**HOMEWORK 1 Approximate Average**

**Due date:**

## Overview

這次作業一的主要目的為複習Verilog語言和熟悉開發環境，請實作出近似平均數(Approximate Average)

## Homework Description

Please design a computational system whose transfer function is defined as follows. A series of 8-bit positive integer is generated as the input of the computational system by the test bench. The output value Y is a 10-bit positive integer, which is calculated according to equations (1), (2), (3) and (4).

|  |
| --- |
| …………………………………………………………………………………….....(1)  ,where Xi is the value of the ith input data and j>=1.  , n = 9 in this homework |
| ………………………………………………………………………...(2) |
| ……...(3)  ,where is the value of the jth approximate average |
| …………………………………………………………………………........…..(4) |

,where is the value of the jth output data.

The computational system produces the output sequence according to the given input sequence. Each input and output data in the respective sequence is indexed. This index, in terms of hardware, is the relative time when the input data is given or the output data is ready. Thinking as a hardware designer, the approximate average is chosen from the last *n* input data which should be stored in the system. The system should be able to calculate the integral part of the real average of the last *n* input data first. And then if the integral part of the real average equals to any one of the last *n* input data, the approximate average is simply the integral part. Else the approximate average is the one which is one of the last *n* input data whose value is smaller than and closest to the integral part of the real average. The above descriptions stated the desired operations as those defined by equations (1), (2), and (3).

After the approximate average is obtained, the output value can be calculated according to equation (4). First, the last *n* input value is added by the corresponding approximate average. And then they are summed up and divided by *n-1*. The output value is the quotient after division.

For example, assume that *n*=4, *X1*=3, *X2*=24, *X3*=16, *X4*=8, and *X5*=3. After the first 5 input items are given, the system should store them and calculate the output value. The average of the first 4 input values is 12(only the integral part is left). Since it is not in the set of {*X1*, *X2*, *X3*, *X4*}, the system selects one from {*X1*, *X2*, *X3*, *X4*} as the approximate average whose value is smaller than 12 and close to 12. In this case, the approximate average is 8. So the first output value is calculated n as

⌊[(3 + 8) + (24 + 8) + (16 + 8) + (8 + 8)] / [(4 - 1)]⌋ = 27 .

Similar to those described above, when the 5th input data item is given, the system should store *X2*, *X3*, *X*4 and *X5* and calculate the corresponding output value. The 2nd output value should be the same as the first one because the values stored in the system is the same.

## Design Specifications

1. Block Overview

Diagram, schematic

Description automatically generated

1. I/O Interface

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | I/O | width | Description |
| clk | I | 1 | clock for the computational system |
| reset | I | 1 | reset the state of the computational system when it asserts |
| X | I | 8 | input data of the computational system |
| Y | O | 10 | computed output |

1. Timing Diagrams



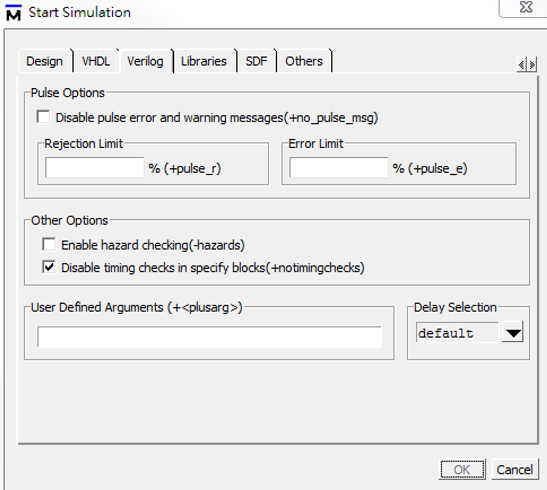
1. File Description

|  |  |
| --- | --- |
| File Name | Description |
| CS.v | RTL code for using Verilog |
| testfixture.v | Test bench for verifying design |
| in.dat | Input patterns |
| out\_golden.dat | Golden output patterns |

## Scoring

1. Functional Simulation (pre-sim) **[100%]**

All of the result should be generated correctly, and you will get the following message in ModleSim simulation. You can turn off the timing check in presim only??.



