NCTU-EE IC LAB – Fall 2017

Lab03 Exercise

Design: Magical Dartboard

Data Preparation

1. Extract test data from TA's directory:

   % tar xvf ~iclabta01/Lab03.tar

2. The extracted LAB directory contains:

   a. **00_TESTBED**

   b. **01_RTL**

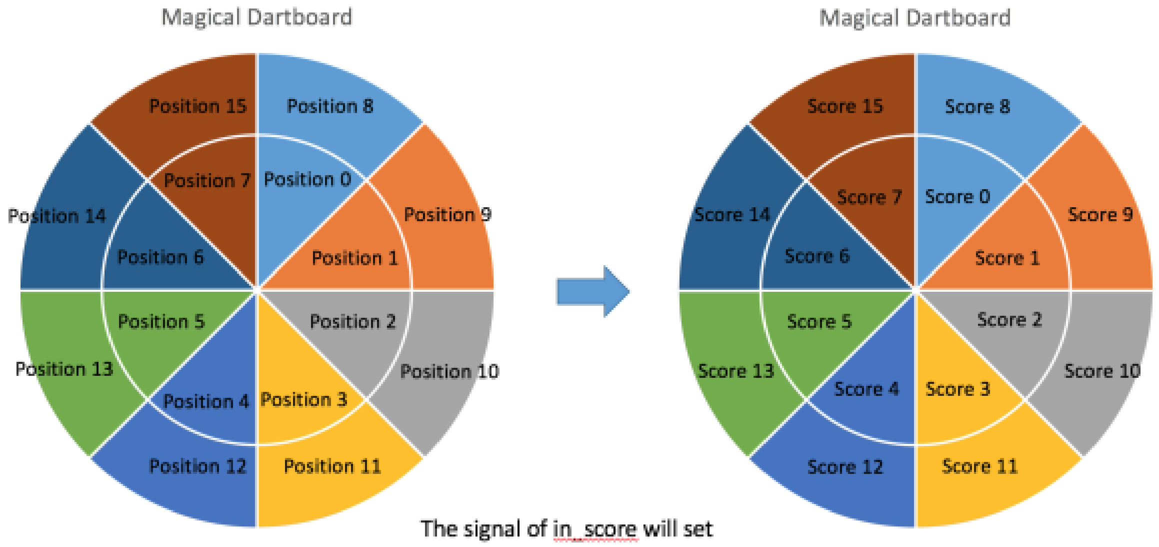
   c. **02_SYN**

   d. **03_GATE**

Design Description

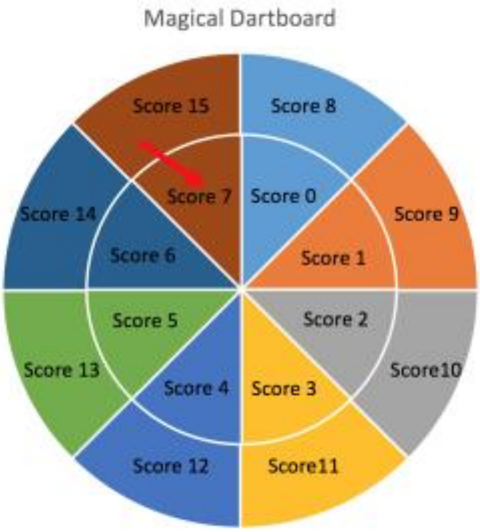
Let's throw the dart! There is a magical dartboard. Before throwing the darts, we set the score to each position of dartboard first. After each throwing, the inner ring of dartboard will rotate or the inner and outer rings exchange the scores. All magical actions of magical dartboard are dependent on the signals which are given when throwing the dart. Finally, you have to calculate the total score after throwing all darts. Following is an illustration:

