VHDL Assignment #4: Design and Implementation of a 4-Bit Comparator

Anja Kroon 260886624
Matt Zeitouni 260930709

Question 1:

The design file **four_bit_comparator.vhd** begins with library declarations and the entity declaration. There are two 4-bit unsigned integer inputs (A and B) and 6 1-bit unsigned integer outputs that represent the relationship between A and B+1 (less than, less than or equal, greater than, greater than or equal, or equal to). Next is the architecture declaration. There, B+1 is computed using B and "0001" as inputs to a ripple carry adder. A component declaration of the ripple carry adder is therefore needed. This links the four_bit_comparator.vhd file to the rca_behavioral.vhd file. The ripple carry adder returns an output B_one, which is the 5-bit unsigned integer sum of B and "0001". There is an additional bit which holds the overflow. Because B_one is a 5-bit unsigned integer, A must also be made into a 5-bit unsigned integer to ensure that the comparison is executed correctly. This is accomplished by declaring a new signal A_new and appending '0' to A as the MSB. Next is the process block which will be activated as either B_one or A_new changes. The process block does the comparison of A_new and B_one using four scenarios evaluated sequentially with if-then-else statements. The signals assignments inside the process block take effect only once all the statements in the process block have been evaluated. The scenarios are as follows:

- **Overflow:** If the MSB of the B_one digit is '1', then overflow has occurred in the RCA. In the case that overflow occurs, all of the outputs are '0' except for overflow which is '1'.
- **Greater-than:** If A_new is greater than B_one, two outputs are assigned values of '1': AgtBplusOne and AgteBplusOne. AgtBplusOne refers to the case where A_new is greater than B_one. AgteBplusOne refers to the case where A_new is greater than or equal to B_one. All other outputs are '0'.
- **Less-than:** If A_new is less than B_one, two outputs are assigned values of '1': AltBplusOne and AlteBplusOne. AltBplusOne refers to the case where A_new is less than B_one. AlteBplusOne refers to the case where A_new is less than or equal to B_one. All other outputs are '0'.
- **Equals:** This is the last scenario in the process block and thus gets activated if the three previous cases are false. It is represented with an else statement. For the range of expected inputs of both A and B (4 bit binary numbers), this statement gets activated when A_new is equal to B_one. When A_new equals B_one, three outputs are assigned values of '1': AgteBplusOne, AlteBplusOne, AeqBplusOne. AgteBplusOne refers to the case where A_new is greater than or equal to B_one, AlteBplusOne refers to the case where A_new is less than or equal to B_one, AeqBplusOne refers to the case where A_new equals B_one.

In the **tb_four_bit_comparator.vhd**, we begin with the library and a blank entity declaration. Then, the architecture begins with a component declaration to connect the testbench to the four_bit_comparator.vhd file. Following are the signal declarations for all inputs and all outputs. A_1 and B_1, both instantiated as "0000" to ensure that there is always an input coming in. After, we move to instantiate the DUT with the signals declared. Then comes the process and begin statements. Following are the two nested for loops which iterate through all the possible input combinations. Given that each

input has 4 bits, the test bench iterates through 16 possible combinations for each variable, iterating through the 256 possible combinations for the two inputs.

Question 2:

For $A=5_{10}$, there are 16 cases in the 4 bit comparator as the only variable that can change is B plus one. The 16 possible outputs can be classified as one of the four groupings of outputs explained in Question 1: Overflow, Greater-than, Less-than and Equals. Simulation plots for each of the 4 cases are shown below.

Please note that the variable names used in the timing diagram are based on the names given in the assignment instructions. In addition, as explained in Question 1, the signal B_one refers to the output of the ripple carry adder when B and 0001 are summed together.

