**ELEC 204 Digital Design Lab Report**

Lab 02

Name-Surname: Barış KAPLAN, ID Number: 69054

Date of Submission: 03/20/2021(20 March 2021)

1. **Introduction and objectives**

Explain the objectives of the lab (refer to the lab instruction sheet),

Explain what your code has to do and describe how you did it.

**The objectives of the lab**

**1-) Learning how to implement the VHDL code of the 4-bit magnitude comparator for unsigned binary numbers.**

**2-) Learning how to implement the VHDL code of the 4-bit comparator for signed binary numbers.**

**3-) Learning how to construct possible input pairs by using VHDL Test Bench.**

**4-) Depending on the input pair, observing what the output values of E,GAB,GBA are for the MComparator question.**

**5-) Depending on the input pair, observing what the output values of E,GCD,GDC are for the SComparator question.**

**6-) Learning how to draw and design an equality comparator circuit for 4-bit binary numbers.**

**7-) Learning how to draw and design a 4-bit magnitude comparator circuit for unsigned binary numbers.**

**8-) Learning how to draw and design a 4-bit comparator circuit for signed binary numbers.**

**9-) Making us familiar with the basic logic gates while designing 4-bit comparator circuits for unsigned and 2’s complement systems.**

1. **Methods**

Explain the inputs (how many bits, names of the inputs),

**INPUTS**

**UNSIGNED(MComparator)**

**There are 2 inputs in the implementation of the 4-bit magnitude comparator circuit for unsigned binary numbers. One of these inputs is A. Another input is B. A is a 4-bit input. B is also a 4-bit input.**

**SIGNED(SComparator)**

**There are 2 inputs in the implementation of the 4-bit comparator circuit for signed binary numbers. One of these inputs is C. Another input is D. C is a 4-bit input. Moreover, D is a 4-bit input.**

**OUTPUTS**

**UNSIGNED(MComparator)**

**There are 3 outputs in the implementation of the 4-bit magnitude comparator circuit for unsigned binary numbers. One of these outputs is E. E returns a value according to the equality of A and B. If they are equal, E returns 1. Otherwise, E returns 0. The second output is GAB. GAB returns a value depending on whether A is greater than B. If A is greater than B, GAB returns 1. Otherwise, GAB returns 0. The third and last output is GBA. GBA returns a value depending on whether B is greater than A. If B is greater than A, GBA returns 1. Otherwise, GBA returns 0. E is a 1-bit output. GAB is a 1-bit output. GBA is a 1-bit output.**

**SIGNED(SComparator)**

**There are 3 outputs in the implementation of the 4-bit comparator circuit for signed binary numbers. One of these outputs is E. E returns a value according to the equality of C and D. If they are equal, E returns 1. Otherwise, E returns 0. The second output is GCD. GCD returns a value depending on whether C is greater than D. If C is greater than D, GCD returns 1. Otherwise, GCD returns 0. The third and last output is GDC. GDC returns a value depending on whether D is greater than C. If D is greater than C, GDC returns 1. Otherwise, GDC returns 0. E is a 1-bit output. GCD is a 1-bit output. GDC is a 1-bit output.**

**WHAT THE VHDL CODE MUST DO**

**For MComparator, if A and B are equal, the VHDL code must give the output E to be 1, it must give the output GAB to be 0, and it must give the output GBA to be 0. Moreover, if A is greater than B, it must give the output GAB to be 1, GBA to be 0, and E to be 0. If A is less than B, it must give the output GBA to be 1, GAB to be 0, and E to be 0.**

**For SComparator, if C and D are equal, the VHDL code must give the output E to be 1, the output GAB to be 0, and the output GBA to be 0. In addition to that, if C is greater than D, it must give the output GCD to be 1, GDC to be 0, and E to be 0. If C is smaller than D, it must give the output GCD to be 0, GDC to be 1, and E to be 0.**

