 = 0 (round down for division) **Example 2** Input : * + 2 4 - + 1 3 5 (prefix expression) **Step 1** : 1 + 3 = 4 **Step 2** : 4 - 5 = -1 **Step 3** : 2 + 4 = 6 **Step 4** : 6 * -1 = -6 **Example 3** Input : + + 5 / * 2 8 3 6 (prefix expression) **Step 1** : 2 * 8 = 16 **Step 2** : 16 / 3 = 5 (round down for division) **Step 3** : 5 + 5 = 10

	Step 4 : 10 + 6 = 16 **Example 4** Input : - - + * / 1 2 3 4 5 * 6 2 (prefix expression) **Step 1** : 6 * 2 = 12 **Step 2** : 1 / 2 = 0 (round down for division) **Step 3** : 0 * 3 = 0 **Step 4** : 0 + 4 = 4 **Step 5** : 4 – 5 = -1 **Step 6** : -1 – 12 = -13 **Example 5** Input : + + - * 3 8 4 / 5 1 * 3 2 (prefix expression) **Step 1** : 3 * 2 = 6 **Step 2** : 5 / 1 = 5 (round down for division) **Step 3** : 3 * 8 = 24 **Step 4** : 24 – 4 = 20 **Step 5** : 20 + 5 = 25 **Step 6** : 25 + 6 = 31
opt = 1	Input : 3 + 5 / 4 – 6 (infix expression) **Step 1** : reverse the string : 6 – 4 / 5 + 3 **Step 2** : In this step, you will get the **R**eversed **P**refix **E**xpression (**RPE**). You need to build a stack to store the operator and scan the string (6 – 4 / 5 + 3) from left to right element by element. 1. If this element is operand, then put it in the **RPE**. 2. If this element is operator and your stack of operator is empty, then put this operator in the stack. 3. If this element is operator and your stack of operator is not empty. (1) You need compare the priority with the operator on the top of stack. If current operator’s priority is smaller than the top operator’s priority, go step (2). Else, go step (3). **(the priority of ‘+’ and ‘-’ are smaller than ‘*’ and ‘/’)** (2) Pop the operator on the top of stack, put it in the **RPE**, and goes to step (1) to compare the priority again. (3) Put the current operator in the stack. 4. If all elements in the string have been scanned, then sequentially pop out the remaining operators in the stack and put them into

	the **RPE**. **Step 3** : In step2, you get the **RPE** : 6 4 5 / 3 + -, reverse the **RPE**, you can get - + 3 / 5 4 6, then it is the prefix expression we wanted. out = {5’b10001, 5’b10000, 5’b00011, 5’b10011, 5’b00101, 5’b00100, 5’b00110} = {-, +, 3, /, 5, 4, 6}

Inputs

Input	Bit Width	Definition and Description
clk	1	Clock
rst_n	1	**Asynchronous** active low reset
in_valid	1	Raised for 19 cycles to indicate opt signal and data signal is valid.
opt	1	opt signal is only valid for the first cycle when in_valid signal is at high level. When **opt = 0**, in_data is given in prefix expression. You need to do prefix evaluation. When **opt = 1** ： in_data is given in infix expression. You need to convert the in_data into prefix expression.
in_data	5	in_data[4] is to indicate the type for data signal. in_data[4] = 0, in_data[3:0] means a value in 1~15, the pattern will not generate the 0 value. in_data[4] = 1, in_data[3:0] means an operation symbol. ◆ in_data[3:0] = 4’b0000 means “ + ” ◆ in_data[3:0] = 4’b0001 means “ - ” ◆ in_data[3:0] = 4’b0010 means “ * ” ◆ in_data[3:0] = 4’b0011 means “ / ” Note： (1) The pattern will not generate the case for denominator = 0. In other word, no need to consider divided by 0. (2) the priority of ‘+’ and ‘-’ are smaller than ‘*’ and ‘/’. (3) round down for division

Outputs

Output	Bit Width	Definition and Description
out_valid	1	Raised for only 1 cycle to indicate output. It should be LOW after reset.
out	95	When opt = 0, you need to output the prefix evaluation. (signed value) When opt = 1, you need to output the prefix expression. The first element in the prefix expression is put on LSB side, and the last element is put on MSB side.

