**Assignment 2\_Report**

**Introduction**

Assignment 2 is continued to complete and improve the content of Assignment 1. This Assignment 2 is also to create a digital system on the DE2 board using Quartus II. Previously, in Assignment 1, the counter was used to count from 0 to 73, and then the four numbers needed were taken out. But now it is required to replace the counter with a random number generator, so that the system can randomly pick out four numbers between 0 and 73, and the numbers can be displayed on the DE2 board for 4 seconds.

**Table of Contents**

[**Introduction**](#_Toc512788800)

[Table of Contents](#_Toc512788801)

[**1** **Block Diagrams 1**](#_Toc512788802)

[**2** **ASM Charts 2**](#_Toc512788802)

2.1 Key module 2

2.2 Time module 3

2.3 Controller module 4

2.4 Random module 5

2.5 Number module 6

2.6 Bcd module 7

2.7 7-segment module 8

[**3** **Full simulations for each module 9**](#_Toc512788802)

3.1 Key module 9

3.2 Time module 10

3.3 Random module 11

3.4 Bcd module 12

3.5 7-segment module 13

[**4** **RTL Schematic 1**](#_Toc512788802)**3**

[**5** **Simulation of the full system 1**](#_Toc512788802)**4**

[**6** **Results 15**](#_Toc512788802)

[**7** **Conclusion 15**](#_Toc512788810)

[**8** **Coding 16**](#_Toc512788803)

8.1 Controller module 16

8.2 Key module 17

8.3 Time module 17

8.4 Random module 18

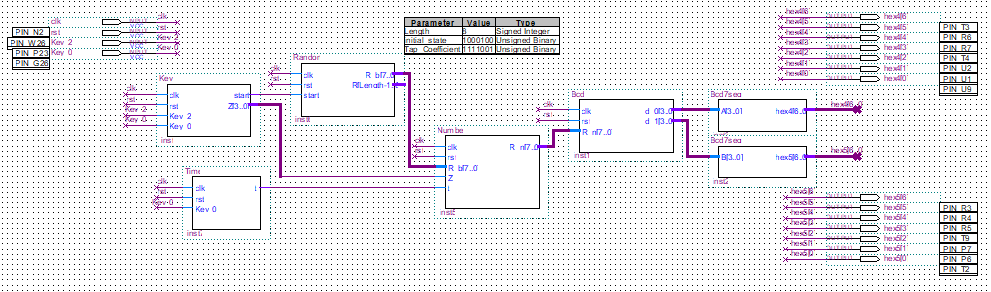
8.5 Number module 19

8.6 Bcd module 20

8.7 7-segment\_0 module 21

8.8 7-segment\_1 module 21

**1. Block Diagrams**

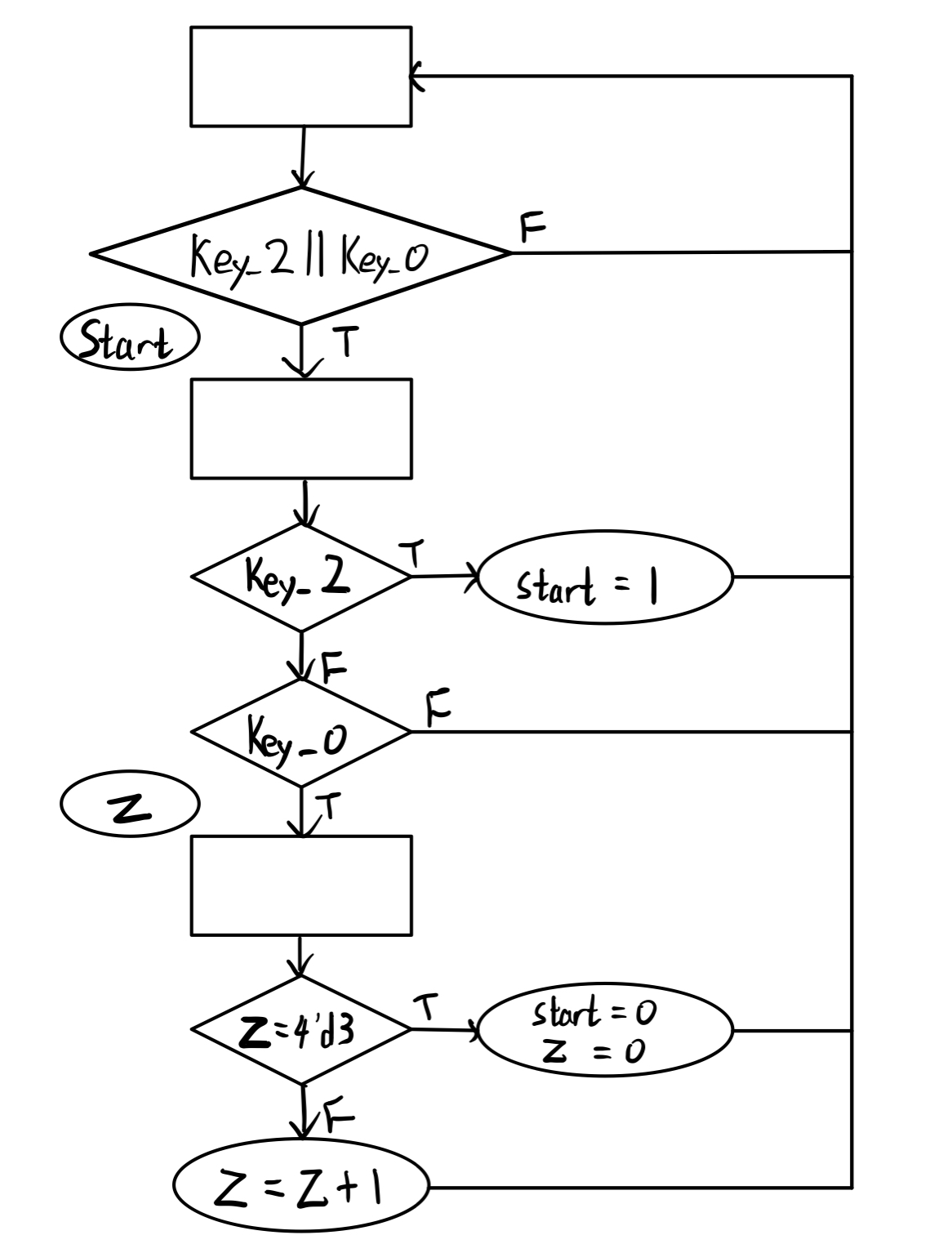
Figure 1 is a block diagram of the entire system, including Key module, Controller module, Random module, Time module, Number module, Bcd module and two 7-segment modules. The Key module is used to record the number of times Key\_0 is pressed, and tell the system to start operation when Key\_2 is pressed. The controller module is used to link each module and will not appear in Figure 1. The random module is used to generate random numbers between 0 and 73. The time module is used to record the time when the number is displayed on the DE2 board, the setting is 4 seconds. The number system is used to avoid displaying the same number. The bcd module converts the binary code generated by the random module into Bcd code. The 7-segment module converts the bcd code into the corresponding light of each DE2 board and displays the number. The 7-segment module output to HEX4 displays single digits, while the 7-segment module output to HEX5 displays tens digits. On the system, the clock is set to 50M Hz, and the button corresponding to rst is Key\_3.

**Figure 1:** Block Diagrams of Architecture

**2. ASM Charts**

2.1 Key module

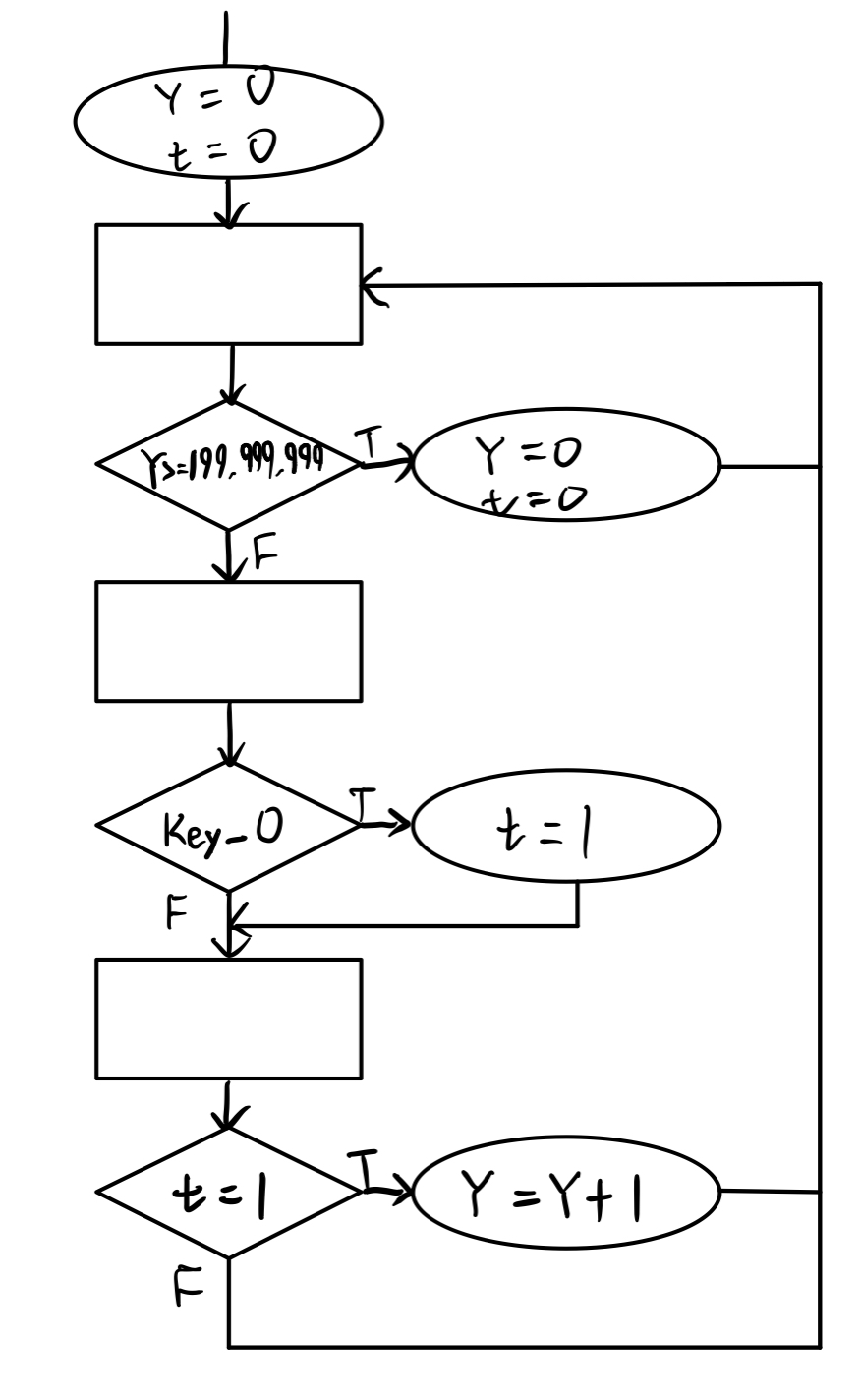
Figure 2 is the ASM chart of the key module. When the Key is pressed, it will be judged whether it is Key\_0 or Key\_2. When Key\_2 is pressed, the start signal will change to ‘1’, and the system will start to operate. When you press Key\_0, it will start counting with Z. In order to make the system only show four numbers and after the fourth number appears, the start signal will change to ‘0’ to stop the system.



