2023 Digital IC Design

Homework 2: Rails

1. Introduction

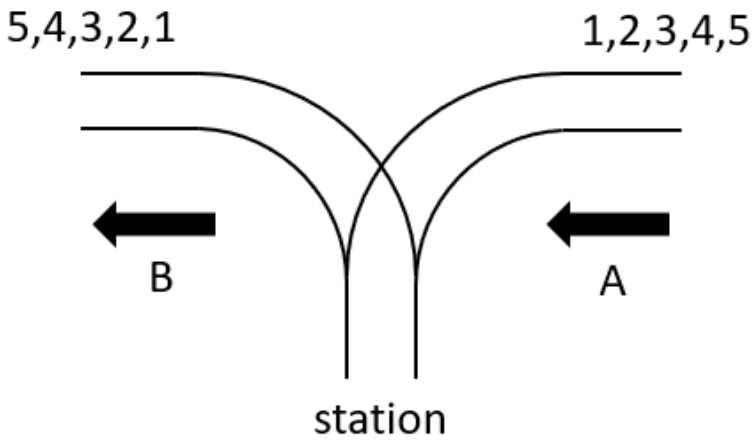

Fig. 1. Rails.

There is a railway station that established only a surface track. Furthermore, it has been discovered that the station can only be designed as a dead-end station, as shown in Fig. 1. Additionally, due to the limited space available, the station can accommodate only a single track. In this homework, you are required to design a circuit that decides whether it is possible to get the required order of coaches.

2. Design Specifications:

2.1 Block Overview

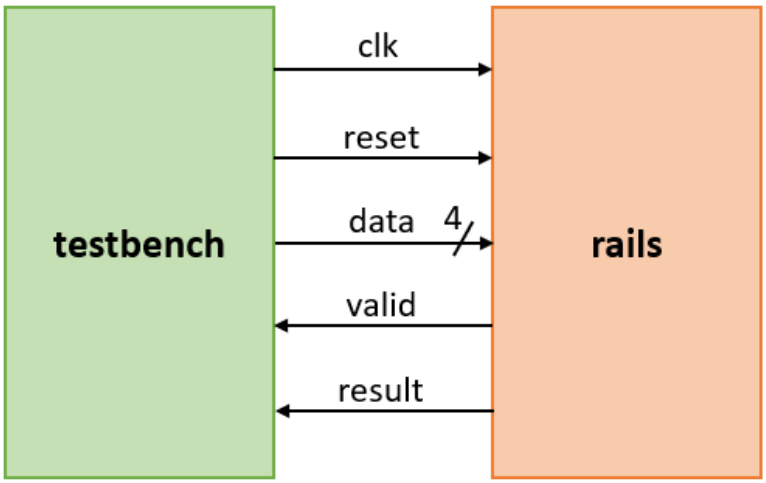

Fig. 2. The block overview.

2.2 I/O Interface

Table I. I/O interface.

Signal Name	I/O	width	Description
clk	I	1	This circuit is a synchronous design triggered at the positive edge of *clk*.
reset	I	1	Active-high asynchronous reset signal.
data	I	4	Data input.
valid	O	1	When output result, set *valid* signal to high. The testbench will check the output result when *valid* signal is high.
result	O	1	If the order is possible for the coaches continuing in direction B, ***result*** will output high; otherwise, it will output low

2.3 File Description

File Name	Description
rails.v	The module of rails, which is the top module in this design.
tb.v	The testbench file. The content in this file is **not allowed** to be modified.
test_data_rails.dat	Test data for rails verification.
golden_data_rails.dat	Golden data for rails verification.

3. Rails Description:

3.1 Description

Every train arriving from direction A has a variable number of coaches, ranging from 3 to 10, with each coach numbered in ascending order from 1 to N. The aim is to determine whether it is feasible to rearrange the coaches in a particular order, *a1*,*a2*, . . . , *aN*, before continuing in direction B. You can assume that individual coaches can be disconnected from the train before entering the station and can move independently to the track in direction B. Moreover, any number of coaches can be kept in the station at any given time. However, once a coach has entered the station, it cannot return to the track in direction A, and once it has left the station in direction B, it cannot return.

3.2 Main functions of rails

After reset, the ***data*** will first input the number of coaches for the first test

pattern, followed by the order of the coaches continuing in direction B in sequence each cycle. Once the testbench has finished inputting the input pattern, it will wait for the rails circuit to output a result. If the order is possible for the coaches continuing in direction B, ***result*** will output high; otherwise, it will output low. When the result is ready to be output, the ***valid*** signal must be set to high, and the ***result*** will be output in the same cycle. Then, in the next cycle, ***valid*** will be set back to low. After ***valid*** is low, the testbench will start inputting the next test pattern from the next cycle. To avoid testbench misjudgments, when the circuit is reset, ***valid*** must be set to low, and after each result is output, ***valid*** must be set to low again.

Hint: you can think of "station" as a stack.

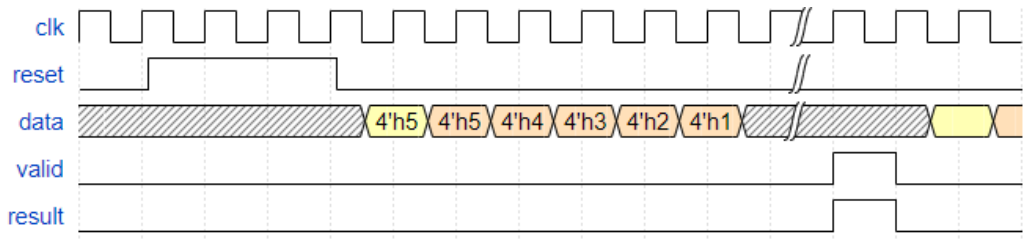

Fig. 3. Timing diagram of input and output

4. Scoring:

Functional Simulation [100%]

The grading rule for this homework is that one point can be obtained by passing one test pattern. If you run the simulation, you will get the score message in ModelSim. Please don't design specifically for the test pattern. Otherwise, you will get 0 point.

```
# ------------------------------------------------------------
# ----    Simulation finish,  ALL PASS,  Score = 100     ----
# ------------------------------------------------------------
```

Fig. 4. Simulation result for all passes.

```
# ------------------------------------------------------------
# -- Simulation finish,  Pass = 53 , Fail = 47, Score = 53 --
# ------------------------------------------------------------
```

Fig. 5. Simulation result for partial passes.

5. Submission:

5.1 Submitted files

You should classify your files into two directories and compress them to .zip format. The naming rule is HW2_studentID_name.zip. If your file is not named according to the naming rule, you will lose five points.

	RTL category
*.v	All of your Verilog RTL code
	Documentary category
*.pdf	The report file of your design (in pdf).

5.2 Report file

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

5.3 Note

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

Deadline: 2023/4/4 23:55

If you have any problem, please contact TA by email

p76114040@gs.ncku.edu.tw