2024 Digital IC Design

Homework 2: Check-in and Pickup

1. Introduction:

Speaking of traveling, it is convenient to fly to your destination. Check-in can help the airline maintain order, especially with a significant number of passengers filling the entire airport. Also, baggage check-in provides passengers with a more comfortable trip. Therefore, pickup occurs every time after a complete flight. This homework requires you to design a finite-state machine and simulate a simple check-in and Pickup process.

2. Specification:

2.1. The CIPU

The illustration is shown in Fig 1, and the circuit architecture is shown in Fig 2. The CIPU system includes two kinds of data structures: the First-in, First-out (FIFO) and Last-in, First-out (LIFO). We instantiate two FIFOs and one LIFO for 1. passenger check-in, 2. baggage pickup, and 3. baggage check-in. Notice that the input of one FIFO is the output of the LIFO, just like the luggage conveyor.

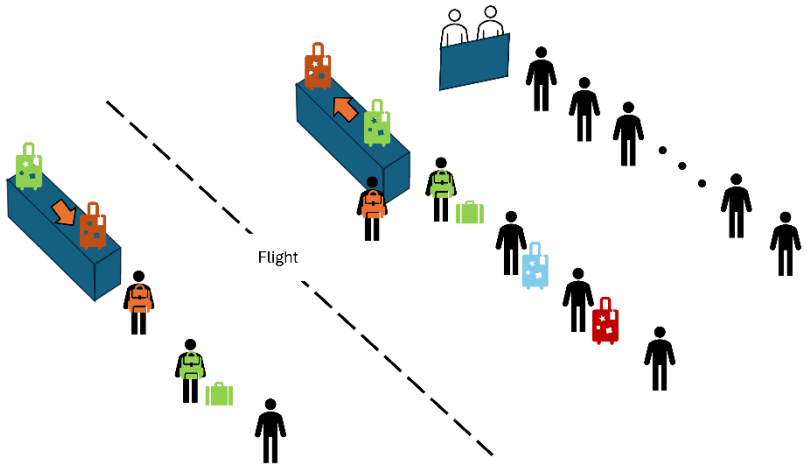


Fig. 1 Illustration of CIPU

FIFO and LIFO have length limits, which help us simulate the reality of the scenario in one airline. In addition, to simplify the CIPU

problem, this system is a session-based design. In other words, a non-pipelined design is sufficient for this problem.

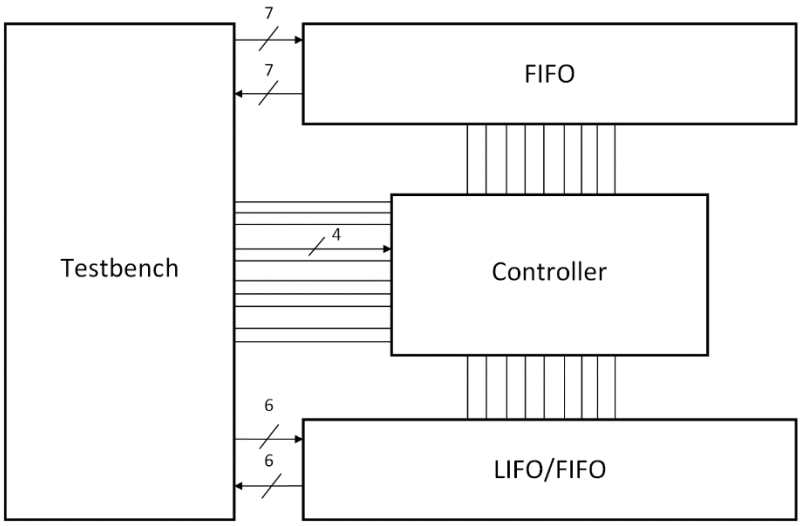


Fig. 2 Circuit Architecture of CIPU

The specifications your design needs to meet are listed in Table I and Table II. Table I provides the details of FIFO and LIFO, and Table II shows the I/O specifications.

Table I FIFO/LIFO description

Name	Length	Description
FIFO	16	The one comes first at head, first leave at tail.
LIFO	16	The last one comes at head, first leaves at head.

Table II I/O specification of the CIPU

Signal	I/O	width	Description
clk	I	1	Input port connected to clock source. Active when positive edge.
rst	I	1	Input port to reset the circuit. Active when positive edge, synchronous reset.
people_thing_in	I	8	“1” ~”9” stands for baggage.