**Figure 2:** The ASM chart of Key module

2.2 Time module

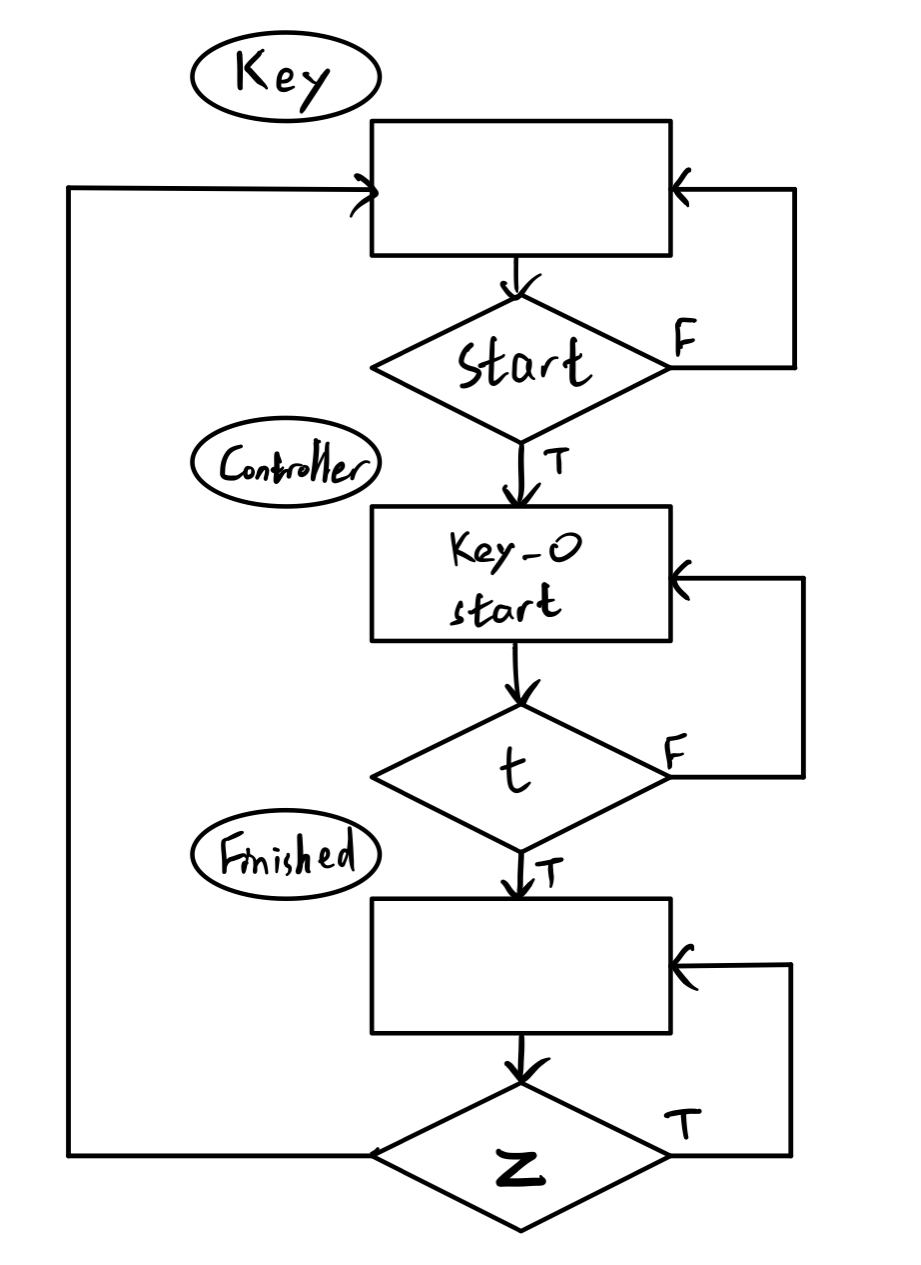
Figure 3 is the ASM chart of the control module. When Key0 is pressed, the t signal becomes ‘1’ and Y starts to count. 50Mhz is a cycle of 0.00000002 seconds, so Y counts 200 million times is 4 seconds. The t signal will become ‘0’ after 200 million times.



**Figure 3:** The ASM chart of Time module

2.3 Controller module

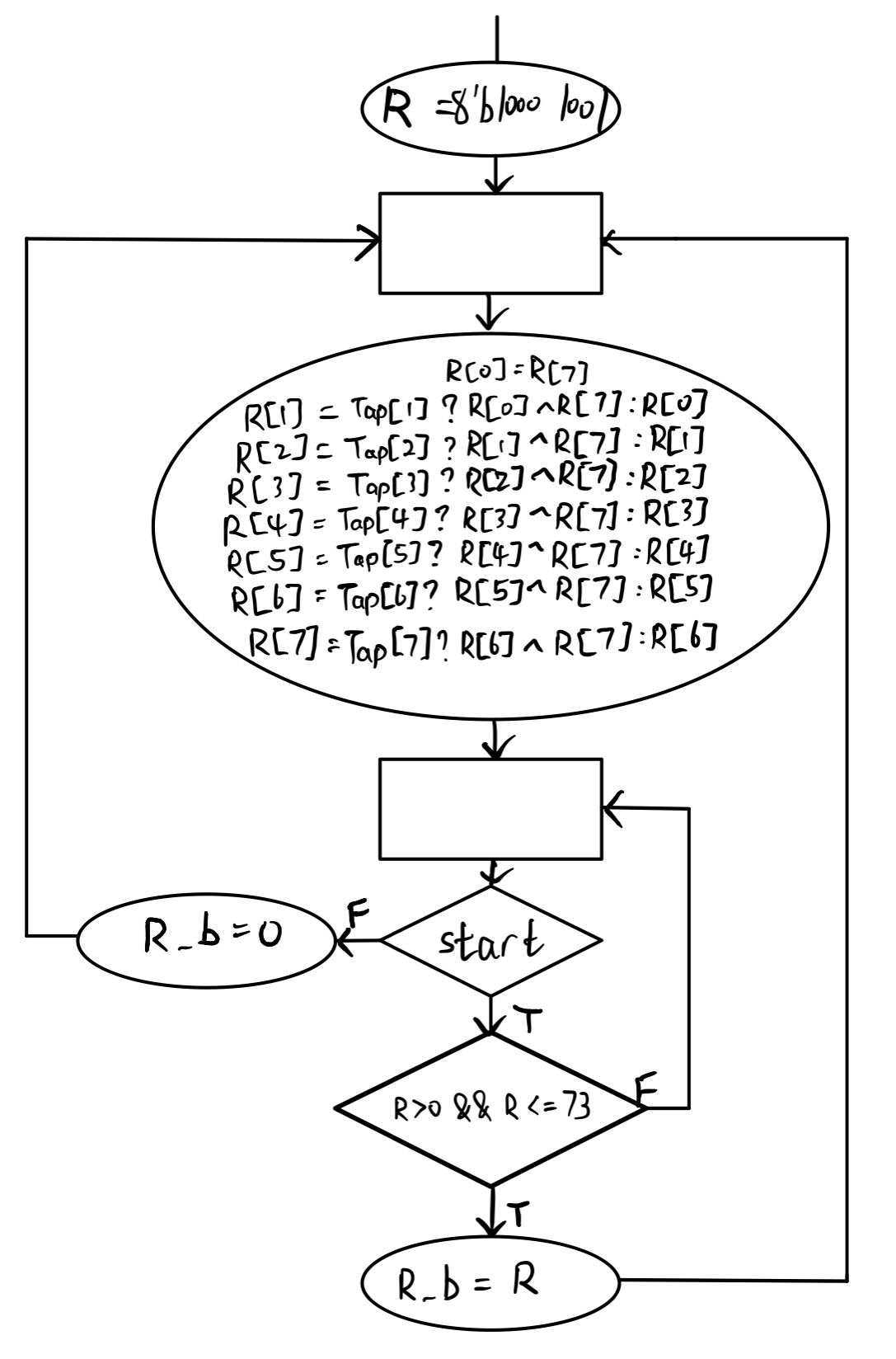
Figure 4 is the ASM chart of the controller module, which can use the output of one module as the input of another module.