Specifications

1. Top module name: **PREFIX** (File name: **PREFIX.v**)
2. Input pins: **clk, rst_n, in_valid, opt** and **in_data [4:0].**
3. Output pins: **out_valid** and **out[94:0].**
4. It is **asynchronous** reset and **active-low** architecture.
5. All output signals should be **LOW** after reset
6. **out_valid** CANNOT be raised when **in_valid** is high.
7. **out** signal should be 0 when **out_valid** is LOW.
8. The **out_valid** will be high only for 1 cycle.
9. The latency of your design in each pattern should not be larger than **100** cycles.
10. The clock period is fixed to **45ns**.
11. The input delay for all inputs is set to **0.5*45 ns**.
12. The output delay for all outputs is set to **0.5*45ns**.
13. The output loading is set to 0.05.
14. The synthesis result of data type cannot include any **Latch** and Error in syn.log
15. **You will fail demo if your synthesis time is over 1 hours.**
16. After synthesis, you can check PREFIX.area, PREFIX.timing and PREFIX.resource.
17. The slack in the end of PREFIX.timing should be **non-negative** and the result should be **MET**.
18. You can't have timing violation in gate-level simulation.
19. Don't use any wire/reg/submodule name called "*latch*", "*error*" or "*congratulation*", or you will fail the lab.

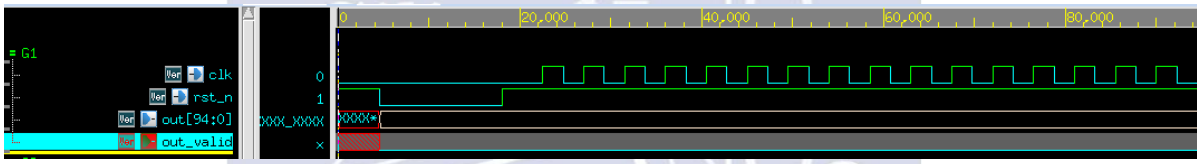
Grading Policy

1. We will use the PATTEN.v we provided to validate your DESIGN. **You cannot output the answers directly; otherwise, you will get a negative 100 points.**
2. **Part1 (40%)** : **opt** is fixed to **0,** RTL and gate-level simulation correctness. (./01_run_vcs_rtl PART1 and ./01_run_vcs_gate PART1)
3. **Part2 (60%)** : **opt** would be **0 or 1**, RTL and gate-level simulation correctness. (./01_run_vcs_rtl and ./01_run_vcs_gate)
4. **The grade of take home (3rd demo) would be 50% off.**
   (Take home (3rd demo) deadline: 04/15(Mon) 12:00)

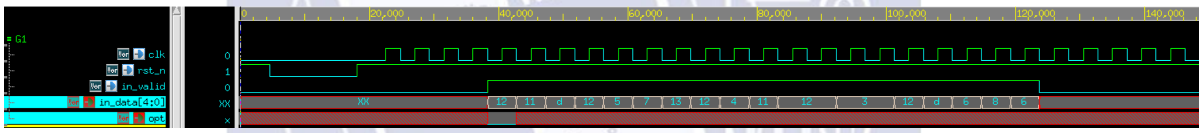
(Example : Student A completed Part 1 on 1st demo and finished Part 2 on 2nd demo, then he can get 40 + 0.5*60 = 70 points.)