Set the scores



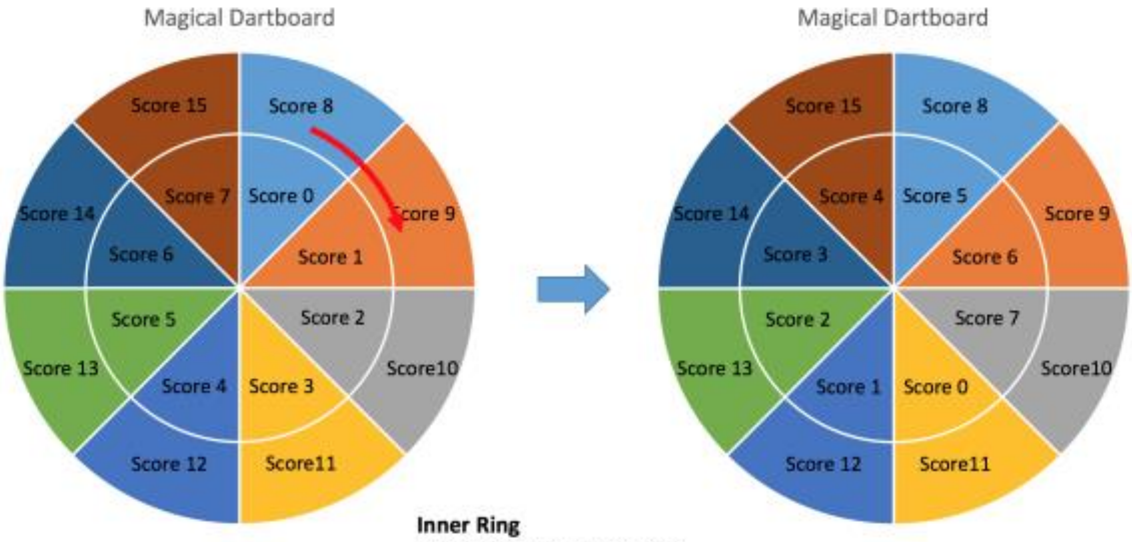
The signal of in_score will set the scores form position 0 to position 15

Throw the dart



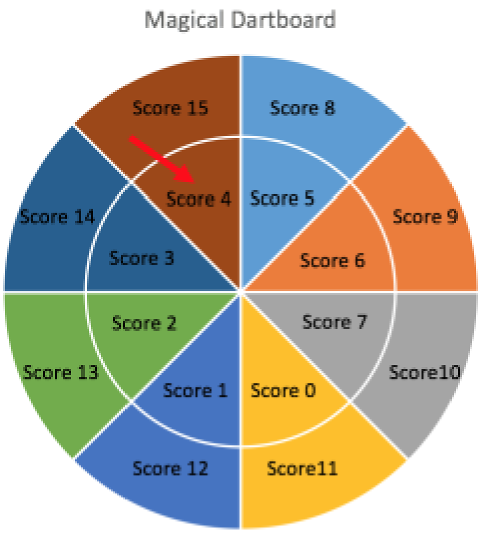
The hitting point of dart is position 7, getting the score 7 !
(the signal of in_dart shows the hitting point of the dart)