**Figure 4:** The ASM chart of Controller module

2.4 Random module

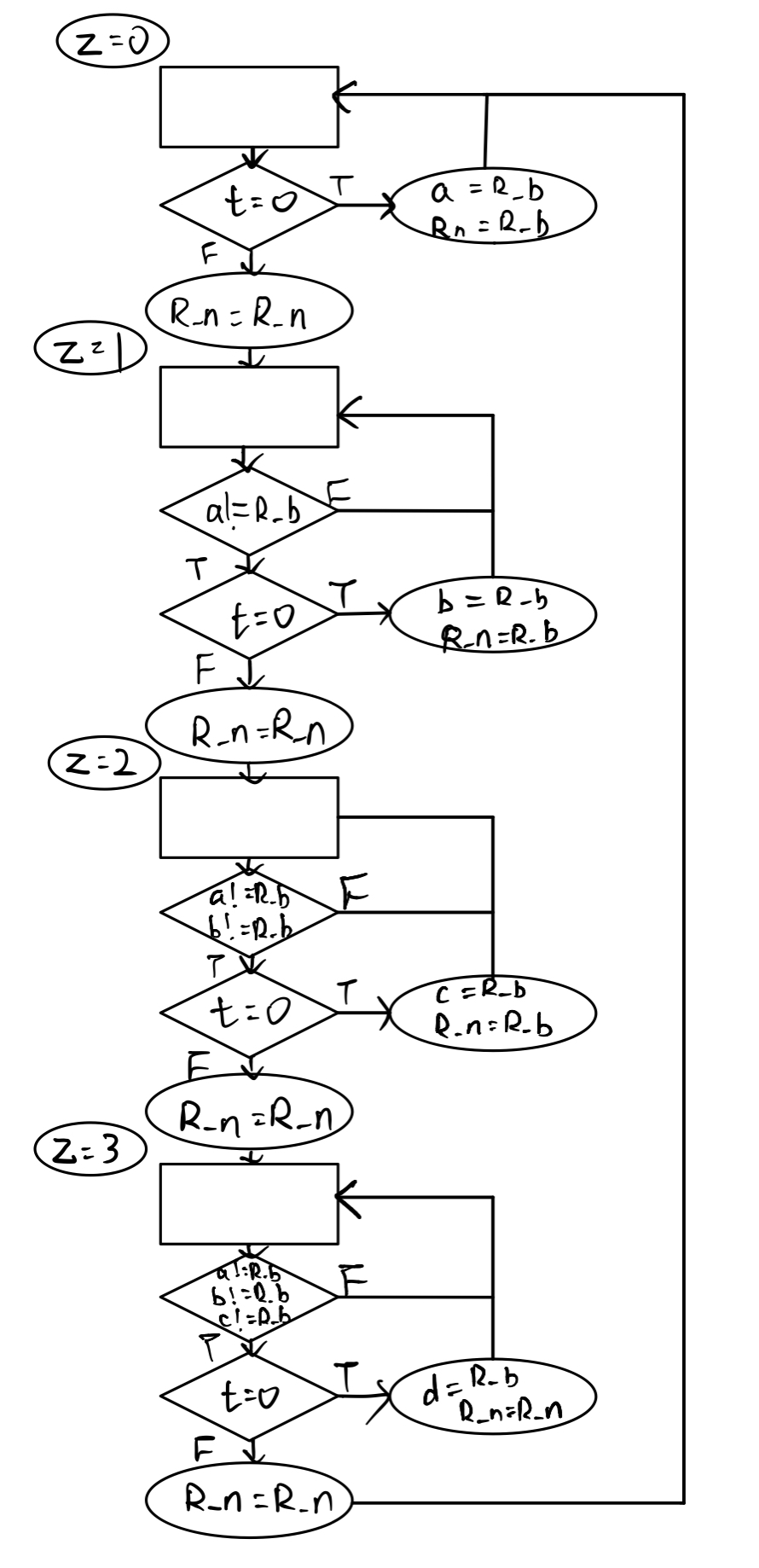
Figure 5 is the ASM chart of the random module. The random number is generated by the linear feedback shift register when the start signal is ‘1’, and a number between -128 and 128 is randomly generated. Then, the system extracts numbers from 0 to 73 from these numbers.



**Figure 5:** The ASM chart of Random module

2.5 Number module

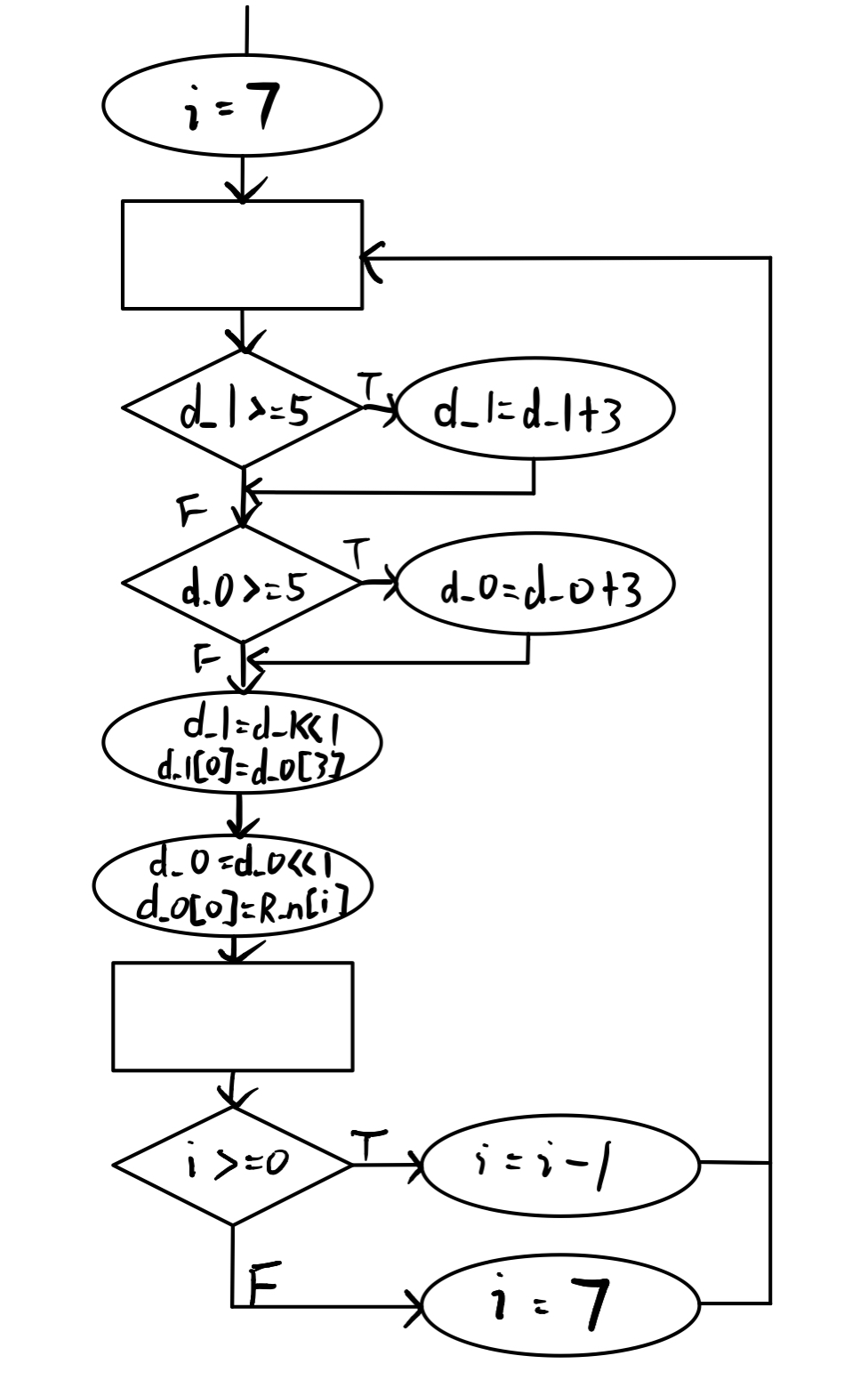
Figure 6 is the ASM chart of the number module. It compares all the previous numbers with the following numbers to select four different numbers to avoid duplicate numbers.



**Figure 6:** The ASM chart of Number module

2.6 Bcd module

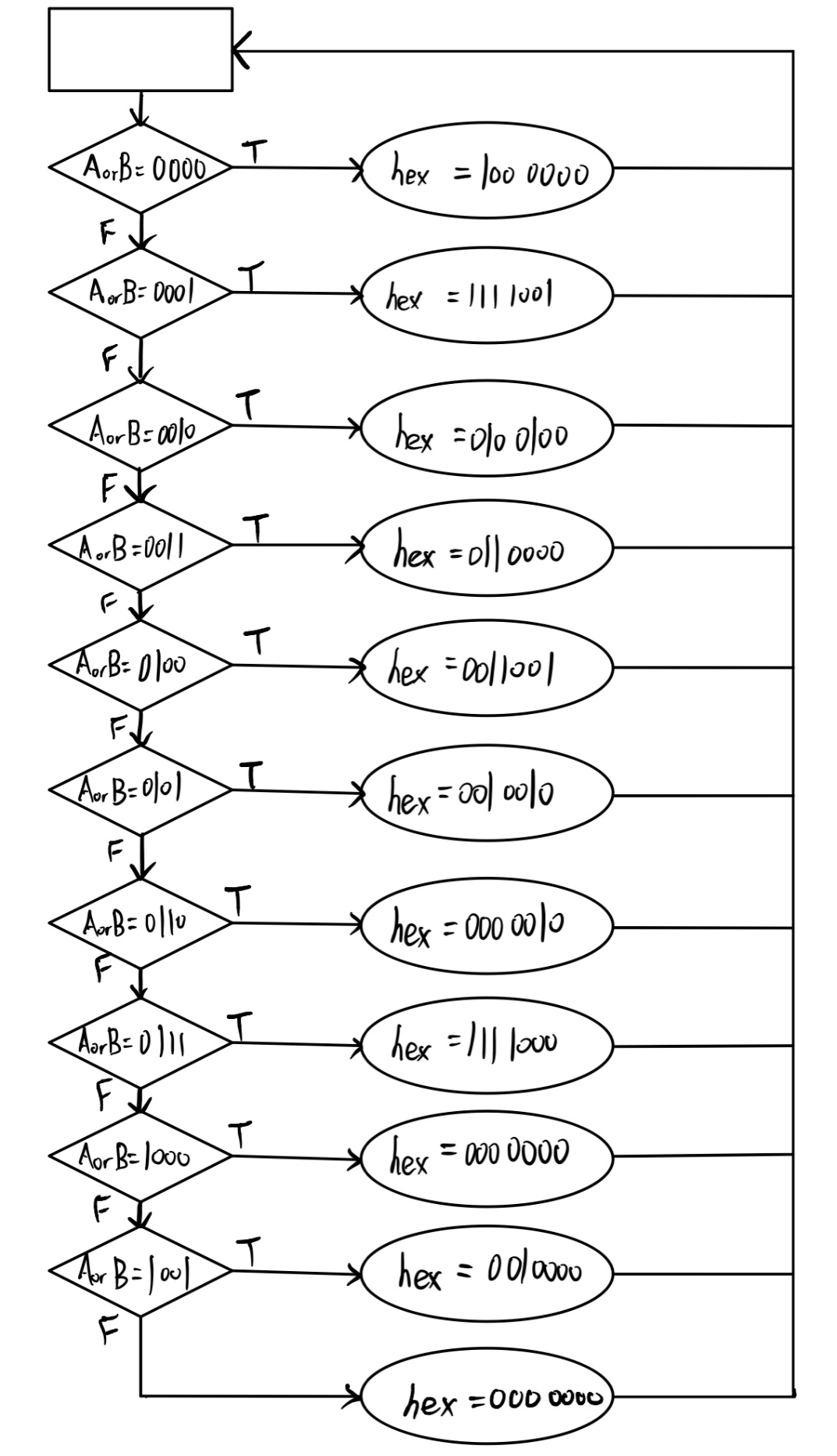
Figure 7 is the ASM chart of the Bcd module, which converts the finally selected binary number into Bcd encoding format.