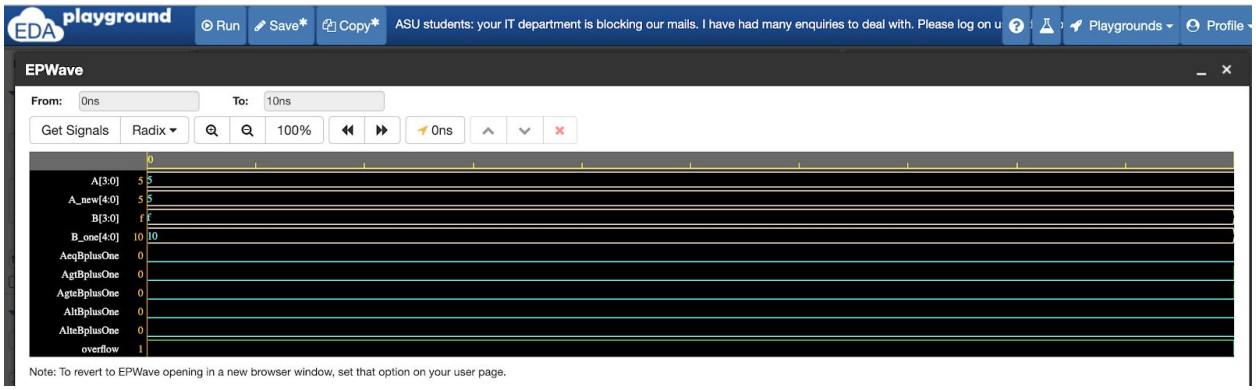

Figure 1: The Overflow Case occurs from 0 ns to 10 ns.

In Figure 1, the overflow output is activated. From the EPWave we see that B has a value of 'f'$_{16}$. For the comparator, however, we are comparing A and B+1. Therefore, we will have to take the B value and add 1. 15 + 1 = 16 which is a value that cannot be represented with 4 bits alone. Therefore, overflow has occurred (i.e. overflow = '1'). While overflow is activated, every other output is zeroed.

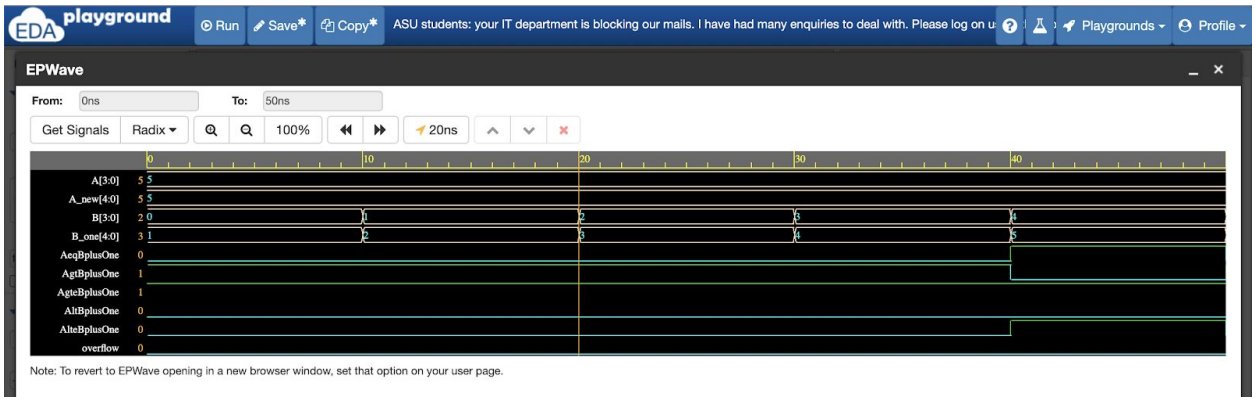

Figure 2: Greater-than Case occurs from 0 to 40 ns. Greater-than or Equal Case occurs from 0 to 50 ns.

In Figure 2, the greater than case is activated from 0 to 40 ns. This holds true as the values for B in this time frame range from 0 to 3. Recall that the comparator operates with B_one. Therefore, the output that A is only greater than B_one for values of B that range from 0 to 3 is correct. The greater than or equal to case occurs between 0 and 50 ns and is activated for one more iteration beyond the greater

case. Thus, the AgteBplusOne is ‘1’ for B values of 0 to 4. Examining the edge case, take B as 4, add one so B_one is 5, now compare A to B_one and you see they are equal thus the case AgteBplusOne holds true.