			“A” ~” Z” stands for passengers. “$” stands for the terminated symbol. This port will transmit the information with order of baggage->passengers->baggage….
people_thing_out	O	8	Serial output passengers with the correct order.
ready_fifo	I	1	The signal will inform your design to prepare for the data flow. (active high)
valid_fifo	O	1	The signal tells the testbench your data is valid, and it will be checked. (active high)
done_fifo	O	1	The signal tells the testbench the FIFO finishes outputting. (active high)
thing_in	I	8	“1” ~”9” stands for baggage. “;” stands for separate symbol. “$” stands for the terminated symbol.
thing_out	O	8	Serial output baggage with the correct order.
thing_num	I	4	The signal tells your design how much baggage should be popped back.
ready_lifo	I	1	The signal will inform your design to prepare for the data flow. (active high)
valid_lifo	O	1	The signal tells the testbench your data is valid, and it will be checked. (active high)
done_thing	O	1	The signal tells the testbench that the designated baggage has been popped back and is prepared to transmit the next baggage. (active high)

done_lifo	O	1	The signal tells the testbench the LIFO finishes outputting. (active high)
valid_fifo2	O	1	The signal tells the testbench your data is valid, and it will be checked. (active high)
done_fifo2	O	1	The signal tells the testbench the FIFO finishes outputting. (active high)

2.2. CIPU Process

First, passengers will line up before the check-in counter. The baggage is coded as "1" ~ "9", and the passengers are coded as "A" ~" Z". The ASCII code table can help you check the value of these characters. Passengers with their baggage will be sent from the testbench file to your design with the order of "baggage, passengers, baggage, passengers, …". This is also called a session. Each session's sequence started with the baggage number and ended with the "$" symbol. For example,

"1", "2", "X", "1", "7", "B", …"3", "A"," $" ----- session 1

"4", "7", "R", "Y", "6", "8", "1", "9", "F"," $" -----session 2

The rule of our check-in FIFO is to output the check-in order only to the passengers. Therefore, your FIFO output should be:

"X", "B", …, "A". ---- session 1

"R", "Y", "F" ---- session 2

Second, you must maintain a LIFO to handle how much luggage is pushed into the conveyor (stack) and how much is popped to the owner. For your convenience, there are two signal ports in your design. One is to provide the pop number, and the other can handle the luggage information from the testbench file instead of the passengers' FIFO results. As a result, to some extent, you can design a parallel CIPU to handle the passengers' FIFO and baggage LIFO. Notice that ";" is a separate symbol among passengers' baggage, and "$" is the end symbol for a session. For example,

"1", "2", ";", "1", "7", ";", …, "3",";", "$" ---- session 1

1, 1, …, 0, ---- session 1 pop number

"4", "7", ";", "6", "8", "1", "9", ";", "$" ---- session 2

1, 3 ---- session 2 pop number

The rule of this LIFO is that your design can use the signal to control the testbench file, whether it sends the data or not. When you encounter the "**;**", the testbench files will automatically stop sending the data until you output the done signal. The pop number informs you how much luggage will be popped out. This signal value will change immediately after your design restarts the testbench. With the above session examples, your output will be:

"2", "7", … (with a stack containing "1", "1", …)

"7", "9", "1", "8" (with a stack containing "4", "6")

Last, we will check that the stack you maintain has the correct value and its corresponding order. Our check rule is that the testbench will compare the result output from the tail of the stack. You can see it as a FIFO but sharing the same memory space with the stack discussed above. Therefore, the output will be:

"1","1", … (with an empty stack)

"4","6" (with an empty stack)

Notice that each valid signal should be carefully handled when you finish outputting the results. In other words, you should pull down the signal to prevent the testbench from checking the wrong target. In addition, in this homework, if the thing_num is "0", let the 0 be the value of thing_out. This example is shown in Fig 7.