**Figure 7:** The ASM chart of Bcd module

2.7 7-segment module

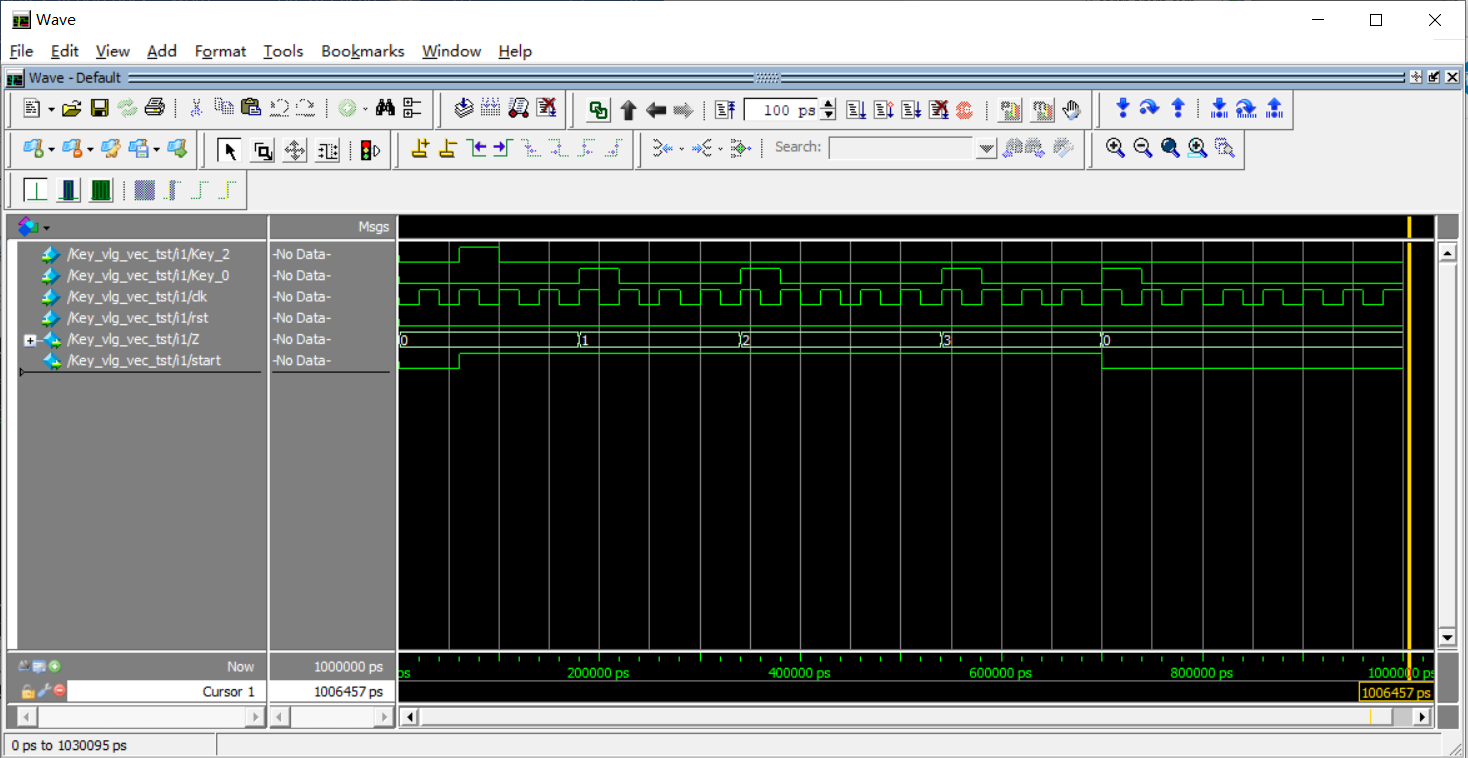
Figure 8 is the ASM chart of the 7-segment module, as a 7-segment decoder to display the number represented by the Bcd code on the DE2 board.



**Figure 8:** The ASM chart of 7-segment module

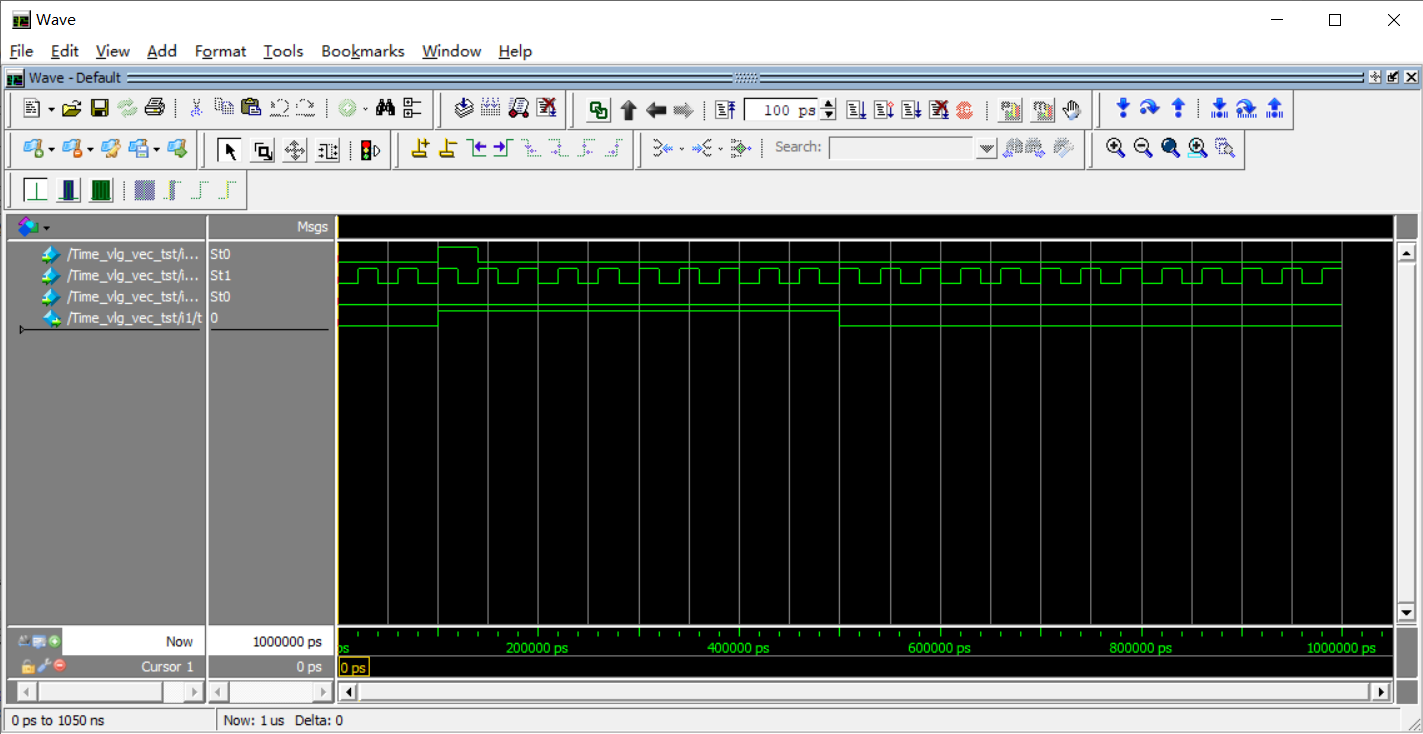
**3. Full simulations for each module**

3.1 Key module

Figure 9 is the result of simulating the key module. The Key\_X is a level signal. When Key\_2 is pressed, the start signal will change to ‘1’. After pressing key0 four times, start will change back to ‘0’, ending the system. Z keeps following Key\_0, and when you press Key0 for the fourth time, the system ends.

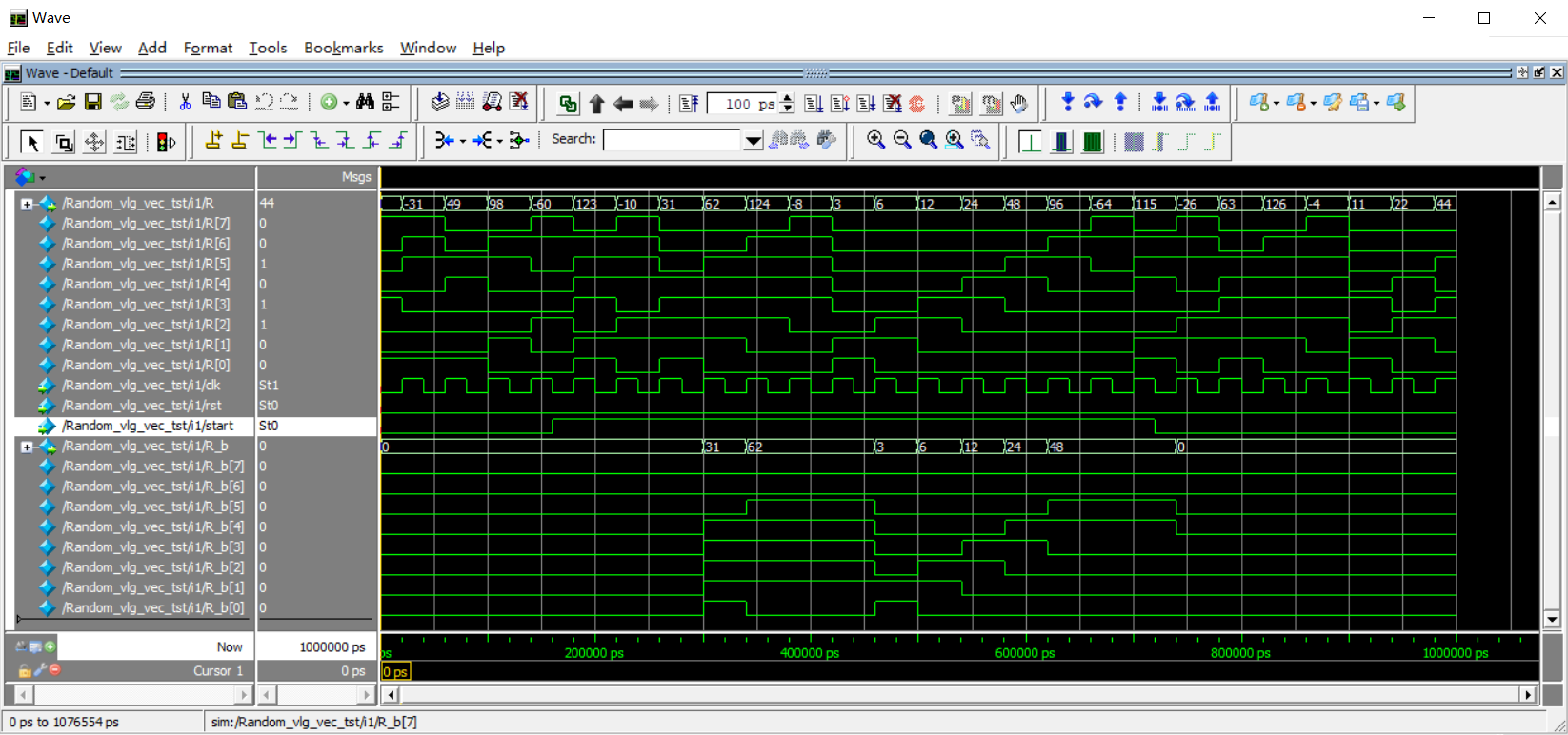
**Figure 9:** The simulation of Key module

3.2 Time module

Figure 10 is the result of simulating the Time module. When Key\_0 is pressed, the t signal will change to ‘1’, and Y will start counting. When Y counts to the set number, t will change back to ‘0’ and the DE2 board will not display the number. Therefore, the system uses Y to determine how long the number needs to be displayed.

