2023 FPGA Final Exercise

Topic：Polish Notation Demystification

I. Introduction

Polish Notation in the data structure is a method of expressing mathematical, logical, and algebraic equations universally. This notation is used by the compiler to evaluate mathematical equations based on their order of operations. When parsing mathematical expressions, three types of notations are commonly used : Infix Notation, Prefix Notation, and Postfix Notation. Here is some example in the table.

Infix Expression	Prefix Expression (normal Polish notation)	Postfix Expression (reverse Polish notation)
(A＋B)＊C	＊＋ABC	AB＋C＊
A＊B＋C＊D	＋＊AB＊CD	AB＊CD＊＋
A－(B＋C)－D	－－A＋BCD	ABC＋－D－

As you can see in the table, we do not need to use Parentheses in prefix and postfix expression. Take (A＋B)＊C for example. In prefix expression, the addition will do before multiplication because its operator priority is higher.

II. Lab Introduction

In this LAB, you will need to complete calculations using Polish notation. The calculation method will differ depending on the mode, and the output signals will also need to be adjusted accordingly. There will be more details provided below.

When the "**in_valid**" signal is high, TESTBED will send the 3-bit signal "**in**" and 1-bit signal "**operator**". The 1-bit "**operator**" signal will decide whether the "**in**" signal is an operand or operator.

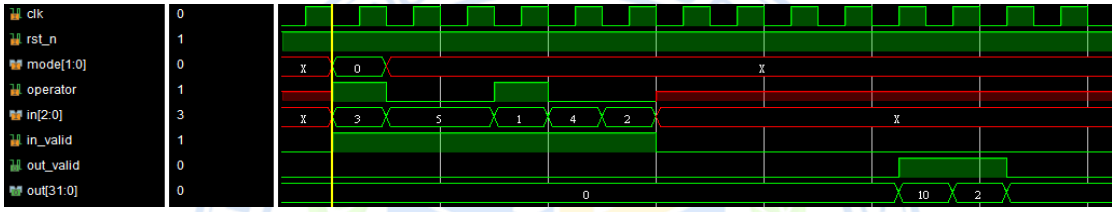
In(**unsigned**)	I	3	Operator signal is 1'b0： 3'b000~3'b111：represent the unsigned Number between 0~7 Operator signal is 1'b1： 3'b000：＋ => a ＋ b 3'b001：－ => a － b 3'b010：＊ => a ＊ b 3'b011：$ => | a ＋ b |

The two-bit "**mode**" signal will be given in the first cycle while "**in_valid**" is high, then turn back to unknown. And the following description will tell you what should do in the different modes.

case 1

If the “**mode**” signal is 2’d0, prefix notation will be chosen. Every three continuous “**in**” signals will be considered as a group of prefix expression. After completing the calculation for each group, you need to sort the answers in descending order.

For example, suppose “**in_valid**” is high for 6 cycles, the “**mode**” signal is 2’d0, the “**in**” signal is 3’d3, 3’d5, 3’d5, 3’d1, 3’d4, 3’d2, and the “**operator**” signal is 1’b1, 1’b0, 1’b0, 1’b1, 1’b0, 1’b0 for the continuous six cycles. In this case, it means that there are two prefix expressions: expression_1 is "$55", and expression_2 is "－42". After calculation, you will get ans_1 ： ”10”, and ans_2 ： ”2”. Before pulling the "**out_valid**" signal high, you should sort the answer in descending order. The “**out**” signal will be 32’d10, and 32’d2 in continuous 2 cycles, and the “**out_valid**” signal should raise 2 cycles.



case 2

If the “**mode**” signal is 2’d1, postfix notation will be chosen. The calculation method is the same as **case1**. The only different is that you need to sort the answer in ascending order.

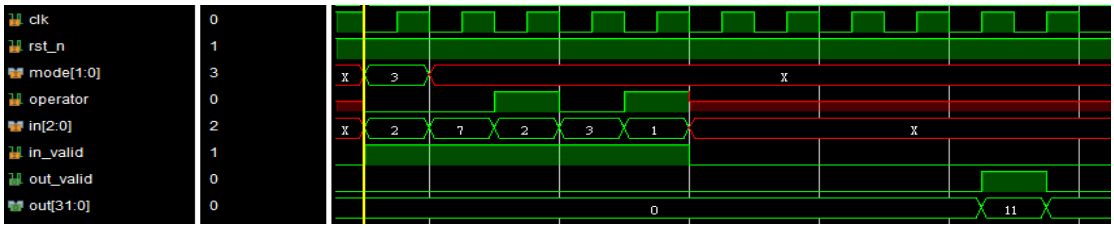
case 3

If the “**mode**” signal is 2’d2, prefix notation will be chosen. The calculation method is the same as **case4**.

case 4

If the “**mode**” signal is 2’d3, postfix notation will be chosen. After the calculation, you will only get one answer.

For example, if the "**in_valid**" signal is high for 5 continuous cycles, the "**mode**" signal is 2'd3 at the first cycle. The "**in**" signal is 3'd2, 3'd7, 3'd2, 3'd3, 3'd1 and the “**operator**” signal is 1’b0, 1’b0, 1’b1, 1’b0, 1’b1 in each cycle. In this case, the input signal represents a postfix expression "27＊3－". After calculation, the answer will be "11". The "**out**" signal will be 32'd11 for one cycle, and the "**out_valid**" signal will also be high for one cycle.