2.3. Timing diagrams

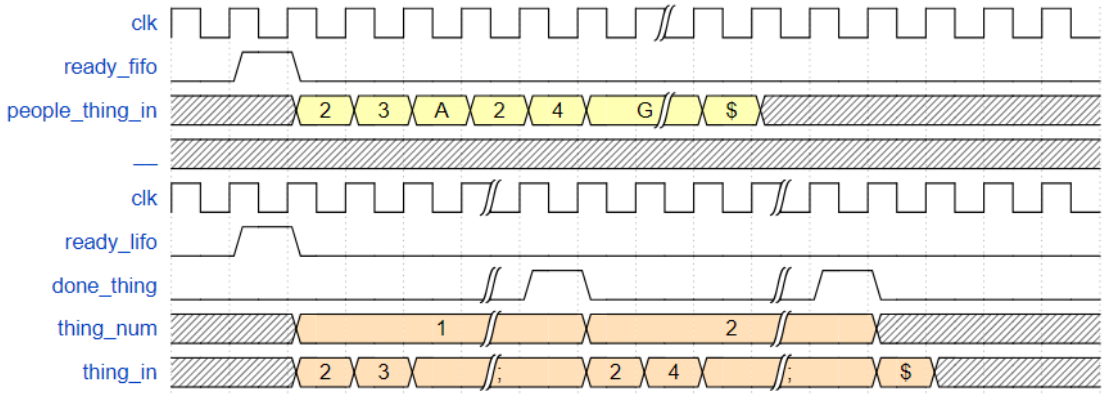


Fig. 3 FIFO/LIFO Read operation.

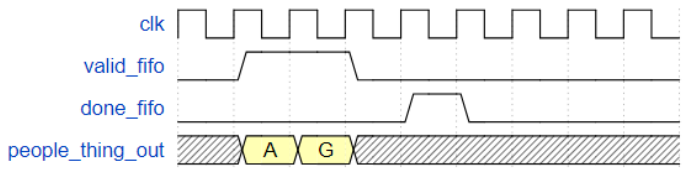


Fig. 4 FIFO (passenger) Write operation.

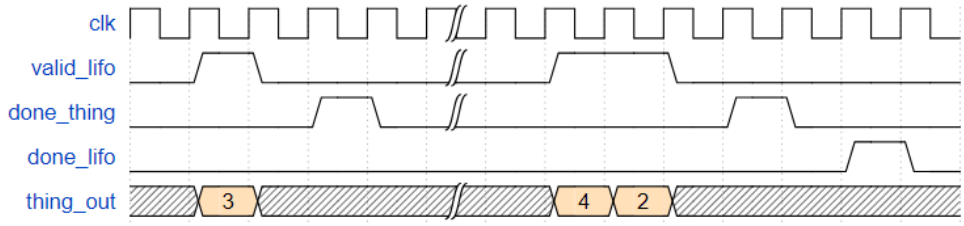


Fig. 5 LIFO Write operation.

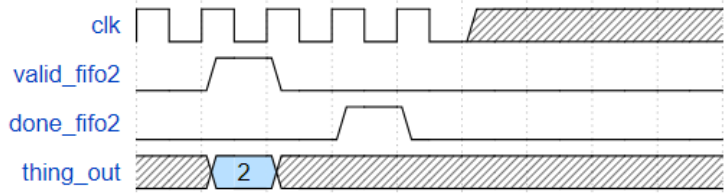


Fig. 6 FIFO (stack) Write operation.

