

# NCTU-EE IC LAB – Fall 2023

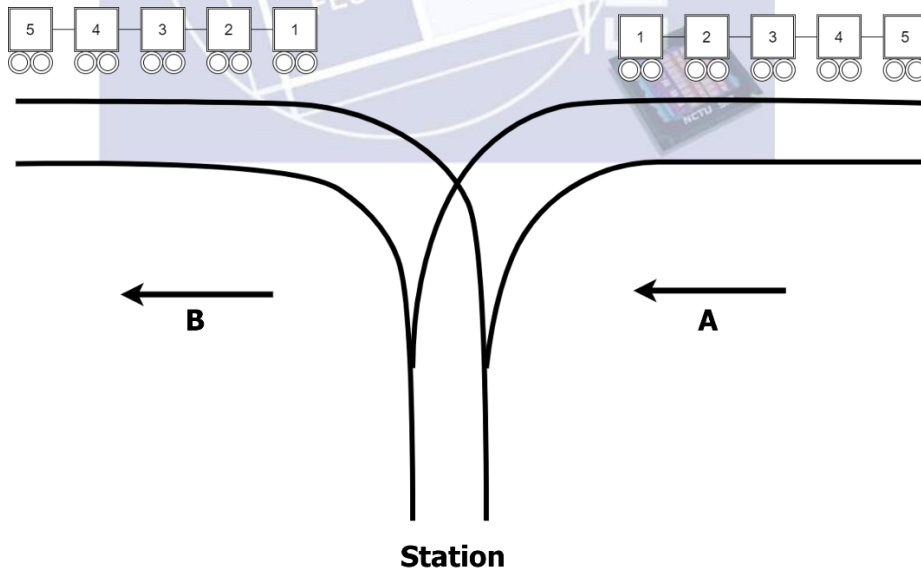
Online test 2023/11/04

## Design: Train departure problem

### Data Preparation

Extract test data from TA's directory:  
`% tar -xvf ~iclabTA01/OT.tar`

### Design Description

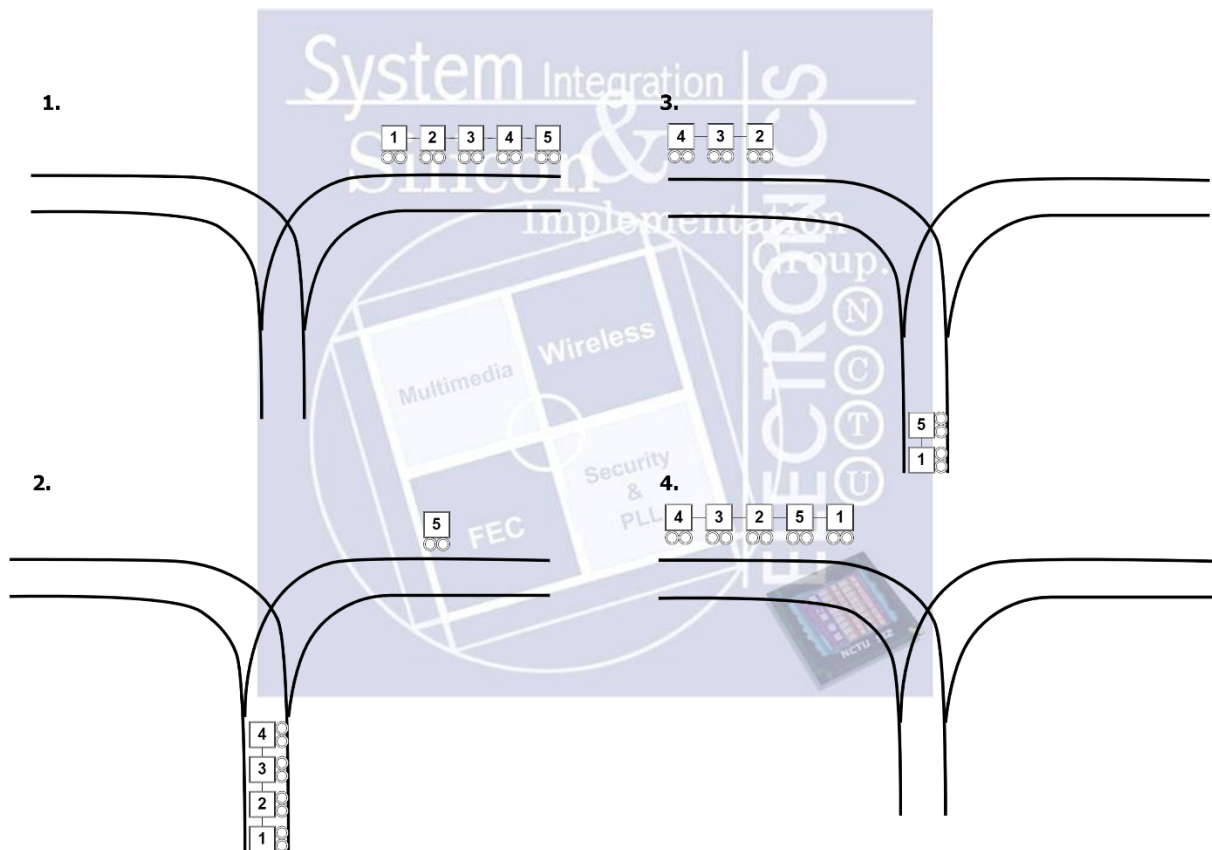


The train is approaching from the A direction and is set to depart from the B direction. It comprises  $N$  carriages ( $3 \leq N \leq 10$ ), numbered sequentially from 1 to  $N$ . You can assume that each carriage can be separated from others before entering the station and can be individually positioned 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.

