2022 Digital IC Design

Midterm Exam

Meeting Room Number：9

Please check if the last digit of your student id is equal to the meeting room number. Each meeting room has different exam questions. TAs will verify your code according to your student id, the mismatch between student id and meeting room number may cause you to lose part of the score.

Question 1：(30%)

In Question 1, you are requested to design an 8-bit Ripple Carry Adder (RCA). The RCA circuit can be constructed with a cascade architecture of 1-bit full-adders. As a result, an 8-bit RCA comprises eight 1-bit full-adders, which is shown in Fig. 1. A full-adder can be constructed with two half-adders and one OR gate. Assume that x and y are 1-bit input signals and s and c are outputs standing for sum and carry. The computation of the half adder can be represented as equation (1). Fig. 2 illustrates the architecture of a half adder and a full adder.

The module of the half adder and the full adder are already provided in the files HA.v and FA.v, respectively. The hierarchy architecture of the RCA is also provided in the file RCA.v, with parts of it missing. Please fill in the missing parts and complete the module of RCA.

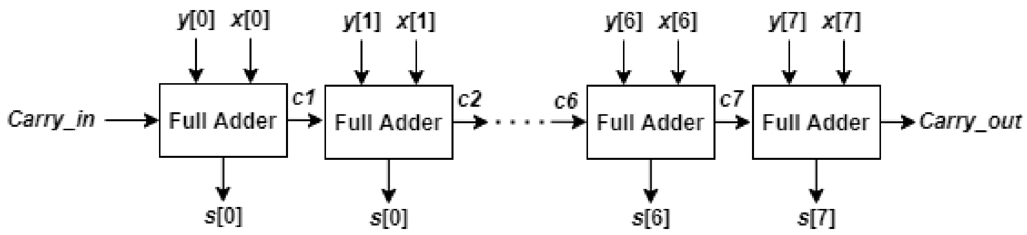
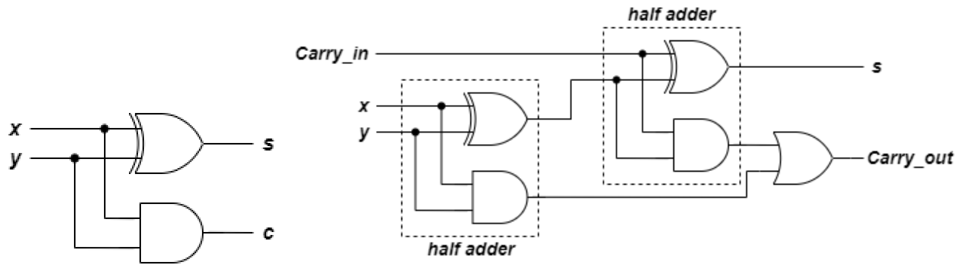


Fig. 1. The architecture of an 8-bit ripple carry adder.



(a) (b)

Fig. 2. The architecture of (a) a half-adder and (b) a full adder.

$$s = x \oplus y \qquad (1)$$

$$c = x \cdot y$$

1. Design Specifications:

1.1 Block Overview

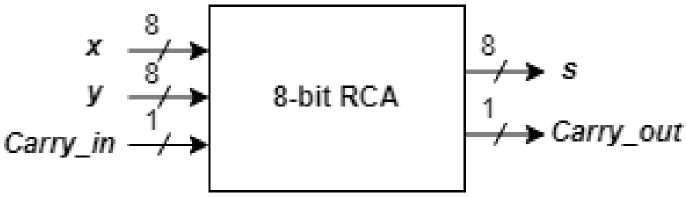


Fig. 3. The block overview.

1.2 I/O Interface

Signal Name	I/O	width	Description
x	I	8	augend
y	I	8	summand
Carry_in	I	1	carry_in
s	O	8	sum
Carry_out	O	1	carry_out

1.3 File Description

File Name	Description
HA.v	The module of half-adder.
FA.v	The module of full-adder.
RCA.v	The module of ripple carry adder, which is the top module in this design.
RCA_tb.v	The testbench file. The content in this file is not allowed to be modified.

Question 2：(70%)

In Question 2, you are requested to design a Basic Operation Engine (BOE). The BOE module will take a series of data as input, and finding the minimum, summation and decreasing sequence of them. The specification and function of BOE circuit will be described in detail in the following section.