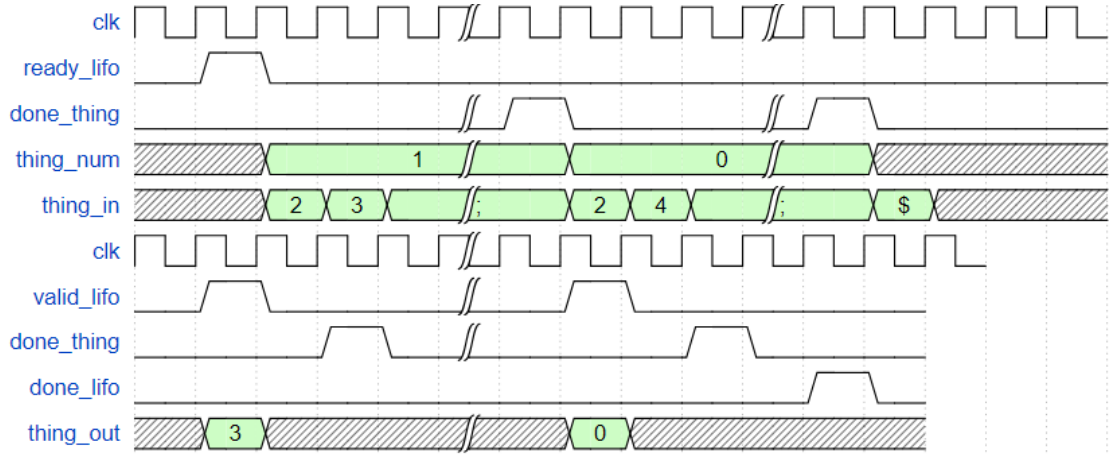


Fig. 7 LIFO Zero thing_num case

3. Scoring:

3.1. FIFO **[40%]**

We will check the FIFO output from the port “people_thing_out”. The results should be generated in the correct order, and you will get the following message in ModelSim simulation.

```
There are total   0 errors in FIFO  !!
```

3.2. Stack **[30%]**

We will check the LIFO output from the port “thing_out”. The results should be generated in the correct order corresponding to the thing_num, and you will get the following message in Modelsim simulation.

```
There are total   0 errors in LIFO  !!
```

3.3. CIPU **[30%]**

We will check the LIFO output from the port “thing_out”, but the order of the results should be followed FIFO behavior. You will get the following message in Modelsim simulation.

```
There are total   0 errors in FIFO2 !!
```

If you have your circuit well designed, the successful message would be

```
*****************************
**                         **       |__||
**  Congratulations !!     **      / 0.0  |
**                         **    /_____   |
**  Simulation PASS!!      **   /^ ^ ^ \  |
**                         **  |^ ^ ^ ^ |w|
*****************************   \m___m__|_|

Correct / Total :   100 / 100
```

Otherwise, you can also check how much score you have with following message

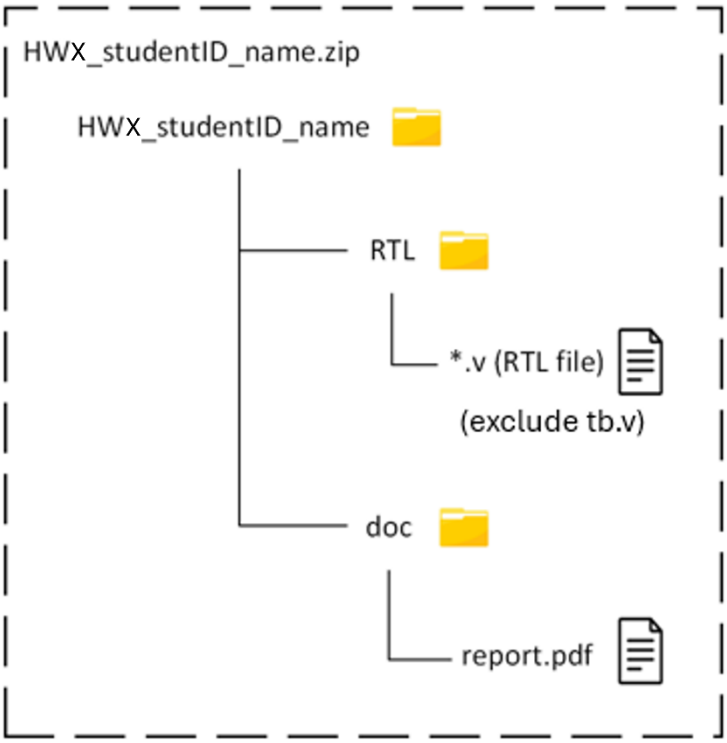
```
There are total  10 errors in FIFO2 !!


        *****************************
        **                         **       |__||
        **  OOPS!!                 **      / X,X  |
        **                         **    /_____   |
        **  Simulation Failed!!    **   /^ ^ ^ \  |
        **                         **  |^ ^ ^ ^ |w|
        *****************************   \m___m__|_|

        Correct / Total :    70 / 100
```

4. Submission

4.1. Submitted files

Please submit your files, followed by the illustration below.

For example, HW2_P78121506_廖國佑.zip. (Note: if your operating system is MACOS, please remove the directory "__MACOSX" when compressing your files.



Report file

Please follow the specifications of the report. You are asked to describe how the circuit is designed as detailed as possible.

4.2. Note

Please submit your .zip file to folder HW2 in moodle.

Deadline: 2024/4/29 23:55 p.m.

If you have any problem, please contact TA by email

diclab65a01tas@gmail.com