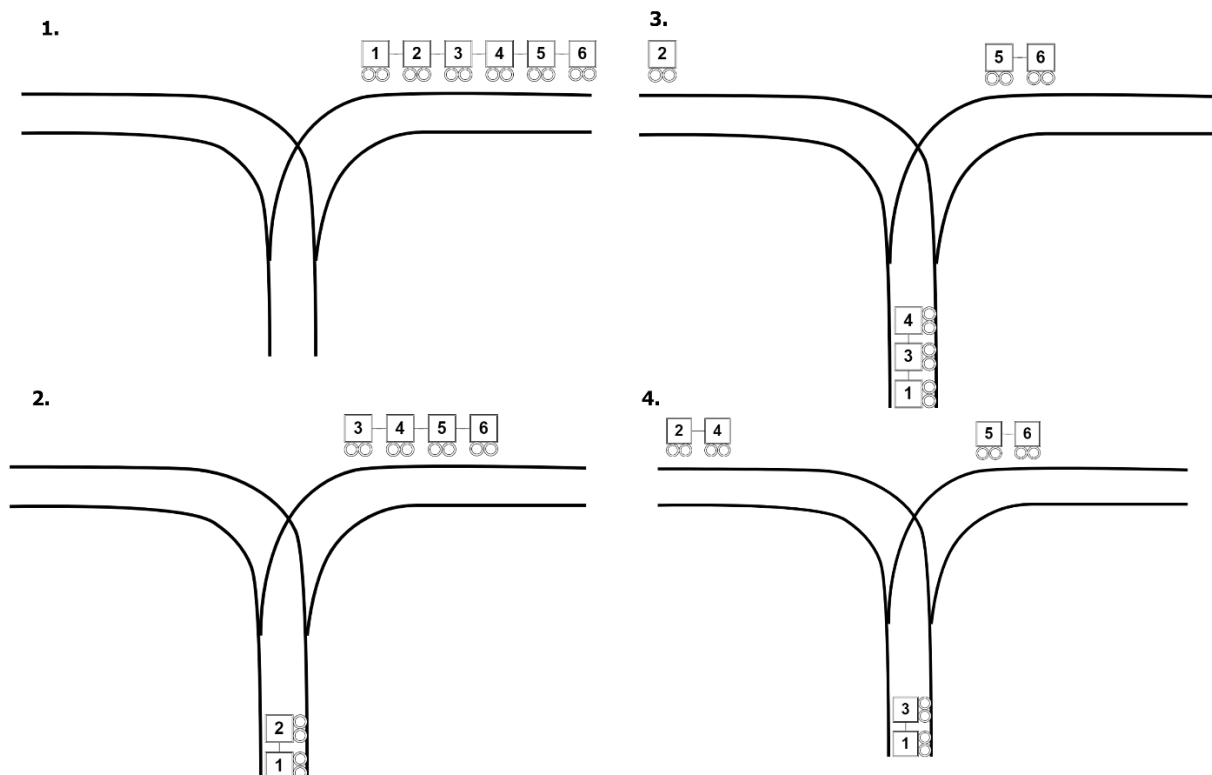
**Your task is to create a circuit that determines if the train can be organized in a specific order on the B direction track.**

