Worcester Polytechnic Institute

ECE 5723: Methodologies for System Level Design and Modeling

RT Level Design and C++ Logic Modeling

by

Vladimir Vakhter

Laboratory Report

09/29/2020

Table of Contents

1. Introduction

In this work, a circuit producing the average of the 8 serial byte-data on its input (Fig. 1) is designed using C++:

- The circuit starts its operation with a complete pulse on *start* (0-1-0).
- After that, the next eight bytes on *data* will be received and the average of the eight will be calculated.
- When this is completed, the *avg* output will contain the integer part of the calculated average, and the *rdy* signal becomes 1. This signal remains active until the next time that *start* becomes 1.

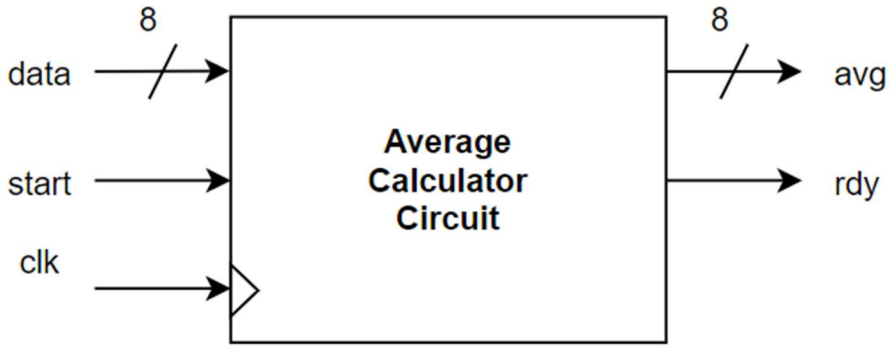

Fig. 1 – Diagram of Average Calculator Circuit

2.Analysis of the Task

Based on the information given in the introduction, the graph shown in **Fig. *2*** for the waveforms was derived

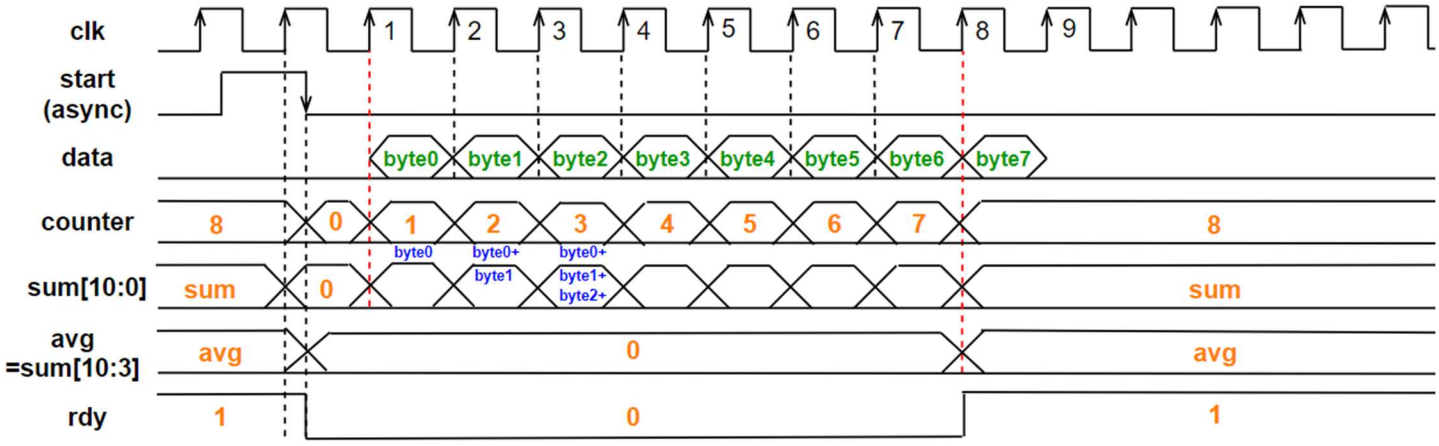

Fig. 2 – The waveforms demonstrate the operation of the average calculator

3. Schematic Diagram of the Datapath

In this chapter, the schematic diagram (according to the Huffman model) of the data path of the designed circuit is presented in **Fig. *3***.