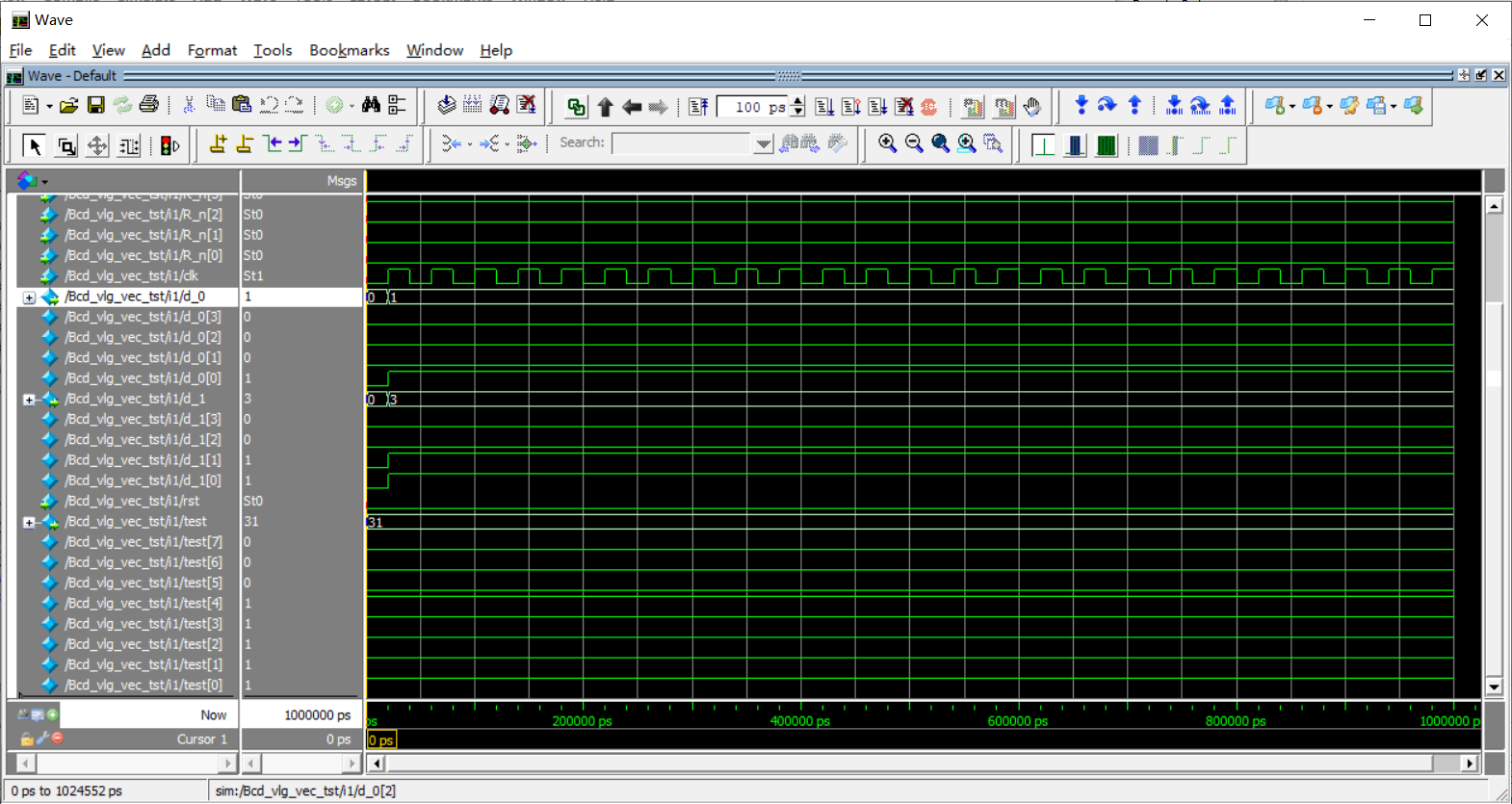
**Figure 10:** The simulation of Time module

3.3 Random module

Figure 11 is the result of simulating the Random module. When the start signal is ‘1’, only numbers between 0 and 73 are randomly output. When the start signal is ‘0’, the output remains unchanged.

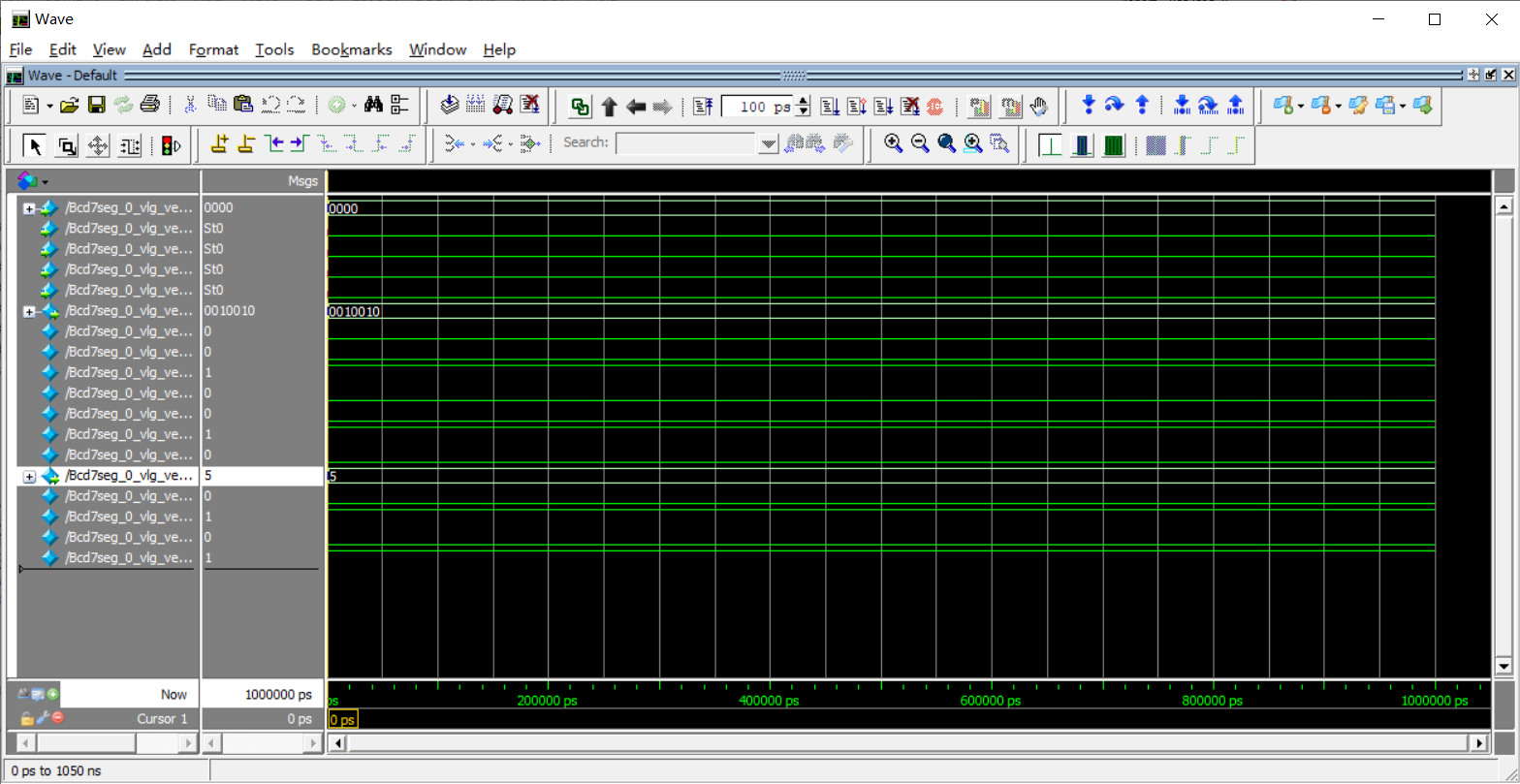
**Figure 11:** The simulation of Random module

3.4 Bcd module

Figure 12 is the result of simulating the Bcd module. Input a test value test = 8b'0001 1111 (=8’d31), the output obtained are d\_1 = 4’b0011 (tens =8’d3) and d\_0 = 4’b0001 (ones =8’d1) respectively.

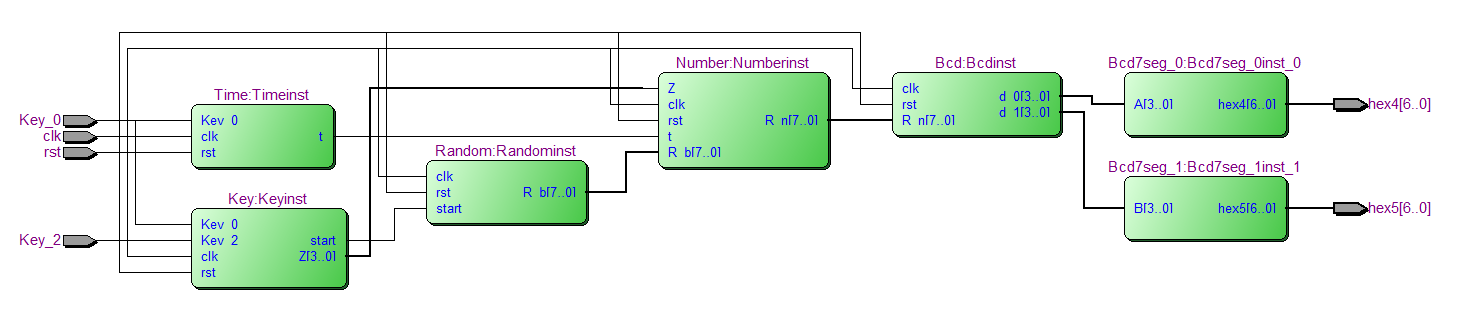
**Figure 12:** The simulation of Bcd module

3.5 7-segment module

Figure 13 is the result of simulating the 7-segment module. Input a test value test =4d'5, the output obtained is hex\_X = 7’b0010010, which conforms to the code.