1. Design Specifications:

1.1 Block Overview

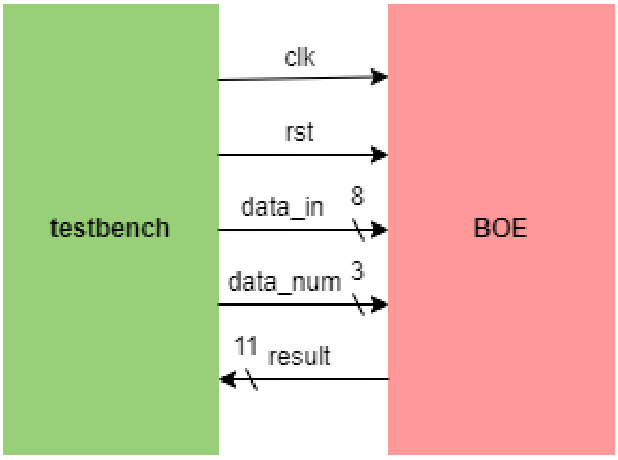


Fig. 4. The block overview.

1.2 I/O Interface

Signal Name	I/O	width	Description
clk	I	1	This circuit is a synchronous design triggered at the positive edge of clk.
rst	I	1	Active-high asynchronous reset signal.
data_in	I	8	A series of 8-bit data input port.
data_num	I	3	The number of the input data in a series. The range of *data_num* is from 2 to 6.
result	O	11	The result of basic operations of a series of data.

1.3 File Description

File Name	Description
BOE.v	The module of basic operation engine, which is the top module in this design.
BOE_tb.sv	The testbench file. The content in this file is **not allowed** to be modified.

2. Timing Specifications

After the system reset (T1), the number of the input data in a series will be input by *data_num* port. Meanwhile, the data in a series will be input by *data_in* port within *data_num* cycles. At T2, all the data is input. Then you must output the operation result from *result* port. At T3, the minimum of the series should be output. The sum of the series should be output at T4. From T5, the sequence of the data will be output from big to small for *data_num* cycles. At T6, a new series of data will be input, and the BOE will start a new round of operations.

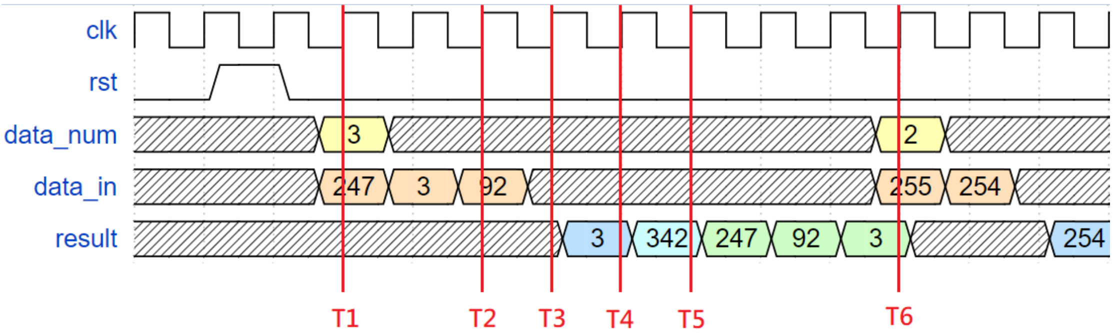


Fig. 5. Timing diagram of data input and result output.

Fig. 6 shows the finite state machine of the BOE. You can complete the BOE module according to Fig. 6, or design your own state machine.