**Appendix**

(Hexadecimal number)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Index i | Xi |  |  |  | Yj | Index j |
| 1 | 8f |  |  |  |  |  |
| 2 | 0b |  |  |  |  |  |
| 3 | 5d |  |  |  |  |  |
| 4 | 20 |  |  |  |  |  |
| 5 | f3 |  |  |  |  |  |
| 6 | 3e |  |  |  |  |  |
| 7 | e5 |  |  |  |  |  |
| 8 | 03 |  |  |  |  |  |
| 9 | 0c | 5c | 3e | 056a | 00ad | 1 |
| 10 | 74 | 59 | 3e | 054f | 00a9 | 2 |
| 11 | 79 | 65 | 5d | 06d4 | 00da | 3 |
| 12 | 01 | 5b | 3e | 0561 | 00ac | 4 |
| 13 | 30 | 5c | 3e | 0571 | 00ae | 5 |
| 14 | 2e | 46 | 3e | 04ac | 0095 | 6 |
| 15 | a4 | 52 | 30 | 0494 | 0092 | 7 |
| 16 | 76 | 45 | 30 | 0425 | 0084 | 8 |
| 17 | 84 | 54 | 30 | 04a6 | 0094 | 9 |
| 18 | 51 | 5b | 51 | 0614 | 00c2 | 10 |
| 19 | d6 | 66 | 51 | 0676 | 00ce | 11 |
| 20 | 70 | 65 | 51 | 066d | 00cd | 12 |
| 21 | 35 | 6b | 51 | 06a1 | 00d4 | 13 |
| 22 | 10 | 68 | 51 | 0681 | 00d0 | 14 |
| 23 | 23 | 66 | 51 | 0676 | 00ce | 15 |
| 24 | e7 | 6e | 51 | 06b9 | 00d7 | 16 |
| 25 | 3b | 67 | 51 | 067e | 00cf | 17 |
| 26 | 6d | 65 | 51 | 0667 | 00cc | 18 |
| 27 | 34 | 61 | 3b | 0584 | 00b0 | 19 |
| 28 | 61 | 54 | 3b | 050f | 00a1 | 20 |
| 29 | 89 | 57 | 3b | 0528 | 00a5 | 21 |
| 30 | bf | 67 | 61 | 0708 | 00e1 | 22 |
| 31 | dc | 74 | 6d | 0840 | 0108 | 23 |
| 32 | d3 | 91 | 89 | 09ec | 013d | 24 |
| 33 | 9c | 88 | 6d | 08a5 | 0114 | 25 |
| 34 | 8f | 92 | 8f | 0a2b | 0145 | 26 |

(Decimal number)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Index i | Xi |  |  |  | Yj | Index j |
| 1 | 143 |  |  |  |  |  |
| 2 | 11 |  |  |  |  |  |
| 3 | 93 |  |  |  |  |  |
| 4 | 32 |  |  |  |  |  |
| 5 | 243 |  |  |  |  |  |
| 6 | 62 |  |  |  |  |  |
| 7 | 229 |  |  |  |  |  |
| 8 | 3 |  |  |  |  |  |
| 9 | 12 | 92 | 62 | 1386 | 173 | 1 |
| 10 | 116 | 89 | 62 | 1359 | 169 | 2 |
| 11 | 121 | 101 | 93 | 1748 | 218 | 3 |
| 12 | 1 | 91 | 62 | 1377 | 172 | 4 |
| 13 | 48 | 92 | 62 | 1393 | 174 | 5 |
| 14 | 46 | 70 | 62 | 1196 | 149 | 6 |
| 15 | 167 | 82 | 48 | 1172 | 146 | 7 |
| 16 | 118 | 69 | 48 | 1061 | 132 | 8 |
| 17 | 132 | 84 | 48 | 1190 | 148 | 9 |
| 18 | 81 | 91 | 81 | 1556 | 194 | 10 |
| 19 | 214 | 102 | 81 | 1654 | 206 | 11 |
| 20 | 112 | 101 | 81 | 1645 | 205 | 12 |
| 21 | 53 | 107 | 81 | 1697 | 212 | 13 |
| 22 | 16 | 104 | 81 | 1665 | 208 | 14 |
| 23 | 35 | 102 | 81 | 1654 | 206 | 15 |
| 24 | 231 | 110 | 81 | 1721 | 215 | 16 |
| 25 | 59 | 103 | 81 | 1662 | 207 | 17 |
| 26 | 109 | 101 | 81 | 1639 | 204 | 18 |
| 27 | 52 | 97 | 59 | 1412 | 176 | 19 |
| 28 | 97 | 84 | 59 | 1295 | 161 | 20 |
| 29 | 137 | 87 | 59 | 1320 | 165 | 21 |
| 30 | 191 | 103 | 97 | 1800 | 225 | 22 |
| 31 | 220 | 125 | 109 | 2112 | 264 | 23 |
| 32 | 211 | 145 | 137 | 2540 | 317 | 24 |
| 33 | 156 | 136 | 109 | 2213 | 276 | 25 |
| 34 | 143 | 146 | 143 | 2603 | 325 | 26 |

## General rules for deliverables

* You need to complete this homework INDIVIDUALLY. You can discuss the homework with other students, but you need to do the homework by yourself. You should not copy anything from someone else, and you should not distribute your homework to someone else. If you violate any of these rules, you will get NEGATIVE scores, or even fail this course directly
* When submitting your homework, compress all files into a single **zip** file, and upload the compressed file to Moodle.
  + Please follow the file hierarchy shown in Figure 1.

**F740XXXXX ( your id ) (folder)**

**src( folder ) \* Store your source code**

**report.docx ( project report. The report template is already included. Follow the template to complete the report. )**

|  |
| --- |
|  |
| Figure 1. File hierarchy for homework submission |

* **Important!** DO NOT submit your homework in the last minute. Late submission is not accepted.
* You should finish all the requirements (shown below) in this homework and Project report.
* If your code can not be recompiled by TA successfully using modelsim, you will receive NO credit.
* Verilog and SystemVerilog generators aren’t allowed in this course.