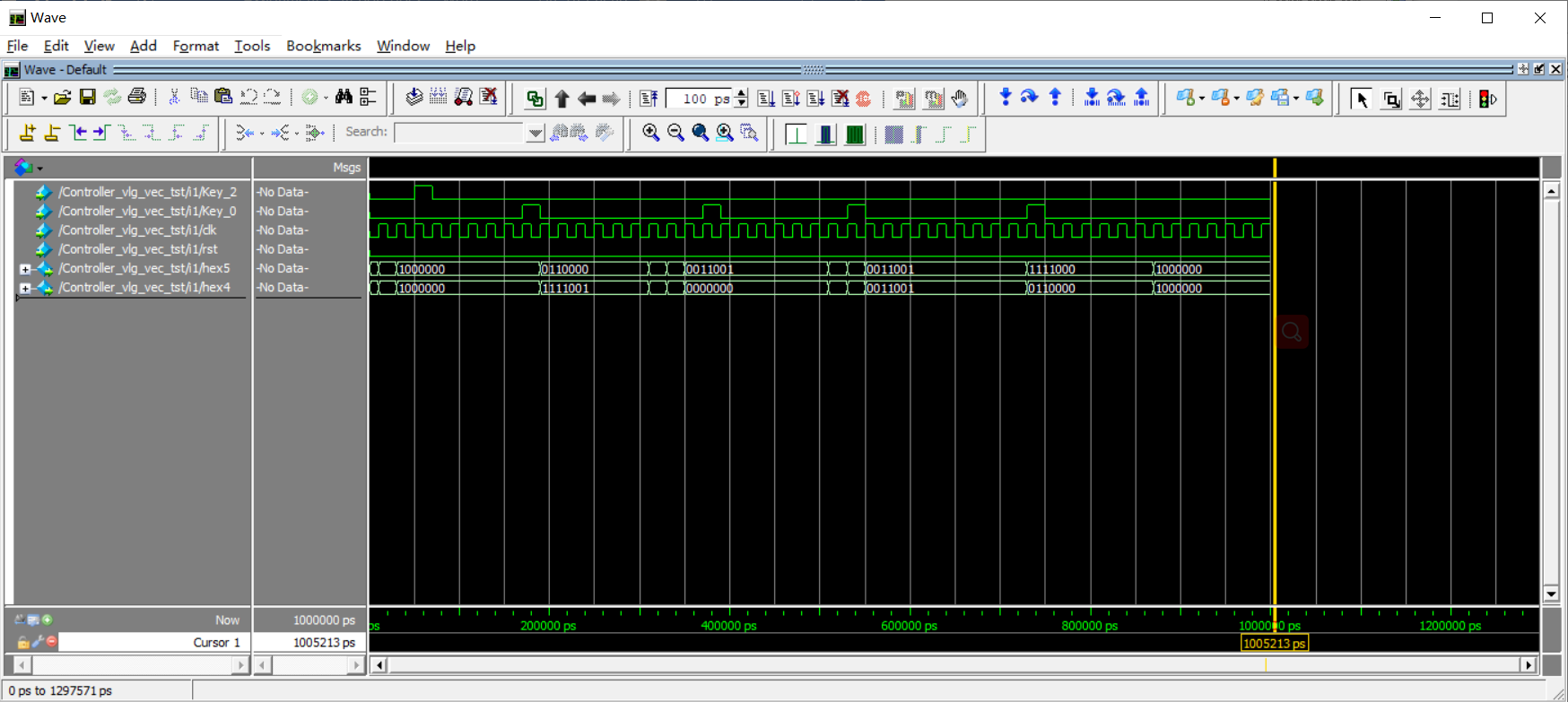
**Figure 13:** The simulation of 7-segment module

**4. RTL Schematic**

Figure 14 is an RTL viewer of the entire system.

**Figure 14:** The RTL Schematic of the full system

**5. Simulation of the full system**

Figure 15 is the simulation result of the entire system. Two keys are input, and a tens digit in HEX\_5 and a single digit in HEX\_4 are output respectively. It starts when Key\_2 is pressed and the system starts to operate. End the system when Key\_0 is pressed four times. In addition, every time you press Key\_0, it will keep displaying for several seconds.

**Figure 15:** The simulation of the entire system

**6. Result**

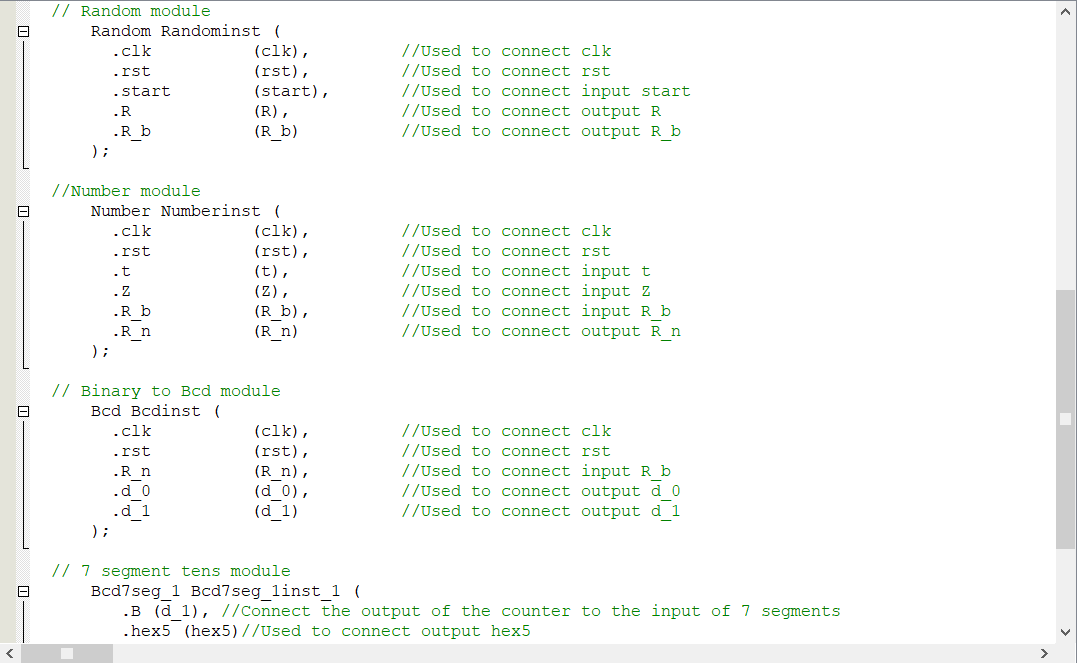
It can be seen from the simulated experimental results that each module can operate normally. But in the final simulation of the entire system, it can be seen that there is a little delay problem. According to the random number in Figure 11, it takes some cycles to select a number between 0 and 73 from the random number, and not every cycle has an allowable number to output. Therefore, when key2 is pressed, no new number is displayed immediately. Since we are all operating and viewing in a few nanoseconds, if we observe in one second, we should not see any errors. Finally, when the system did press Key\_0 for the fourth time, it stopped after waiting for the number to be displayed.

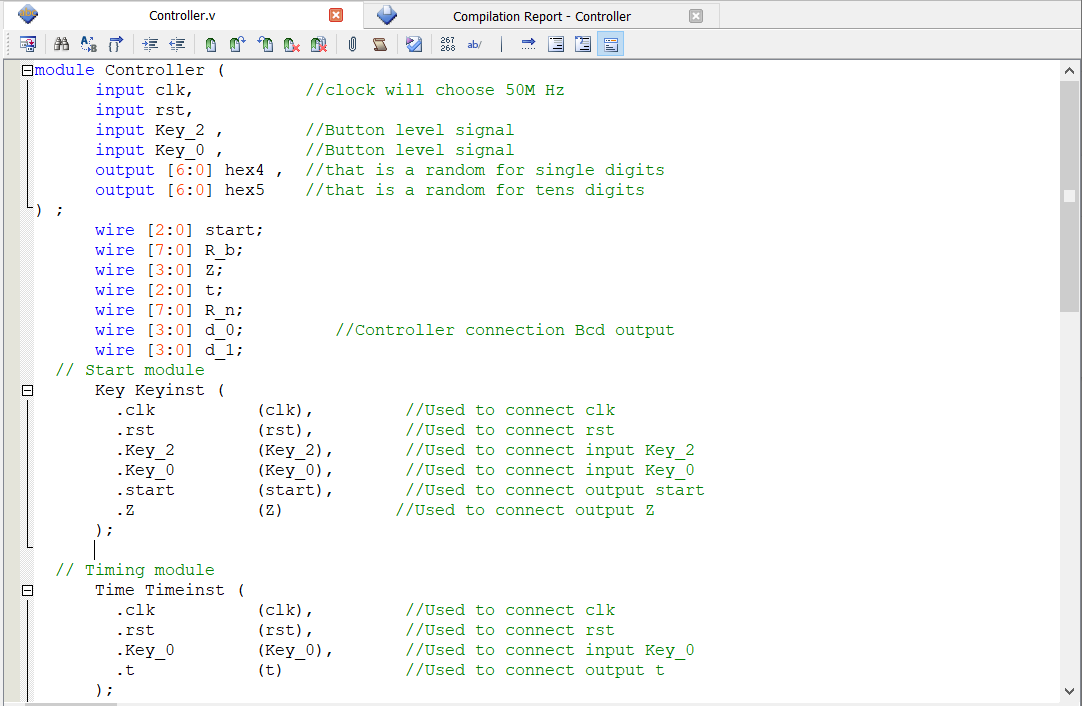
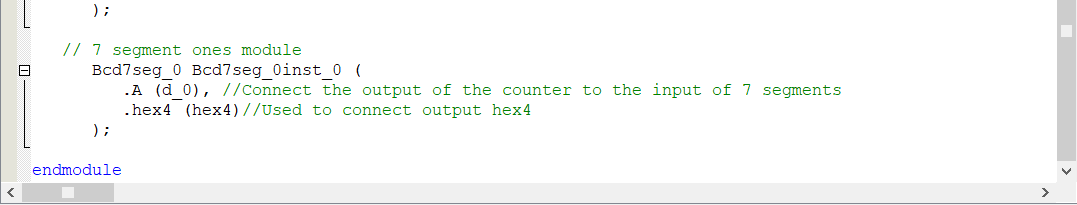
The random number generation uses LFSR to generate random numbers, which requires a relatively long cycle period. LFSR is composed of n D flip-flops and several XOR gates. There are 8 D flip-flops in the system, so 256 numbers (-128 to 128) are randomly generated, but the sequence of randomly generated numbers will not change unless the initial value is changed. However, the Number module is used to avoid displaying repeated numbers.