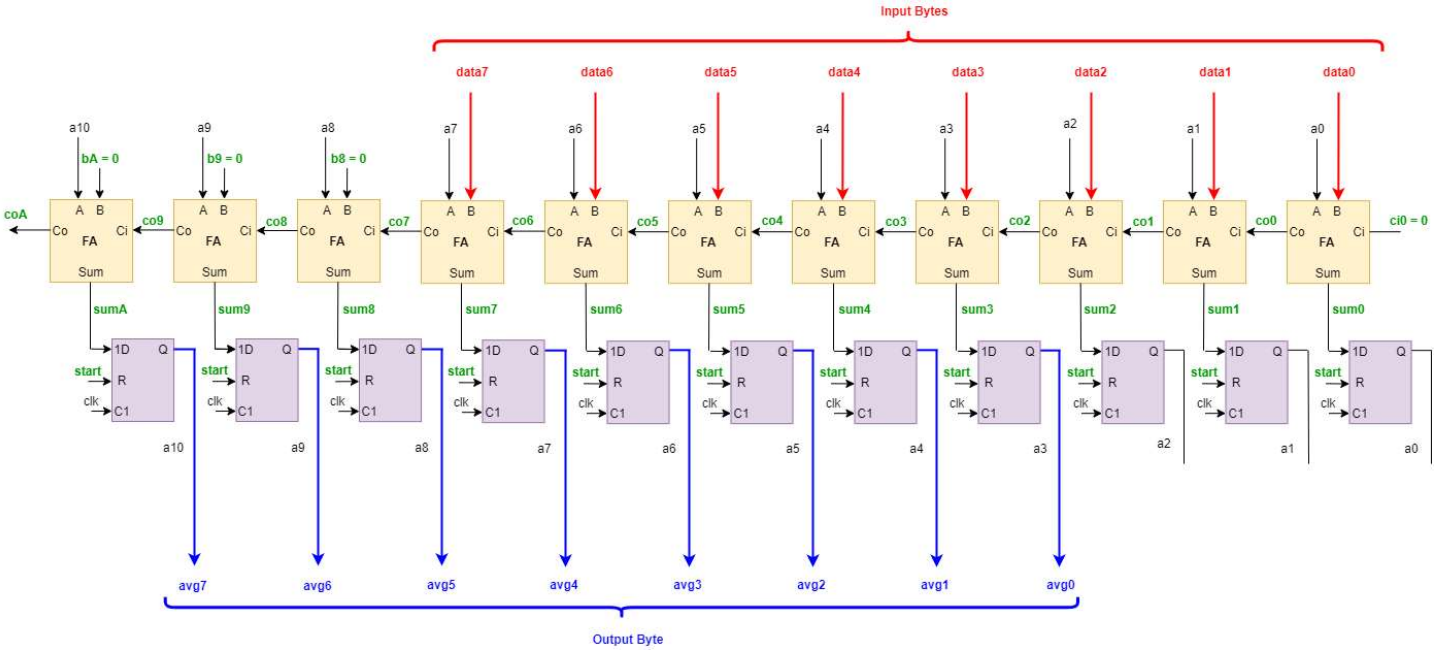

Fig. 3 – The schematic diagram of average calculator

Here, the division by 8 is implemented by dropping three least significant bits of the 11-bit output register.

4. Controller State Diagram

In this chapter, the controller state diagram of the designed circuit is shown in **Fig. 4**.

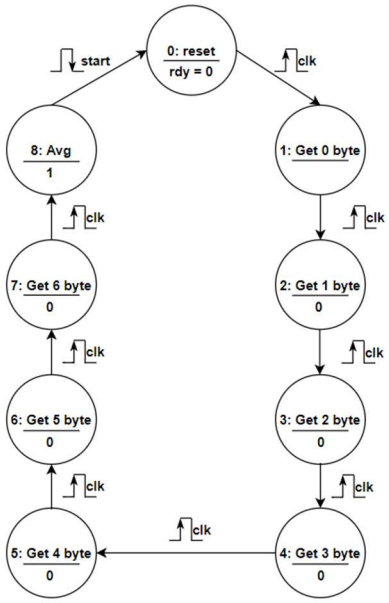

Fig. 4 *– The state diagram of the average calculator circuit*

5. Schematic Development in C++

For implementing designs, first, a class for the average calculator circuit was created. This class contains full-adders and D flip-flops connected with the use of internal wires. The evl() function of this class, ensure a proper order of the evl() functions of the contained components. The complete design was instantiated inside the main() function.

6. Testbench

In this chapter, the results of the verification of designed circuit are provided for four different scenarios.

6.1. Scenario 1

In this test scenario, all the 8 bytes of input data are 0x00.

```
Test case 1 begins...
=============================================
Expected avg = 00000000

Negedge of start was received...
avg[7:0]: 00000000 @ 0
rdy: 0 @ 0

Results:
avg[7:0]: 00000000 @ 2
rdy: 1 @ 2
Test case 1 ends...
```

6.2. Scenario 2

In this test scenario, all the 8 bytes of input data are 0xFF.

```
Test case 2 begins...
=============================================
Expected avg = 11111111

Negedge of start was received...
avg[7:0]: 00000000 @ 2
rdy: 0 @ 2

Results:
avg[7:0]: 11111111 @ 50
rdy: 1 @ 50
Test case 2 ends...
```

6.3. Scenario 3

In this test scenario, the input data bytes are 00000001, 00000011, 00000111, …, 11111111.

```
Test case 3 begins...
=============================================
Expected avg = 00111110

Negedge of start was received...
avg[7:0]: 00000000 @ 50
rdy: 0 @ 50

Results:
avg[7:0]: 00111110 @ 61
rdy: 1 @ 61
Test case 3 ends...
```

6.4. Scenario 4

In this test scenario, the input data bytes are 10000000, 11000000, 11100000, …, 11111111.

```
Test case 4 begins...
=============================================
Expected avg = 11100000

Negedge of start was received...
avg[7:0]: 00000000 @ 61
rdy: 0 @ 61

Results:
avg[7:0]: 11100000 @ 110
rdy: 1 @ 110
Test case 4 ends...
```

7. Conclusions

In this work, a circuit producing the average of the 8 serial bytes on its input was designed and tested in C++.