Preferred Networks Internship 2020 Chip Topic 25 and 26 selection task

For the chip theme, please choose one of the following three choices.

1. Task 1,2,3 in this PDF
2. (Q1,Q2,Q3 of the coding task in `README-en.pdf`) + (Task 3 in this PDF)
3. (Task 1 in this PDF) + (Any one of Q1, Q2, or Q3 from the coding task in `README-en.pdf`) + (Task 3 in this PDF)

If you'd like to solve the coding tasks in `README-en.pdf`, please use Python3, C, C++, Rust or Perl as programming language.

Please refer to `README-en.pdf` for notes, submissions and inquiries.

First of all

If you cannot access EDA tools, please consider using publicly accessible tools such as edaplayground.com. If you're using SystemVerilog or Verilog, you can also use Verilator together with GTKWave. Note that the software and web sites are not related to PFN and we can not answer any questions regarding them.
https://www.edaplayground.com/
https://www.veripool.org/wiki/verilator/
http://gtkwave.sourceforge.net/

Task 1: A simple FIFO

Task 1.1:

FIFOs play a vital role in communication and synchronization. Write a parameterizable FIFO in a HDL of your choice. While the component is rather elementary, think about your choices especially in regard to what flag signals you forward outside the module.

Task 1.2:

Write a testbench and test your FIFO.

Submission

- Please submit your FIFO RTL and your testbench RTL
- A screenshot showing the waveform of your testbench documenting that your FIFO is working properly

Task 2: A queue of two FIFOs

Task 2.1: Create a queue of two FIFOs

FIFOs are often used to control communication flows between different components inside a chip. In such situations it is essential that no data is lost. Create a queue consisting of two FIFOs with a depth of four entries. The queue has a particular protocol, it accepts a data word every clock cycle unless it signals to stop.

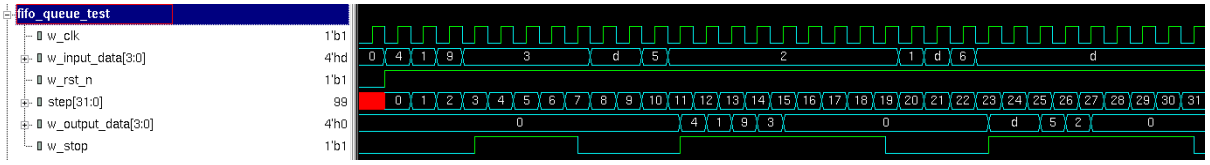
The queue's protocol is as follows:
FIFO1 is not full: Accept inputs
FIFO1 is full and FIFO2 is empty: Stop to accept inputs (STOP=1) and pop all entries of FIFO1 and push them into FIFO2. Once FIFO1 is empty, resume accepting inputs (STOP=0).
FIFO1 is full and FIFO2 is full: Stop to accept inputs (STOP=1), first output all entries of FIFO2 to make room for FIFO1's entries, and then transfer all entries of FIFO1 into FIFO2. After that the pipeline resumes to accept inputs (STOP=0).

Add a stop output to signal the sender to stop sending data. Any inputs can be ignored as long as we signal to stop. However, we want to make waiting times as short as possible, so don't signal longer than necessary.

Task 2.2: Create a testbench and test your queue

Create a testbench that follows the above protocol. I.e. input a new data word every clock cycle as long as the queue doesn't signal to stop.

The testbench waveform should look like the figure below.



