2024 Digital IC Design Homework II

NAME	楊晴雯
Student ID	P76114511

Functional Simulation Result

FIFO Pass	LIFO Pass	CIPU Pass

Stage 1,2,3

```
Transcript
# Loading work.Fifo2Controller
VSIM 110> run -all
# ************************************************
# ***********      Simulation Start      ***********
# ************************************************
#
#
# There are total   0 errors in FIFO  !!
#
#
# There are total   0 errors in LIFO  !!
#
#
# There are total   0 errors in FIFO2 !!
#
#
#         ***************************
#         **                       **       |__||
#         **  Congratulations!!     **      / O.O  |
#         **                       **    /_____   |
#         **  Simulation PASS!!     **   /^ ^ ^ \  |
#         **                       **  |^ ^ ^ ^ |w|
#         ***************************   \m___m__|_|
#
#         Correct / Total :   100 / 100
#
#
# ** Note: $finish    : /home/nanaeilish/courseworks/IC-Design/hw2/ref/tb.sv(301)
#    Time: 1295 ns  Iteration: 1  Instance: /textfixture
# 1
```

Description of your design

My directory structure is

ref

```
├── CIPU.v
├── Fifo2Controller.v
├── FifoController.v
├── FIFO.v
├── LifoController.v
├── LIFO.v
├── LIFO.v.bak
└── tb.sv
```

CIPU instantializes 1 FIFO, 1 FifoController for passenger check-in; 2 LIFOs and 2 Controllers for each, in baggage check-in and pick-up.

The Fifo and Lifo takes a reset signal, a clock, data_in and outputs a data; it uses

write_enable, read_enable signals to controlswhether to (1) do nothing (IDLE) (2) push into or (3) pop from the storage. To sum up, the Fifo and Lifo here are simple synchronous-reset data structures and do not involve control-decision logic.
The control signals are decided by the corresponding controller, which is also the most challenging part of hw2. All controllers strictly follow the 2 Comb. 1 Seq. Logics taught in the lecture.

1. **FifoController**: I have a module called “TypeCheck” to check the data types: if the data is a passenger, it sends signals of PUSH state, otherwise it IDLEs. When the final $ is sent in, the process goes into DONE_INPUT state and prepare for the passenger outputting, finally I have POP and DONE_OUTPUT states.
Note that after the signals sent, the corresponding fifo takes another cycle to output or write in, therefore the valid_fifo signal should be delayed 1 more cycle (I work around this by letting the previous state of the first POP to output data, and activate the valid_fifo in the first POP state; not the best practice I guess).
2. **LifoController**:
I use 8 states for this controller:

```
parameter[2:0] IDLE = 3'b000, PUSH = 3'b001, POP = 3'b010;
parameter[2:0] POP_ZERO = 3'b011, DONE_THING_INPUT = 3'b100, DONE_THING_OUTPUT = 3'b101, DONE_LIFO = 3'b110, POP_FIRST = 3'b111;
parameter [DATA_WIDTH-1:0] ENDSIGN = 8'h24;
parameter [DATA_WIDTH-1:0] SEPSIGN = 8'h3b;
// DONE_THING_INPUT: all luggages of the passenger go in and the number of luggages in thing_num are prepared to be popped
// DONE_THING_OUTPUT: all luggages of the passenger are checked and the correct number of luggages are popped
// DONE_LIFO: all passengers' luggages are checked and taken away.
// POP_FIRST: the first luggage of the passenger is popped; this is used for valid_lifo signal
// POP_ZERO: no luggage is popped; this is used for output_zero signal
```

This controller outputs `done_thing, output_zero, wr_enable, rd_enable, valid_lifo, done_lifo` signals. Since the thing_num (luggage to be popped) check is done here, whether to output a zero is also decided here by output_zero. I copy the `thing_num` into an internal variable `pop_num` for counting down; and note that it should be decremented “by clock posedge”, therefore **`pop_num`’s decrement procedure should be put into the state-transition (sequential logic) always block** instead of the other two.

3. **Fifo2Controller**: this controller is quite like the FifoController, except that its ready signal is not given in testbench; I use the done_lifo signal to replace. Furthermore, to simulate a FIFO behavior at this stage, we actually need another LIFO to reverse the first LIFO’s storage. Ideally, these two LIFOs could share the same memory space; but due to wire connection I simply use 2 LIFOs.