Sample waveform

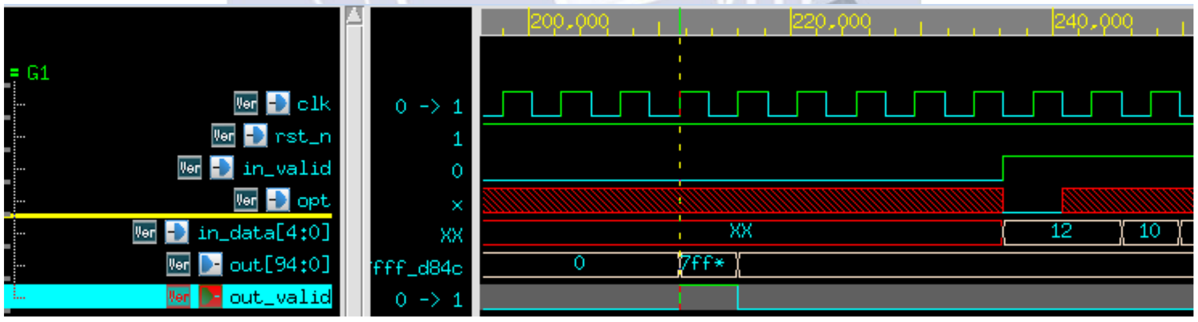
1. Reset



2. Input



3. Output

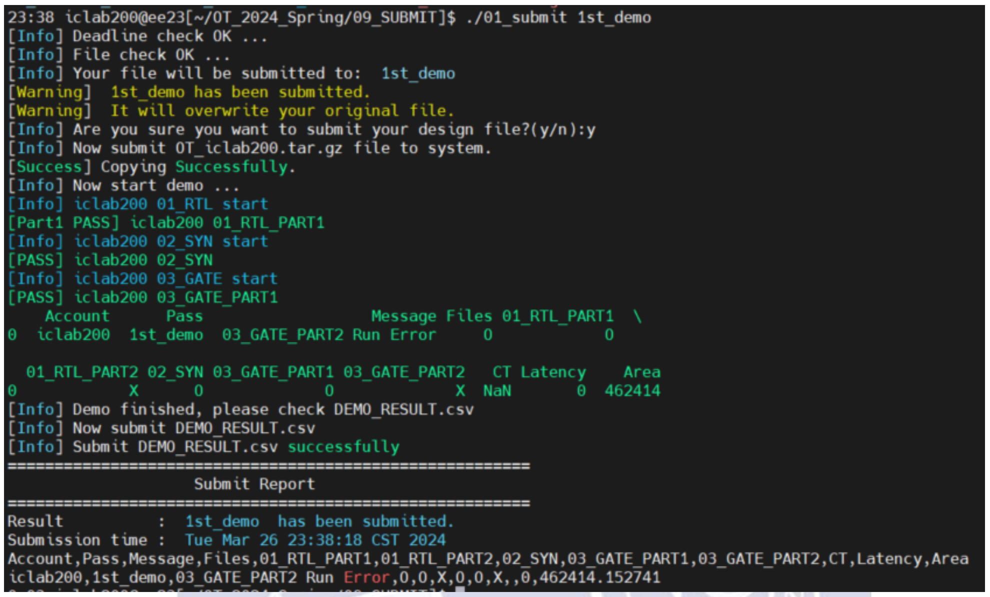


Submission step

1. **cd 09_SUBMIT**
2. **./00_tar 45**
3. **./01_submit**

   (1st_demo for Group A, 2nd_demo for Group B, 3nd_demo for take home)

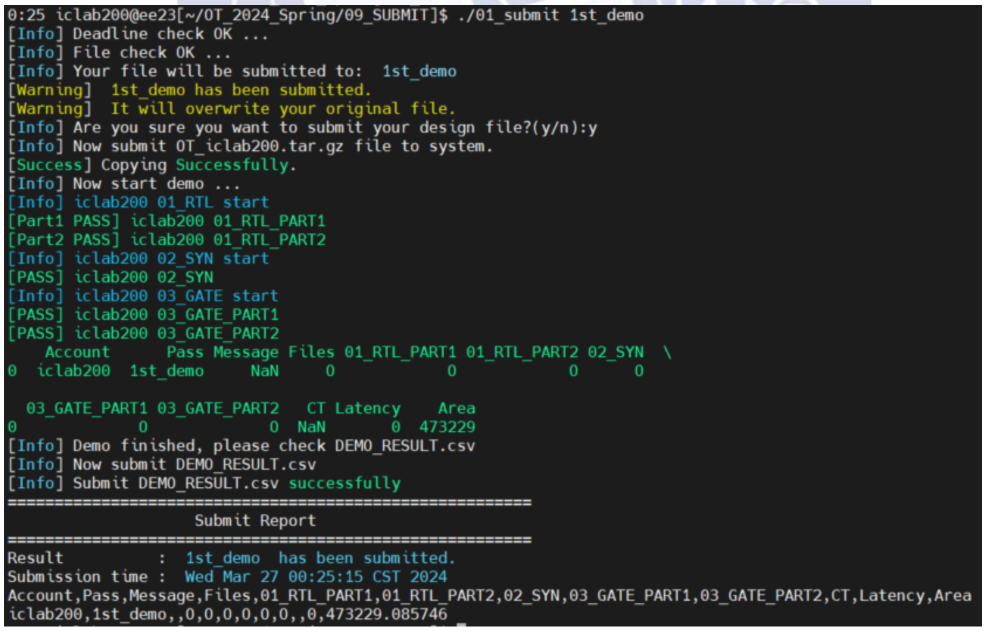
   Only Part1 Pass:



```
23:38 iclab200@ee23[~/OT_2024_Spring/09_SUBMIT]$ ./01_submit 1st_demo
[Info] Deadline check OK ...
[Info] File check OK ...
[Info] Your file will be submitted to:  1st_demo
[Warning]  1st_demo has been submitted.
[Warning]  It will overwrite your original file.
[Info] Are you sure you want to submit your design file?(y/n):y
[Info] Now submit OT_iclab200.tar.gz file to system.
[Success] Copying Successfully.
[Info] Now start demo ...
[Info] iclab200 01_RTL start
[Part1 PASS] iclab200 01_RTL_PART1
[Info] iclab200 02_SYN start
[PASS] iclab200 02_SYN
[Info] iclab200 03_GATE start
[PASS] iclab200 03_GATE_PART1
    Account      Pass                  Message Files 01_RTL_PART1  \
0  iclab200  1st_demo  03_GATE_PART2 Run Error     0            0

  01_RTL_PART2 02_SYN 03_GATE_PART1 03_GATE_PART2   CT Latency    Area
0            X      0             0             X  NaN       0  462414
[Info] Demo finished, please check DEMO_RESULT.csv
[Info] Now submit DEMO_RESULT.csv
[Info] Submit DEMO_RESULT.csv successfully
========================================================
                    Submit Report
========================================================
Result          :  1st_demo  has been submitted.
Submission time :  Tue Mar 26 23:38:18 CST 2024
Account,Pass,Message,Files,01_RTL_PART1,01_RTL_PART2,02_SYN,03_GATE_PART1,03_GATE_PART2,CT,Latency,Area
iclab200,1st_demo,03_GATE_PART2 Run Error,0,0,X,0,0,X,,0,462414.152741
```

