NCTU-EE IC LAB – Fall 2023

Online test 2023/11/04

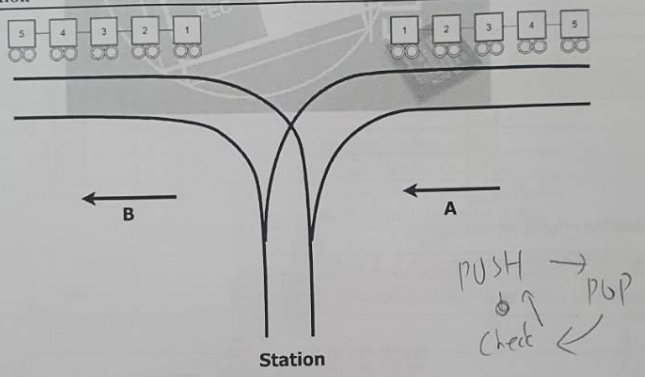
Design: Train departure problem

Data Preparation

Extract test data from TA's directory:

% tar -xvf ~iclabTA01/OT.tar

Design Description

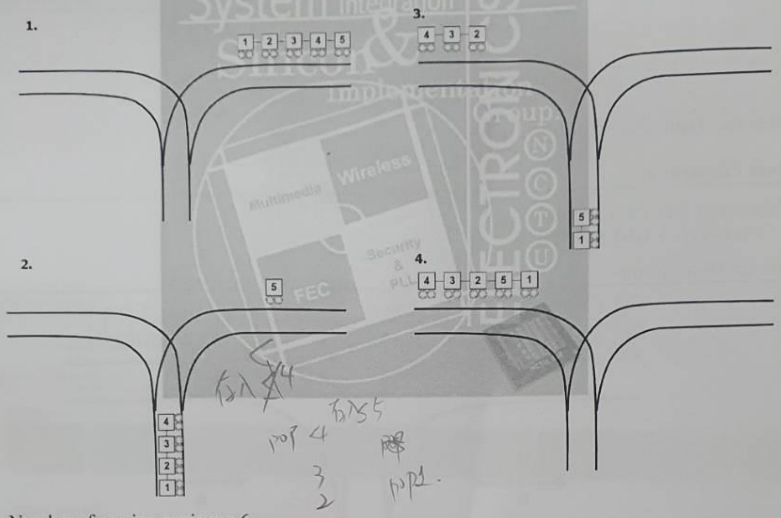


The train is approaching from the A direction and is set to depart from the B direction. It comprises N carriages ($3 \leq N \leq 10$), numbered sequentially from 1 to N. You can assume that each carriage can be separated from others before entering the station and can be individually positioned on the track leading to the B direction. The station can also accommodate all the carriages at any given time. However, once a carriage enters the station, it cannot revert to the A direction track, and once it departs towards the B direction, it cannot return to the station.

Your task is to create a circuit that determines if the train can be organized in a specific order on the B direction track.

Examples

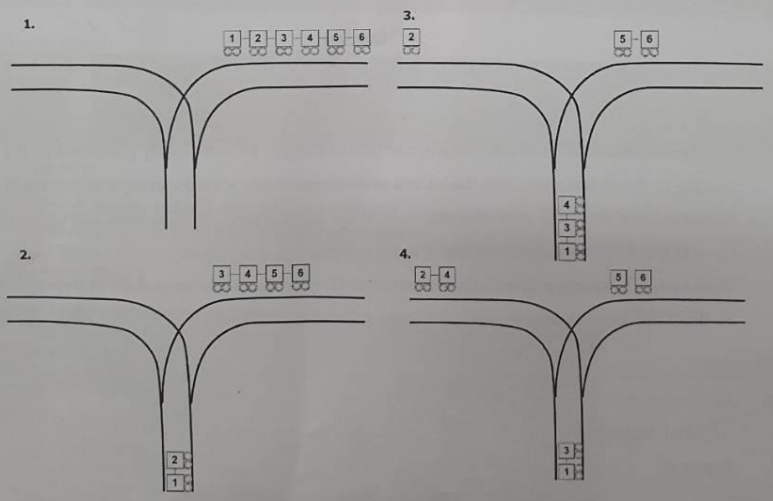
1. Number of coming carriages: 5 **(First cycle input)**
   Required departure order: 4, 3, 2, 5, 1 **(Other cycle input)**
   Result: 1