Here is also an exemplary output of our implementation:

```
0 Pushing 4 stop: 0 out: 0
1 Pushing 1 stop: 0 out: 0
2 Pushing 9 stop: 0 out: 0
3 Pushing 3 stop: 0 out: 0
4 Waiting... stop: 1 out: 0
5 Waiting... stop: 1 out: 0
6 Waiting... stop: 1 out: 0
7 Waiting... stop: 1 out: 0
8 Pushing 13 stop: 0 out: 0
9 Pushing 13 stop: 0 out: 0
10 Pushing 5 stop: 0 out: 0
11 Pushing 2 stop: 0 out: 0
12 Waiting... stop: 1 out: 4
13 Waiting... stop: 1 out: 1
14 Waiting... stop: 1 out: 9
15 Waiting... stop: 1 out: 3
16 Waiting... stop: 1 out: 0
17 Waiting... stop: 1 out: 0
18 Waiting... stop: 1 out: 0
19 Waiting... stop: 1 out: 0
20 Pushing 1 stop: 0 out: 0
21 Pushing 13 stop: 0 out: 0
22 Pushing 6 stop: 0 out: 0
23 Pushing 13 stop: 0 out: 0
24 Waiting... stop: 1 out: 13
25 Waiting... stop: 1 out: 13
26 Waiting... stop: 1 out: 5
27 Waiting... stop: 1 out: 2
28 Waiting... stop: 1 out: 0
29 Waiting... stop: 1 out: 0
30 Waiting... stop: 1 out: 0
31 Waiting... stop: 1 out: 0
```

Hint: The first waiting period takes only 4 cycles as FIFO2 is empty at first. After the first time that FIFO1's contents are transferred to FIFO2, FIFO2 will always be full so it will have to output first for four cycles before FIFO1 can transfer its contents to FIFO2 for another four cycles (eight cycles in total).

Submission

Please submit

- The RTL of your queue and your testbench
- A screenshot showing the waveform of your testbench, documenting proper operation.

- (Optional) If you have any supplementary information like a FSM drawing or anything that documents your thought process, that's very welcome!

Task 3: Mock paper review (Topic 25: Next-generation Architectures for Highly Efficient Learning and Inference)

Please choose an architecture paper that has been presented e.g. at one of the conferences below:
ISCA, HPCA, ASPLOS, MICRO, DATE, DAC, ASP-DAC, ICCAD, VLSI (other conferences are fine as well)

If you cannot decide which one to review, take a look at the papers below.
[1] Schuiki, Fabian, Michael Schaffner, and Luca Benini. "Ntx: An energy-efficient streaming accelerator for floating-point generalized
reduction workloads in 22 nm fd-soi." 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2019. https://arxiv.org/pdf/1812.00182
[2] B. Zimmer et al., "A 0.32–128 TOPS, Scalable Multi-Chip-Module-Based Deep Neural Network Inference Accelerator With Ground-Referenced Signaling in 16 nm," in IEEE Journal of Solid-State Circuits, vol. 55, no. 4, pp. 920-932, April 2020.

After you decided on a paper, please conduct the mock review by first summarizing the main idea. The entire report shouldn't exceed 2 A4 pages in English (1 A4 page in Japanese) and should reflect the points below.

- Name 3 advantages of the paper's approach
- Name 3 disadvantages of the paper's aproach
- How does this approach impact PPA (power consumption, performance, area)
- Are there any points you'd improve and how
- What next step would you take, can you think of something

Submission

- Your mock review

Task 3: Mock paper review (Topic 26: Machine Learning and EDA)

Please choose an EDA paper that has been presented e.g. at one of the conferences below:
ISCA, HPCA, ASPLOS, MICRO, DATE, DAC, ASP-DAC, ICCAD, VLSI

If you cannot decide which one to review, take a look at the papers below.
[1] Han, Seung-Soo, et al. "A deep learning methodology to proliferate golden signoff timing." 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2014. https://vlsicad.ucsd.edu/Publications/Conferences/311/c311.pdf
[2] Ma, Yuzhe, et al. "Understanding Graphs in EDA: From Shallow to Deep Learning." Proceedings of the 2020 International Symposium on Physical Design. 2020. http://www.cse.cuhk.edu.hk/~byu/papers/C94-ISPD2020-GCN.pdf

After you decided on a paper, please conduct the mock review by first summarizing the main idea. The entire report shouldn't exceed 2 A4 pages in English (1 A4 page in Japanese) and should reflect the points below.

- Name 3 advantages of the paper's approach
- Name 3 disadvantages of the paper's aproach
- How does this approach impact PPA (power consumption, performance, area) or algorithm runtime and complexity
- Are there any points you'd improve and how
- What next step would you take, can you think of something

Submission

- Your mock review