### 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

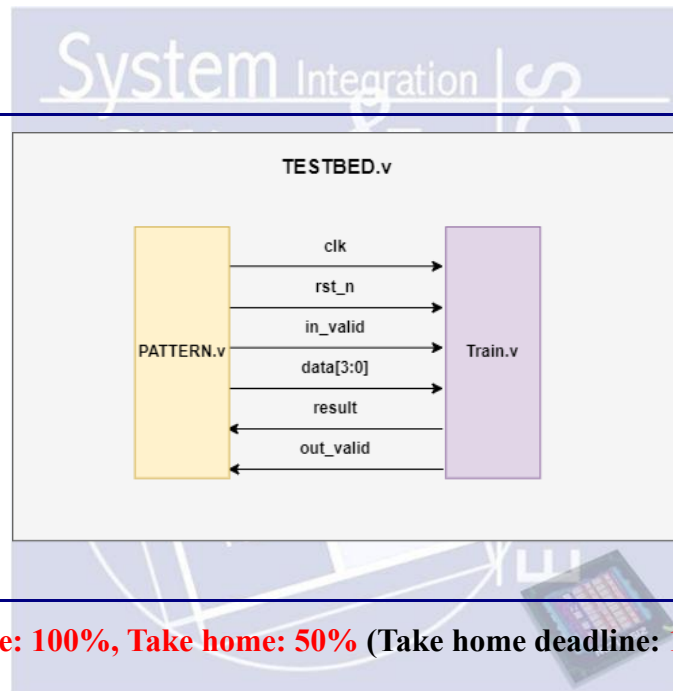
### I/O specification

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

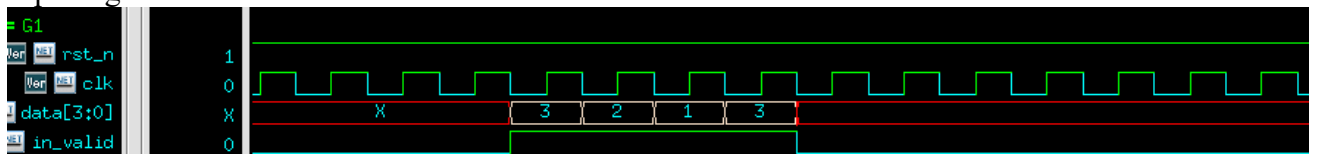
- **Functionality: 1de: 100%, Take home: 50% (Take home deadline: 11/05(Sun) 12:00)**

## Sample Waveform

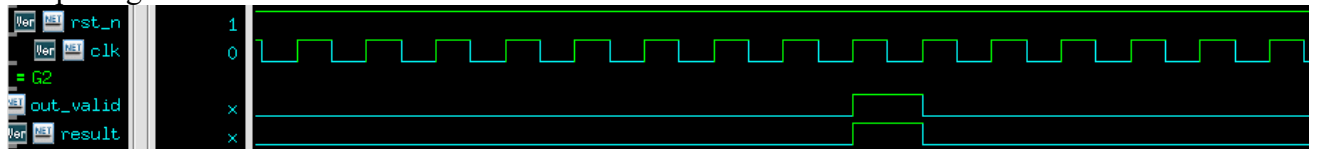
- Reset:



- 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[~/0T_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[~/0T_test]$ cd 09_SUBMIT/
1:42 iclabTA01@ee32[~/0T_test/09_SUBMIT]$
```

## 2. ./00 tar (tar all your design)

```
1:43 iclabTA01@ee32[~/OT_test/09_SUBMIT]$ ./00_tar 10
[Info] Directory created.
[Info] ../00_TESTBED/filelist.f has been copied.
[Info] ../01_RTL/Train.v has been copied.
[Info] 10_iclabTA01.txt has been copied.
cp: cannot stat '../04_MEM': No such file or directory
[Info] ../04_MEM folder has been copied.
[Info] Now start tar zcvf OT_iclabTA01.tar.gz
OT_iclabTA01/
OT_iclabTA01/filelist_iclabTA01.f
OT_iclabTA01/Train_iclabTA01.v
OT_iclabTA01/10_iclabTA01.txt
[Success] OT_iclabTA01.tar.gz Sucessfully.
```

## 3. ./01\_submit

(Auto demo by yourself, after this step, TA will receive your design file and demo result)

```
1:44 iclabTA01@ee32[~/OT_test/09_SUBMIT]$ ./01_submit
[Info] Deadline check OK ...
[Info] File check OK ...
[Info] Your file will be submitted to: 1st_demo
[Info] Are you sure you want to submit your design file?(y/n):y
[Info] Now submit OT_iclabTA01.tar.gz file to system.
[Success] Copying Successfully.
[Info] Now start demo ...
[Info] iclabTA01 01_RTL start
[PASS] iclabTA01 01_RTL
[Info] iclabTA01 02_SYN start
[PASS] iclabTA01 02_SYN
[Info] iclabTA01 03_GATE start
[PASS] iclabTA01 03_GATE
Account      Pass Message Files 01_RTL 02_SYN 03_GATE CT Latency \
0 iclabTA01 1st_demo NaN 0 0 0 0 NaN 8868
Area
0 13558.4
[Info] Demo finished, please check DEMO_RESULT.csv.
[Info] Now submit DEMO_RESULT.csv
[Info] Submit DEMO_RESULT.csv successfully
=====
Submit Report
=====
Result      : 1st_demo has been submitted.
Submission time : Wed Nov 1 01:44:51 CST 2023
Account,Pass,Message,Files,01_RTL,02_SYN,03_GATE,CT,Latency,Area
iclabTA01,1st_demo,,0,0,0,0,,8868.0,13558.406558
```

## 4. ./02 check (check whether you have successfully submitted the file)

```
1:46 iclabTA01@ee32[~/OT_test/09_SUBMIT]$ ./02_check
Please input argument.
Only accept 1st demo or 2nd demo! or 3rd demo!
1:47 iclabTA01@ee32[~/OT_test/09_SUBMIT]$ ./02_check 1st_demo
./OT_iclabTA01_1st_demo.tar.gz has been downloaded!
DEMO_RESULT_iclabTA01.csv has been downloaded!
Account,Pass,Message,Files,01_RTL,02_SYN,03_GATE,CT,Latency,Area
iclabTA01,1st_demo,,0,0,0,0,,8868.0,13558.406558
1:47 iclabTA01@ee32[~/OT_test/09_SUBMIT]$
```