2. Number of coming carriages: 6

Required departure order: 2, 4, 1, 3, 6, 5

Result: 0



3. Number of coming carriages: 3
   Required departure order: 1, 2, 3
   Result: 1
4. Number of coming carriages: 4
   Required departure order: 2, 4, 3, 1
   Result: 1
5. Number of coming carriages: 5
   Required departure order: 4, 2, 3, 5, 1
   Result: 0

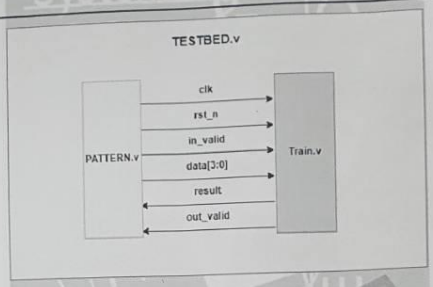
I/O specification

Signals name	Direction	Bit Width	Definition
clk	input	1	Clock.
rst_n	input	1	Asynchronous active-low reset.
in_valid	input	1	High when input signals are valid.
data	input	4	First cycle: Number of coming carriages. Other cycle: Required departure order.
out_valid	output	1	High when output is valid.
result	output	1	If the given departure order is possible to meet, result should be set as high. Otherwise, it should be set as low.

Specifications

1. Top module name：**Train** (Filename: **Train**.v)
2. It is an asynchronous reset and active-low architecture. If you use synchronous reset (reset after clock starting) in your design, you may fail to reset signals.
3. The clock period of the design is fixed to 10ns.
4. The next group of inputs will come in 2~5 cycles after your out_valid pull down.
5. The synthesis result of data type cannot include any LATCH.
6. After synthesis, you can check **Train**.area and **Train**.timing in the folder "Report".
7. The slack in the timing report should be non-negative and the result should be MET.
8. The gate level simulation cannot include any timing violation.
9. The latency of your design in each pattern should not be larger than 1000 cycles. The latency is the clock cycles between the falling edge of the **in_valid** and the rising edge of the **out_valid**.
10. Any words with "error", "latch" or "congratulation" can't be used as variable name.
11. The out_valid cannot be raised when in_valid is high.
12. The out_valid will be high only for 1 cycle.

Block Diagram



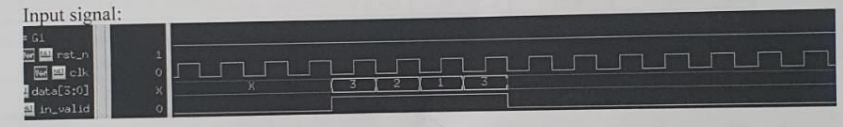
Grading policy

- **Functionality: 1de: 100%, Take home: 50% (Take home deadline: 11/05(Sun) 12:00)**

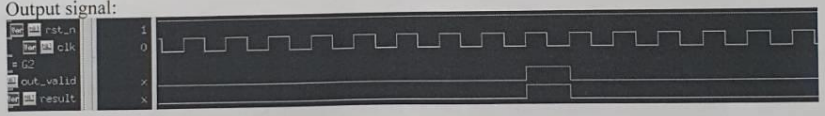
Sample Waveform

- Reset:

- Input signal:



- Output signal:



Submission step

For Group A, please submit your file to 1st_demo
For Group B, please submit your file to 2nd_demo
For Take home, please submit your file to 3rd_demo

1. **cd 09_SUBMIT**

```
1:42 iclabTA01@ee32[~/OT_test]$ ll
total 16
drwxr-xr-x 2 iclabTA01 iclab   147 Oct 30 09:35 00_TESTBED
drwxr-xr-x 2 iclabTA01 iclab   268 Oct 31 22:34 01_RTL
drwxr-xr-x 4 iclabTA01 iclab   267 Oct 31 22:34 02_SYN
drwxr-xr-x 2 iclabTA01 iclab  4096 Oct 31 22:34 03_GATE
drwxr-xr-x 2 iclabTA01 iclab    69 Oct 31 22:35 09_SUBMIT
-rw-r--r-- 1 iclabTA01 iclab 11338 Oct 31 21:09 demo.py
1:42 iclabTA01@ee32[~/OT_test]$ cd 09_SUBMIT/
1:42 iclabTA01@ee32[~/OT_test/09_SUBMIT]$
```