   Part1, Part2 Pass:

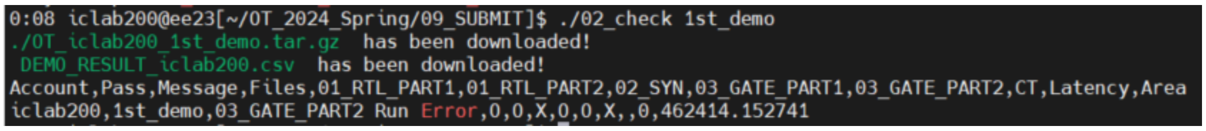


```
0:25 iclab200@ee23[~/OT_2024_Spring/09_SUBMIT]$ ./01_submit 1st_demo
[Info] Deadline check OK ...
[Info] File check OK ...
[Info] Your file will be submitted to:  1st_demo
[Warning]  1st_demo has been submitted.
[Warning]  It will overwrite your original file.
[Info] Are you sure you want to submit your design file?(y/n):y
[Info] Now submit OT_iclab200.tar.gz file to system.
[Success] Copying Successfully.
[Info] Now start demo ...
[Info] iclab200 01_RTL start
[Part1 PASS] iclab200 01_RTL_PART1
[Part2 PASS] iclab200 01_RTL_PART2
[Info] iclab200 02_SYN start
[PASS] iclab200 02_SYN
[Info] iclab200 03_GATE start
[PASS] iclab200 03_GATE_PART1
[PASS] iclab200 03_GATE_PART2
    Account      Pass Message Files 01_RTL_PART1 01_RTL_PART2 02_SYN  \
0  iclab200  1st_demo     NaN     0            0            0      0

  03_GATE_PART1 03_GATE_PART2   CT Latency    Area
0             0             0  NaN       0  473229
[Info] Demo finished, please check DEMO_RESULT.csv
[Info] Now submit DEMO_RESULT.csv
[Info] Submit DEMO_RESULT.csv successfully
========================================================
                    Submit Report
========================================================
Result          :  1st_demo  has been submitted.
Submission time :  Wed Mar 27 00:25:15 CST 2024
Account,Pass,Message,Files,01_RTL_PART1,01_RTL_PART2,02_SYN,03_GATE_PART1,03_GATE_PART2,CT,Latency,Area
iclab200,1st_demo,,0,0,0,0,0,0,,0,473229.085746
```

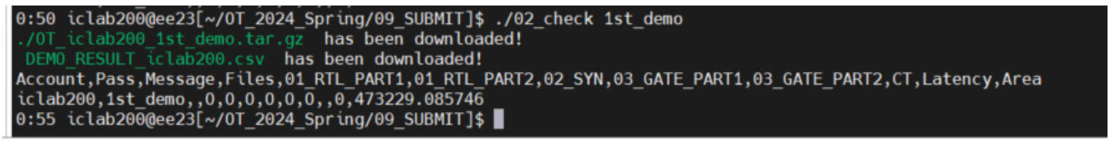
4. **./02_check**

   Only Part1 Pass:



```
0:08 iclab200@ee23[~/OT_2024_Spring/09_SUBMIT]$ ./02_check 1st_demo
./OT_iclab200_1st_demo.tar.gz  has been downloaded!
 DEMO_RESULT_iclab200.csv  has been downloaded!
Account,Pass,Message,Files,01_RTL_PART1,01_RTL_PART2,02_SYN,03_GATE_PART1,03_GATE_PART2,CT,Latency,Area
iclab200,1st_demo,03_GATE_PART2 Run Error,0,0,X,0,0,X,,0,462414.152741
```

Part1, Part2 Pass:



```
0:50 iclab200@ee23[~/OT_2024_Spring/09_SUBMIT]$ ./02_check 1st_demo
./OT_iclab200_1st_demo.tar.gz  has been downloaded!
 DEMO_RESULT_iclab200.csv  has been downloaded!
Account,Pass,Message,Files,01_RTL_PART1,01_RTL_PART2,02_SYN,03_GATE_PART1,03_GATE_PART2,CT,Latency,Area
iclab200,1st_demo,,0,0,0,0,0,0,,0,473229.085746
0:55 iclab200@ee23[~/OT_2024_Spring/09_SUBMIT]$
```