**7. Conclusion**

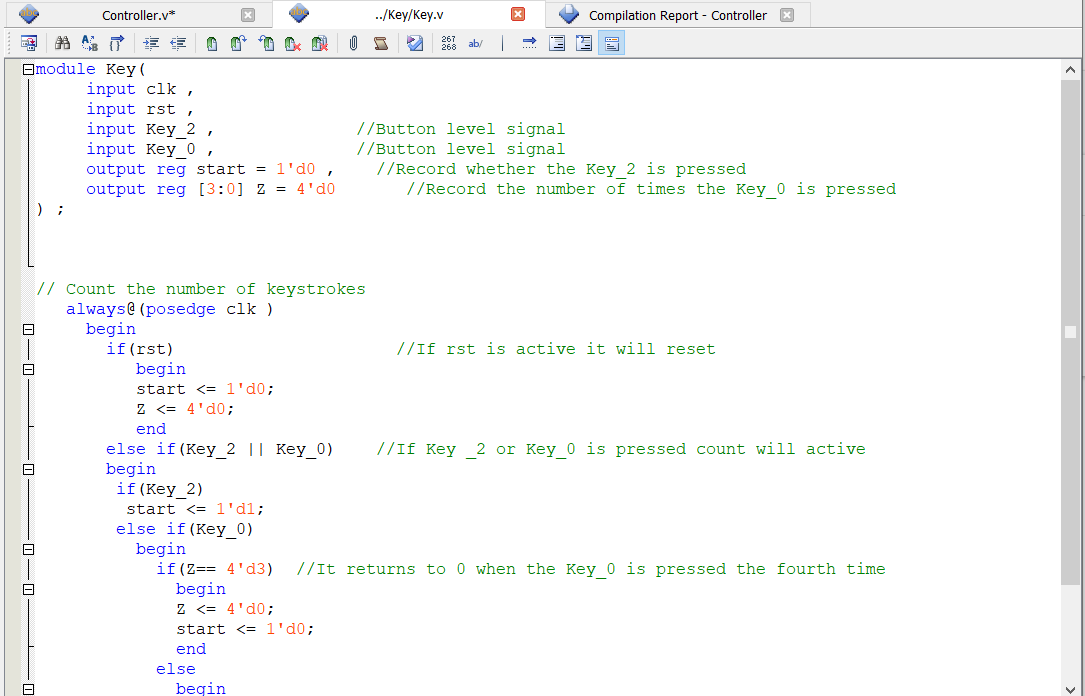
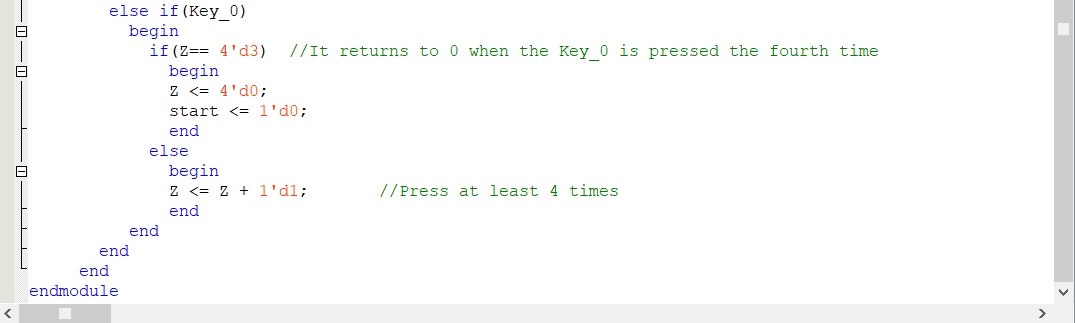
After completing Assignment 2, I am more familiar with the use of quartus. Compared with Assignment 1, many modules and functions have been added to the system. Assignment 2 is mainly to generate random numbers. It would be ideal if the generation and selection of numbers can be completed in one cycle. If you want to change the random sequence of the random module, you can add a counter and change the initial value after counting to 256, so the random sequence will change, but the initial value cannot be 0.

**8. Coding**

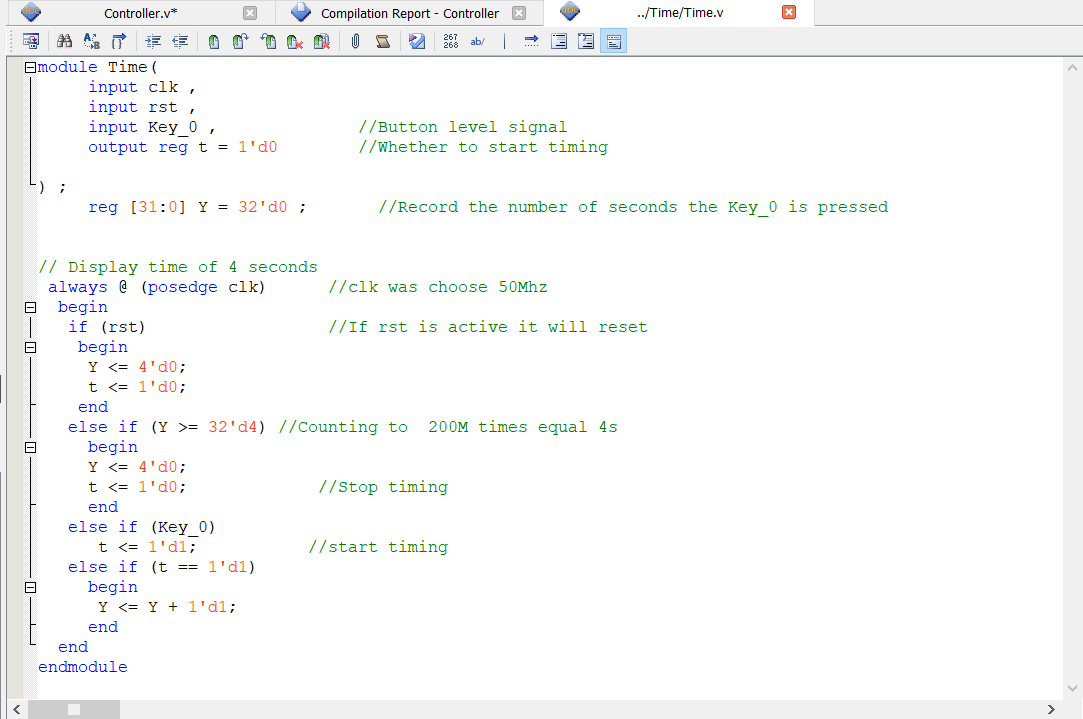
8.1 Controller Module



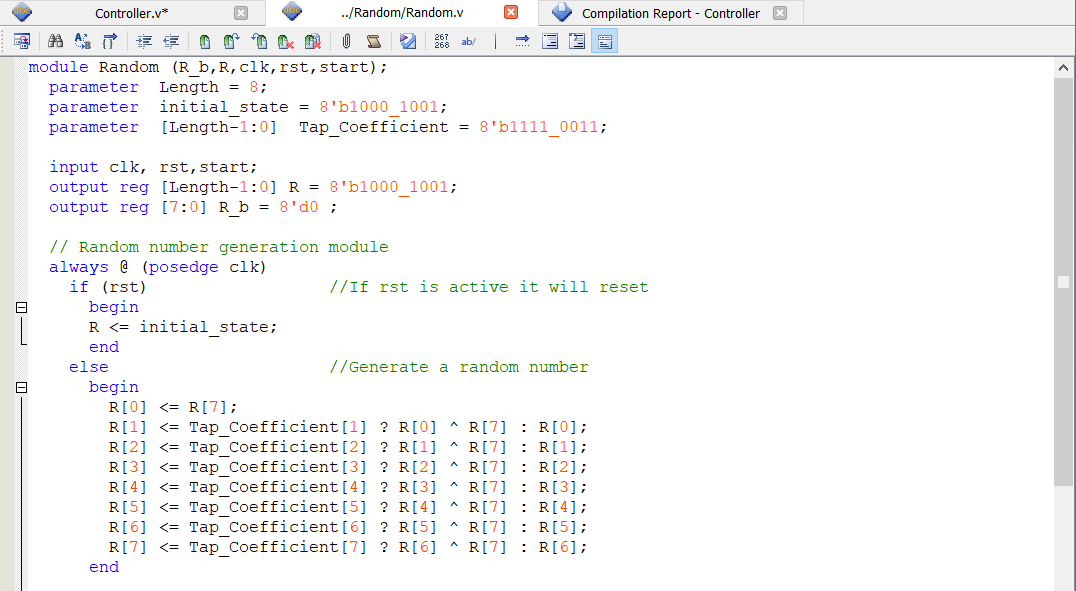
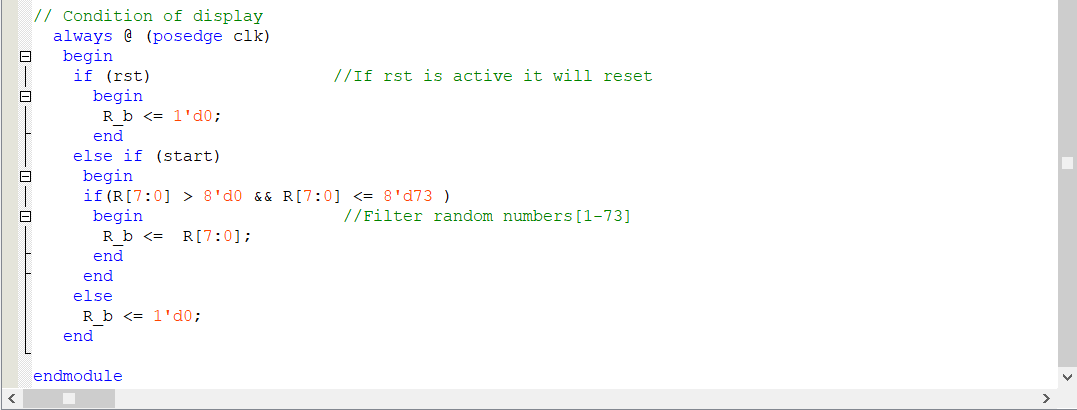
8.2 Key Module