III. I/O Description

Signal Name	I/O	Width	Simple Description
clk	I	1	Posedge triggered Clock
Rst_n	I	1	Asynchronous active－low reset
mode	I	2	0：prefix and sorting in descending order. 1：postfix and sorting in ascending order 2：normal Polish notation(NPN) 3：reverse polish natation(RPN)
Operator	I	1	1'b0：the "**in**" signal is an operand 1'b1：the "**in**" signal is an operator
In(**unsigned**)	I	3	The "**operator**" signal is 1'b0： 3'b000~3'b111：represent the unsigned Number between 0~7 The "**operator**" signal is 1'b1： 3'b000：＋ => a ＋ b 3'b001：－ => a － b 3'b010：＊ => a ＊ b 3'b011：$ => | a ＋ b |
in_valid	I	1	High when in is valid.
Out_valid	O	1	High when **out** is valid.
Out(**signed**)	O	32	The answer of output after calculation

IV. Specifications

1. All output signals should be reset after the "**rst_n**" signal is asserted.
2. The "**out_valid**" signal should not be high when "**in_valid**" signal is high.
3. The "**out**" signal should be 0 when your "**out_valid**" signal is pulled down.
4. Each of the pattern execution latency is limited in 1000 cycles.
5. When the "**mode**" signal is "0,1", the duration of the "**out_valid**" signal should be one-third of the duration of the "**in_valid**" signal. Otherwise, the "**out_valid**" signal can only be high for one cycle.
6. While the "**out_valid**" signal is high, TESTBED will check if your "**out**" signal matches the expected value. If it matches, nothing will happen until you pass the pattern. If it does not match, an error message will tell you the expected output value, and at the end of the message, it will tell you the total number of errors. Your grade will depend on this.
7. Please upload the file to ilearning 3.0 with the format "PN_studentID.v" (including all your modules). If not followed, the score will be discounted by 30%

8. The “**in**” signal is given unsigned, but the “**out**” signal is a signed number because the answer may be negative.
9. The next round of the game will come in 2~4 negative edge of clk.
10. If the “**mode**” signal is 0 or 1, the “**in_valid**” signal will randomly raise up for a continuous period of 6, 9, or 12 cycles.
11. If the “**mode**” signal is 2 or 3, the “**in_valid**” signal will randomly raise up for a continuous period of 5, 7, or 9 cycles.
12. Do not revise the don’t touch TESTBED code, or you may fail this lab.

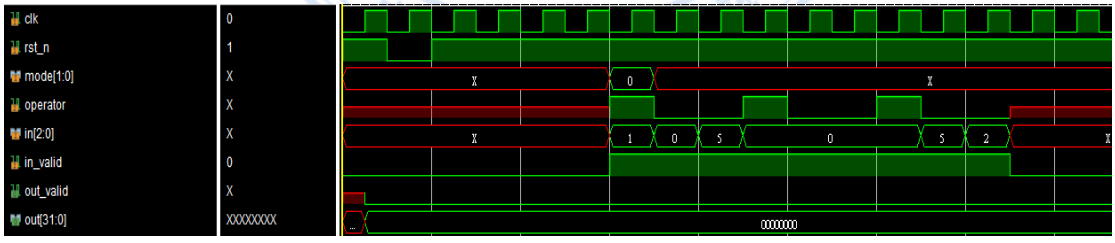
V. Grading Policy

- Violate the Specifications 1~5……0%
- If the total amount of error is less than 498 in first state……...25%
- If the total amount of error less than 300 in first state……50%
- If you pass the first state task but fail in second state……75%
- Without any error……100%

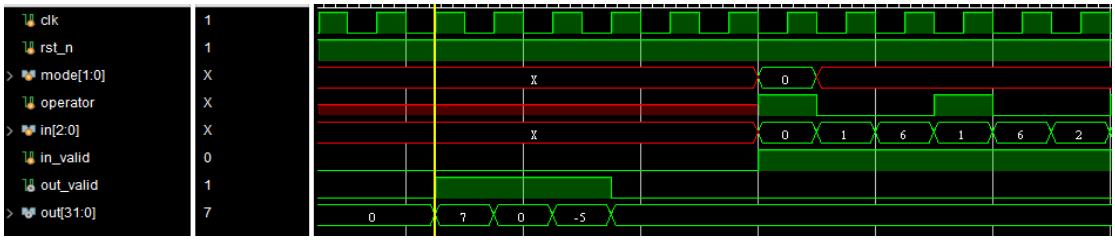
If you write a document explaining how you constructed this lab or what algorithm you chose, you will receive additional bonus points(At most 8%). For example, you may have chosen bubble sort as your sorting algorithm because it is more hardware-friendly, or designed a 4-state FSM for a specific reason. The document can only be one page of A4 in PDF format, and please upload it to ilearning 3.0 with the following format: "PN_studentID.pdf".

VI. Wave Demonstration

Input wave



Output wave



VII. TA Remind ♥♥♥

1. Bubble Sort
   https://ithelp.ithome.com.tw/articles/10195078
2. Bitonic Sort
   https://github.com/mcjtag/bitonic_sorter
3. Insertion Sort
   https://ithelp.ithome.com.tw/articles/10195269
4. Evaluate Reverse Polish Notation using a Stack
   https://stevenpcurtis.medium.com/evaluate-reverse-polish-notation-using-a-stack-7c618c9f80c0