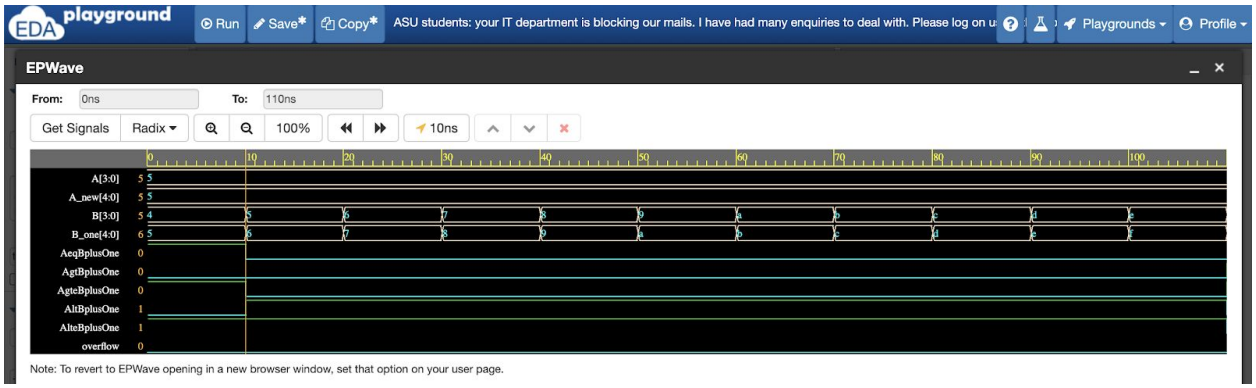

Figure 3: Less-than Case occurs from 10 to 110 ns. Less-than or Equal Case occurs from 0 to 110 ns.

In Figure 3, the less than case is activated between 10 and 110 ns. The values of B in this time frame range from 5 to 14. Checking an edge case of the less than, A is 5 and B_one is 6. Thus, A is less than B plus 1 and the waveform holds the correct output. The less than or equal case occurs between 0 and 110 ns and is activated one iteration before the less than case. It holds true for B values between 4 and 14. Examining the edge case of B = 4, B_one would then be 1+4 = 5 and then A is also 5. Thus, A is less than or equal to B_one.

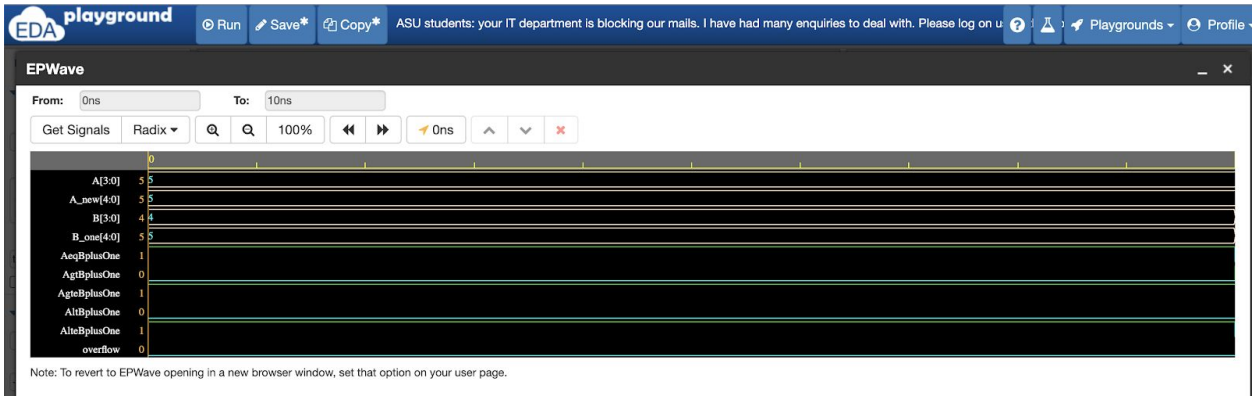

Figure 4: Equals Case occurs between 0 and 10 ns.