Dartboard rotate



Inner Ring
rotate_flag: 0 (clockwise)
In_rotation: 3

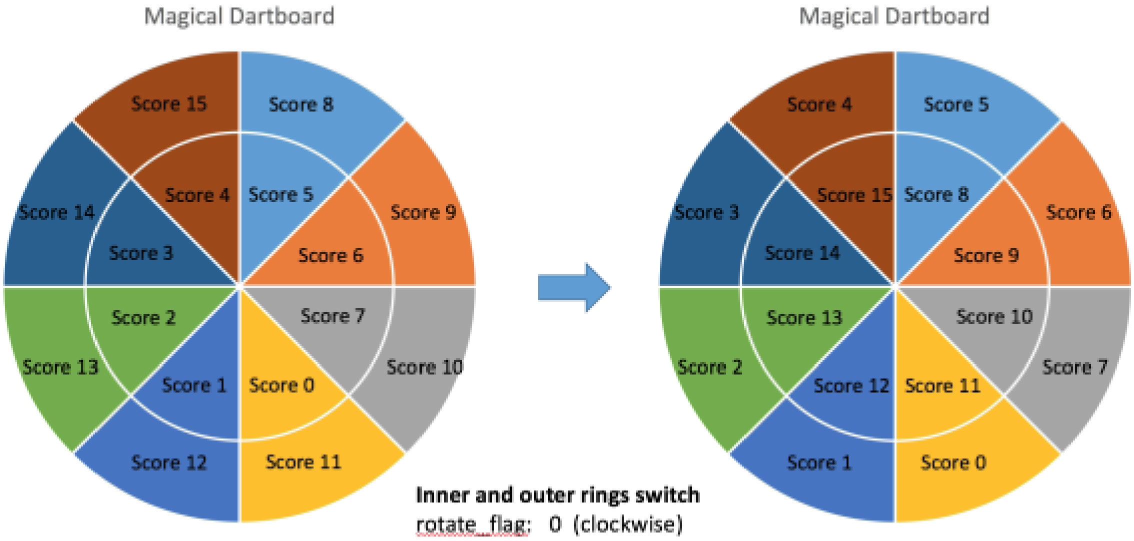
Throw the dart



The hitting point of dart is position 7, getting the score 4!

out_sum = score 7 + score 4

Dartboard rotate



Inner and outer rings switch

rotate_flag: 0 (clockwise)

In_rotation: 0

(If the value is 0, you don't care the signal rotate_flag)

And throw the next dart !

Inputs and Outputs

The following are the definitions of input signals

Input Signals	Bit Width	Definition
clk	1	Clock.
rst_n	1	Asynchronous active-low reset.
in_valid_1	1	High when in_score is valid.
in_valid_2	1	High when in_dart, in_rotation and rotate_flag are valid.
in_score	3	The score of the dartboard.
in_dart	4	The hitting point of the dart.
in_rotation	3	The rotation of the dartboard inner ring. (The inner and outer rings switch when the value equal 0)
rotate_flag	1	Specify the direction of the rotation. 0 for clockwise, 1 for counterclockwise.

The following are the definitions of output signals

Output Signals	Bit Width	Definition
out_valid	1	High when out_sum is valid.
out_sum	7	The total score after throwing all darts.

1. The input of **in_score** is delivered for 16 cycles continuously. When **in_valid_1** is low, input is tied to unknown state.
2. The inputs of **in_dart, in_rotation and rotate_flag** are delivered for 3~10 cycles continuously. When **in_valid_2** is low, inputs are tied to unknown state.
3. All input signals are synchronized at negative edge of the clock.
4. The output signal of **out_sum** must be delivered for 1 cycles, and **out_valid** should be high simultaneously.
5. The **in_valid_2** will come in 1~3 cycles after **in_valid_1** is pulled down
6. The next round of the game will come in 3~5 cycles after your **out_valid** is pulled down. (The scores of the dartboard will reset)
7. Sample waveform is shown in the last page of this document.
8. All operations are unsigned and the **out_sum** signal hasn't the overflow case

Specifications

1. Top module name: MD (design file name: MD.v)
2. It is **asynchronous** reset and **active-low** architecture.
3. The reset signal (**rst_n**) would be given only once at the beginning of simulation. All output signals should be reset after the reset signal is asserted.
4. The **out_sum** should be limited to zero when **out_valid** is low.
5. The **out_valid** is limited to high only **one** cycle when you want to output the result.
6. The execution latency is limited in 100 cycles. The latency is the clock cycles between the falling edge of the last **in_valid_2** and the rising edge of the first **out_valid**.
7. The output signal of **out_sum** must be correct when the signal of **out_valid** is high.
8. The clock period is **3** ns
9. The input delay is set to **0.5*(clock period)**.
10. The output delay is set to **0.5*(clock period)**, and the output loading is set to **0.05**.
11. The synthesis result of data type **cannot** include any latches.
12. The gate level simulation cannot include any timing violations without the notimingcheck command.
13. After synthesis, you can check MD.area and MD.timing. The area report is valid when the slack in the end of timing report should be **non-negative**.
14. The performance is determined by area and total execution latency. The lower, the better.

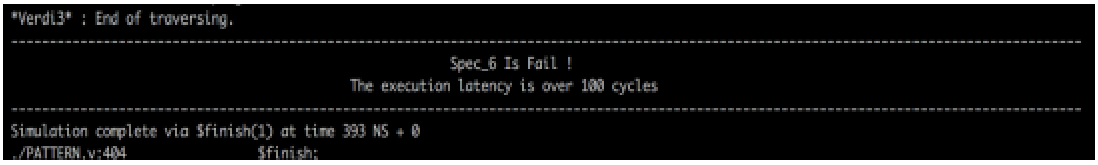
Grading Policy

1. Function Validity: 60%
2. Test Bench: 30%
   - Spec_3: 5%
   - Spec_4: 5% (only check the **out_sum** reset after **out_valid** is pulled down)
   - Spec_5: 5%
   - Spec_6: 5%
   - Spec_7: 5% + 5% (one correct design and one incorrect design)