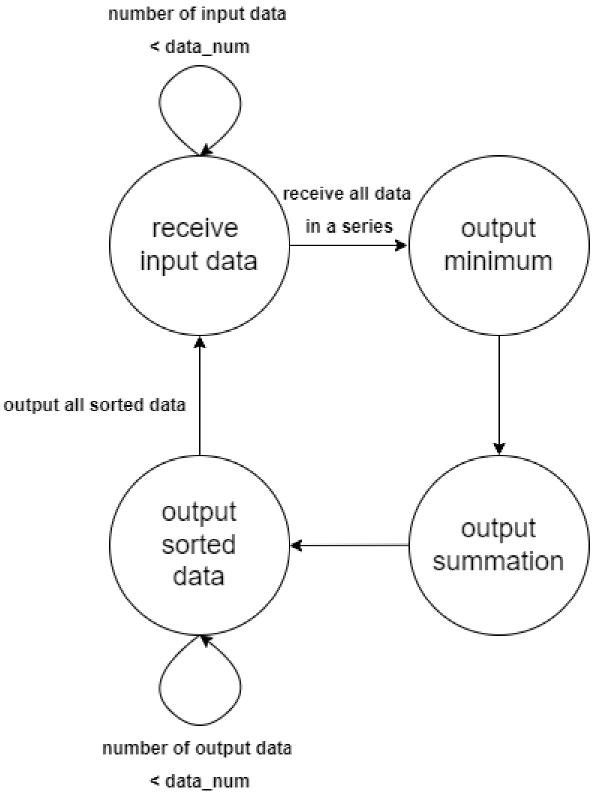


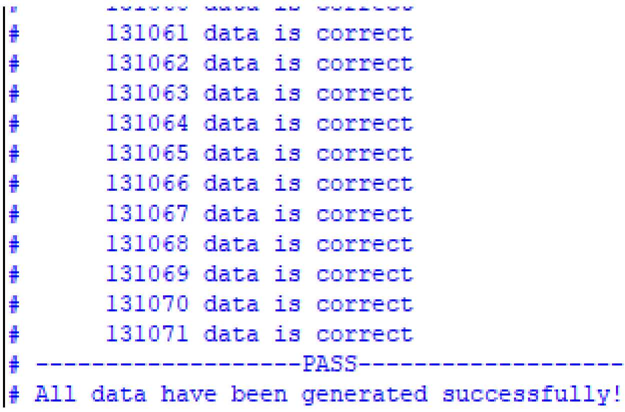
Fig. 6. Finite state machine of the BOE.

Scoring：

In this exam, you only need to pass the functional simulation. The scoring of each part is as follows:

1. Question 1：[30%]

All of the result should be generated correctly, and you will get the following message in ModelSim simulation.



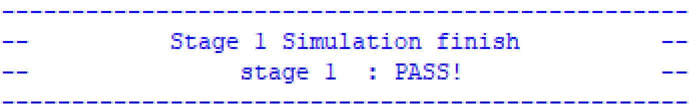
```
#       131060 data is correct
#       131061 data is correct
#       131062 data is correct
#       131063 data is correct
#       131064 data is correct
#       131065 data is correct
#       131066 data is correct
#       131067 data is correct
#       131068 data is correct
#       131069 data is correct
#       131070 data is correct
#       131071 data is correct
# -------------------PASS-------------------
# All data have been generated successfully!
```

Fig. 7. Simulation result of Question 1.

2. Question 2：[70%]

2.1. Minimum function testing **[20%]**

In stage 1, testbench will verify the minimum function of your design. If the minimum of each series of data are all generated correctly, you will get the following message in ModelSim simulation.

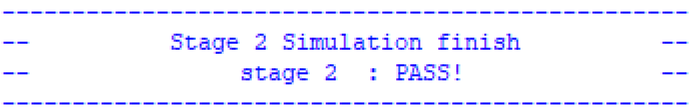


```
--------------------------------------------------
--          Stage 1 Simulation finish           --
--               stage 1  : PASS!               --
--------------------------------------------------
```

Fig. 8. Simulation result of Question 2 in stage 1.

2.2. Summation function testing **[20%]**

In stage 2, testbench will verify the summation function of your design. If the summation of each series of data are all generated correctly, you will get the following message in ModelSim simulation.

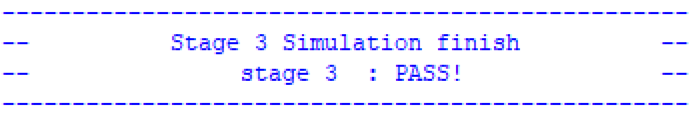


```
--------------------------------------------------
--          Stage 2 Simulation finish           --
--               stage 2  : PASS!               --
--------------------------------------------------
```

Fig. 9. Simulation result of Question 2 in stage 2.

2.3. Sorting function testing **[30%]**

In stage 3, testbench will verify the sorting function of your design. If the sorted result of each series of data are all generated correctly, you will get the following message in ModelSim simulation.



```
--------------------------------------------------
--          Stage 3 Simulation finish           --
--               stage 3  : PASS!               --
--------------------------------------------------
```

Fig. 10. Simulation result of Question 2 in stage 3.

Submission:

1. Submitted files

You should classify your files into two directories and compress them to .zip format. The naming rule is mid_studentID_name.zip. If your file is not named according to the naming rule, you will lose five points. Please submit the compressed file to folder Mid in moodle.

mid_studentID_name.zip	
Q1	
*.v	All of your Verilog RTL code for Q1
Q2	
*.v	All of your Verilog RTL code for Q2

2. Note

Please do not modify any content of testfixture.

Deadline: 2022/4/12 12:00.

No late submissions will be accepted, please pay attention to the deadline.