In Figure 4, the equals case when A = 5 occurs from 0 to 10 ns. In that time frame, B = 4. Therefore, B_one = 5 and we can conclude that A = B_one. When A = B_one, A is less than or equal to B_one, A is greater than or equal to B_one, and A is equal to B_one.

Question 3:

As per the timing analysis from the log content displayed below, the critical path of the circuit is from the input A(3) to the output port AgtBplusOne. The length of the critical path is 3 logic levels. The data arrival time for the critical path of this circuit is **5.024 nanoseconds.**

```
CTE report timing..
# Info:                    CTE Path Report
# Info: Critical path #1, (path slack = 9.976):, Logic Levels = 3
# Info: SOURCE CLOCK: name: <not found>  Path is min/max delay constrained
# Info: NAME                   GATE       DELAY    ARRIVAL DIR  FANOUT LEVEL
# Info: A(3)               (port)                 0.000   dn
# Info: A(3)               (net)        0.000                    1     0
# Info: A_ibuf(3)/I        IBUF                   0.000   dn
# Info: A_ibuf(3)/O        IBUF         1.298     1.298   dn
# Info: A_int(3)           (net)        0.378                    5     1
# Info: ix53065z10052/I0   LUT6                   1.676   dn
# Info: ix53065z10052/O    LUT6         0.124     1.800   dn
# Info: nx53065z1          (net)        0.333                    1     2
# Info: AgtBplusOne_obuf/I OBUF                   2.133   dn
# Info: AgtBplusOne_obuf/O OBUF         2.891     5.024   dn
# Info: AgtBplusOne        (net)        0.000                    0     3
# Info: AgtBplusOne        (port)                 5.024   dn
# Info:           Minmax delay constraint:     15.000
# Info:           Source clock delay:      -    0.000
# Info:           Dest clock delay:        +    0.000
# Info:                                    -----------
# Info:           Edge separation:             15.000
# Info:           Setup constraint:        -    0.000
# Info:                                    -----------
# Info:           Data required time:          15.000
# Info:           Data arrival time:       -    5.024   ( 85.85% cell delay, 14.15% net delay )
# Info:                                    -----------
# Info:           Slack:                       9.976
# Info: End CTE Analysis ..... CPU Time Used: 0 sec.
# Info: //
```

Question 4:

Number of Pins: 14

Number of LUTs: 9

A snapshot of the log content is shown below where the pins and LUTs are reported.

```
# Info: Device Utilization for 7A100TCSG324
# Info: ***********************************************************
# Info: Resource                          Used    Avail   Utilization
# Info: -----------------------------------------------------------
# Info: IOs                               14      210       6.67%
# Info: Global Buffers                    0       32        0.00%
# Info: LUTs                              9       63400     0.01%
# Info: CLB Slices                        2       15850     0.01%
# Info: Dffs or Latches                   0       126800    0.00%
# Info: Block RAMs                        0       135       0.00%
# Info: DSP48E1s                          0       240       0.00%
# Info: -----------------------------------------------------------
# Info: ************************************************************************
# Info: Library: work    Cell: design    View: four_bit_comparator_arc
# Info: ************************************************************************
# Info:  Number of ports :                            14
# Info:  Number of nets :                             31
# Info:  Number of instances :                        23
# Info:  Number of references to this view :           0
# Info: Total accumulated area :
# Info:  Number of LUTs :                              9
# Info:  Number of LUTs with LUTNM/HLUTNM :            2
# Info:  Number of accumulated instances :            23
# Info: ****************************
# Info:  IO Register Mapping Report
# Info: ****************************
# Info: Design: work.design.four_bit_comparator_arc
# Info: +-----------------+-----------+----------+----------+----------+
# Info: | Port            | Direction |   INFF   |  OUTFF   |  TRIFF   |
# Info: +-----------------+-----------+----------+----------+----------+
```