✧ **Spec_3 means the third specification above**

✧ **You don't have second chance for test bench demo, it's served only one demo shoot.**

✧ **If any spec is violated, you have to show "Spec_X Is Fail !" on your screen.**
(X is the number of the spec)

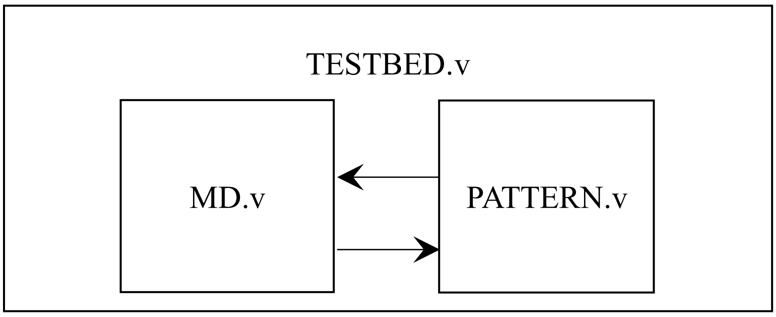


```
*Verdi3* : End of traversing.
------------------------------------------------------------------------------------------------------------
                                              Spec_6 Is Fail !
                                   The execution latency is over 100 cycles
------------------------------------------------------------------------------------------------------------
Simulation complete via $finish(1) at time 393 NS + 0
./PATTERN.v:404                    $finish;
```

3. Performance:10 %

- Area: 5%
- The total execution latency: 5%

Block diagram



Note

1. **Please upload the following files on e3 platform before 12:00 p.m. on Oct. 16:**
   - MD_iclab??.v and PATTERN_iclab??.v
2. **Template folders and reference commands:**

   01_RTL/ (RTL simulation) **./01_run**

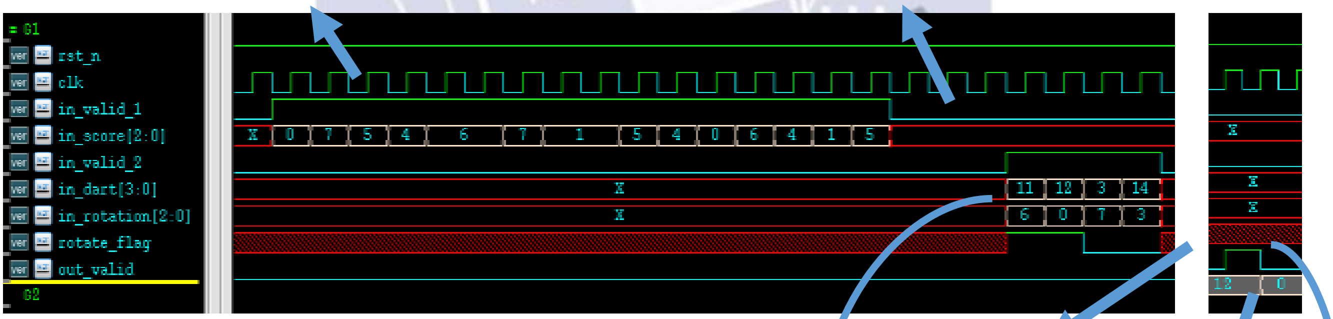
   02_SYN/ (Synthesis) **./01_run_dc**

   (Check the design if there's latch or not in ***syn.log***)

   (Check the design's timing in /Report/***MD.timing***)

   03_GATE / (Gate-level simulation) **./01_run**

Sample Waveform



16 cycles for in_valid_1

1~3 cycles between in_valid_1 and in_valid_2

Execution latency

out_valid is limited to one cycle and out_sum should be reset after pulling down out_valid

1. Hit the position 11

2. Inner ring of dartboard rotates 6 positions counterclockwise

3. Hit the position 12

4. Inner and outer rings switch the scores...

…. (in_dart, in_rotation and rotate_flag are delivered for 3~10 cycles)

next round will come in 3~5 cycles