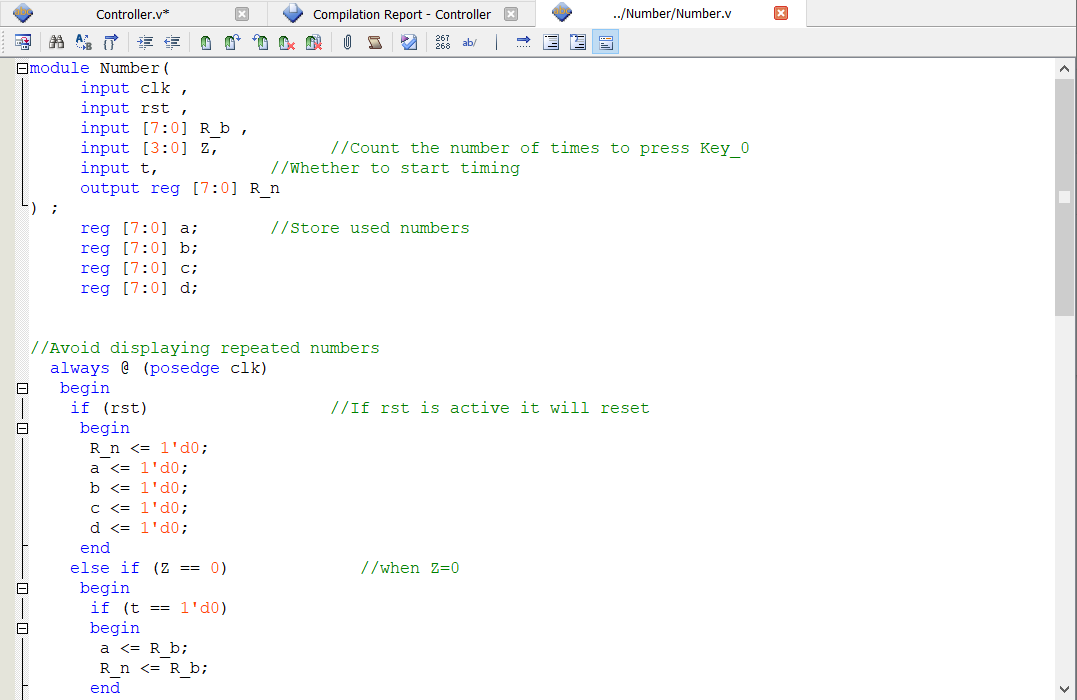
8.3 Time Module

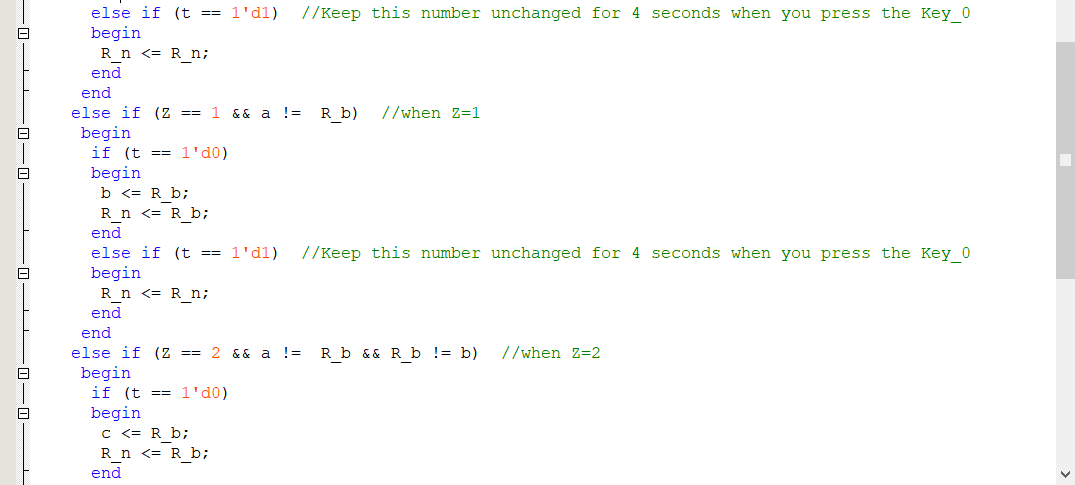


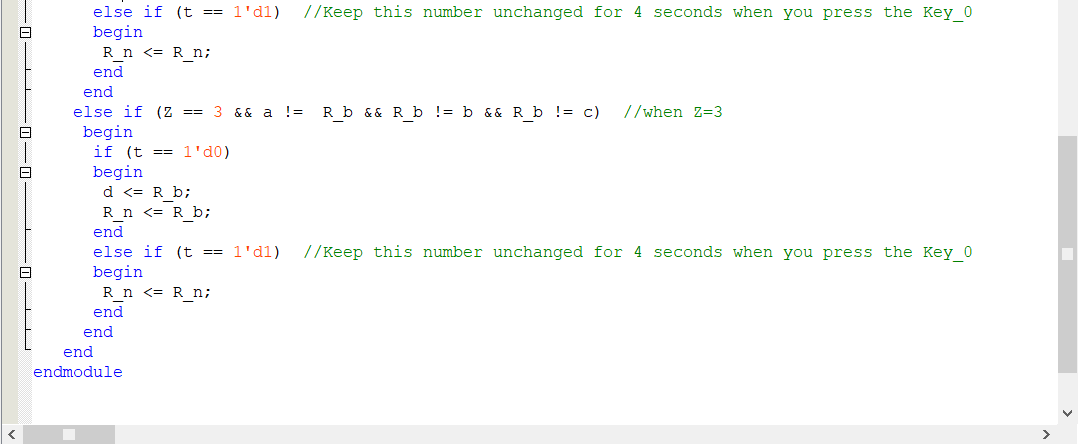
8.4 Random Module

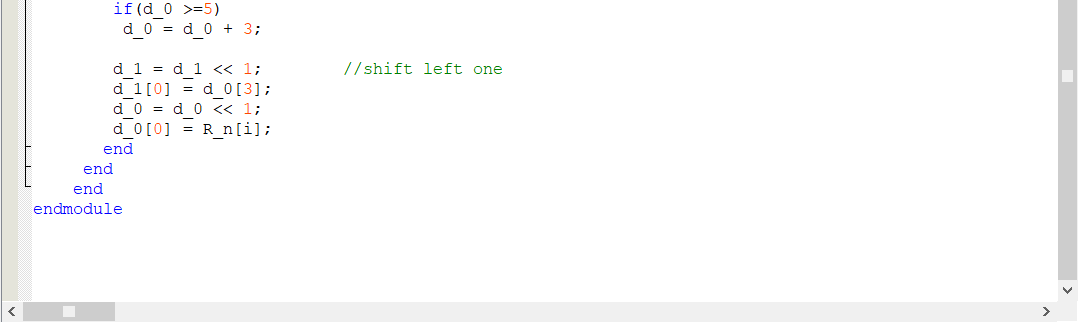


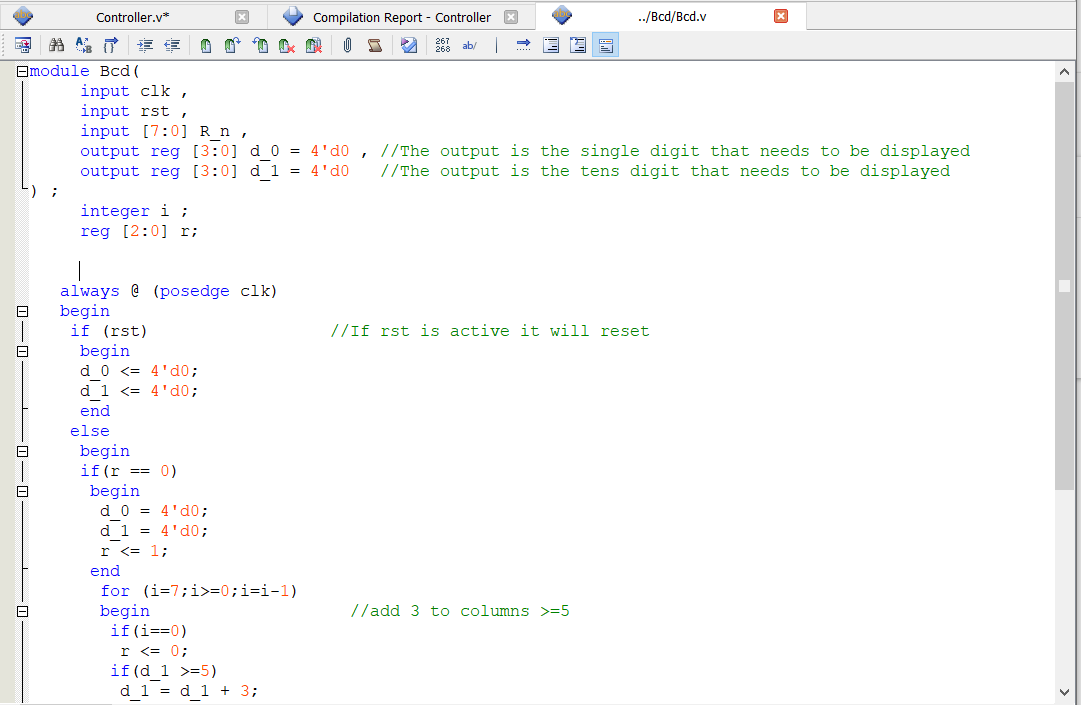
8.5 Number Module



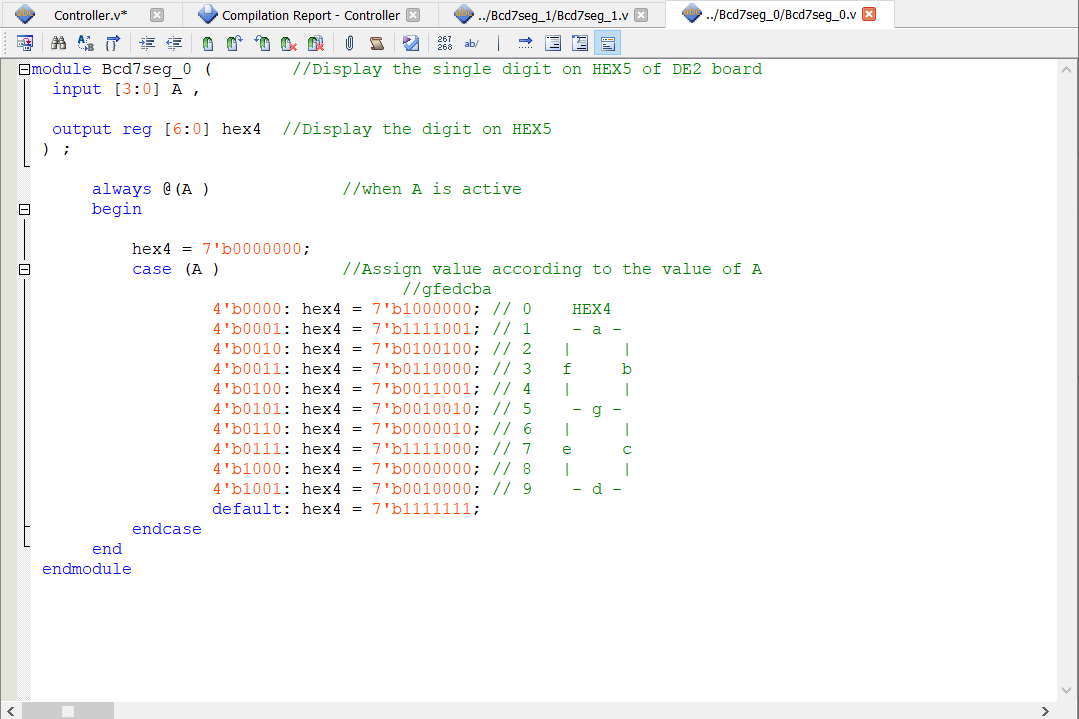




8.6 Bcd Module



8.7 7-segment\_0 Module



8.8 7-segment\_1 Module