**HOW MY CODE WORKS**

**In my VHDL MComparator Implementation, firstly, I defined some signals with their names. Then, I consider the possible cases where the input A is greater than the input B and I also consider the possible cases where the input A is smaller than the input B. For the case where A is equal to B, I have connected A(0) and B(0) with an XOR gate, A(1) and B(1) with an XOR gate, A(2) and B(2) with an XOR gate, A(3) and B(3) with an XOR gate. I have taken the complement of the results coming from the XOR gates. I have assigned the signal names which are e0, e1, e2, and e3 to these results. At the end, to get the output E, I have connected e0, e1, e2 and e3 with an AND gate. For the case where A is greater than B, first possible situation is A(3)>B(3). For this situation, I have connected A(3) and**

**not B(3) with an AND gate. Second possible situation is A(3)=B(3), A(2)>B(2). For this situation, I have connected e3, A(2), and not B(2) with an AND gate. Third possible situation is A(3)=B(3), A(2)=B(2), A(1)>B(1). For this situation, I have connected e3, e2, A(1) and not B(1) with an AND gate. The fourth and last possible situation is A(3)=B(3), A(2)=B(2), A(1)=B(1), and A(0)>B(0). For this situation, I have connected e3, e2, e1, A(0), and not B(0) with an AND gate. At the end, I have connected the results coming from the AND gates with an OR gate.**

**Possible situations where A>B**

**1-) A(3)>B(3)**

**2-) A(3)=B(3), A(2)>B(2)**

**3-) A(3)=B(3), A(2)=B(2), A(1)>B(1)**

**4-) A(3)=B(3), A(2)=B(2), A(1)=B(1), A(0)>B(0)**

**The gate is X and (not Y). (FOR X>Y)**

**Possible situations where A<B**

**1-) A(3)<B(3)**

**2-) A(3)=B(3), A(2)<B(2)**

**3-) A(3)=B(3), A(2)=B(2), A(1)<B(1)**

**4-) A(3)=B(3), A(2)=B(2), A(1)=B(1), A(0)<B(0)**

**The gate is Y and (not X). (FOR Y>X)**

**Possible situations where A=B**

**1-) A(3)=B(3), A(2)=B(2), A(1)=B(1), A(0)=B(0)**

**In my VHDL SComparator implementation, there are some small differences compared to the VHDL MComparator implementation. For the SComparator implementation, we should do the following bit conversions :**

**not C(3)->C(3)**

**C(3)->not C(3)**

**D(3)->not D(3)**

**not D(3)->D(3)**

**I applied nearly same logic in the SComparator implementation; however, I do the bit replacements above in the SComparator Implementation.**

**SIGNED UNSIGNED**

**1000: -8 0000: 0//For the signed and unsigned numbers in this row, only MSB differs.**

**1001: -7 0001: 1//For the signed and unsigned numbers in this row, only MSB differs.**

**1010: -6 0010: 2//For the signed and unsigned numbers in this row, only MSB differs.**

**Note: When we keep continue to increment the numbers by one and write the next one for 4-bit unsigned and 4-bit signed numbers, we will see that for every row, only MSB differs.**

**MSB= Most Significant Bit= Left-most bit.**

**Explain how your code works**

**My codes for MComparator and for SComparator have outputs as expected. I have explained the logic in the “HOW MY CODE WORKS” part.**

**When A and B are equal, my code for MComparator has an output:**

**E=1**

**GAB=0**

**GBA=0**

**When A is greater than B, my code for MComparator has an output:**

**E=0**

**GAB=1**

**GBA=0**

**When A is smaller than B, my code for MComparator has an output:**

**E=0**

**GAB=0**

**GBA=1**

**When C and D are equal, my code for SComparator has an output:**

**E=1**

**GCD=0**

**GDC=0**

**When C is greater than D, my code for MComparator has an output:**

**E=0**

**GCD=1**

**GDC=0**

**When D is greater than C, my code for MComparator has an output:**

**E=0**

**GCD=0**

**GDC=1**

**TRUTH TABLE FOR M COMPARATOR (UNSIGNED)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **GAB** | **GBA** | **E** |
| **0000** | **0000** | **0** | **0** | **1** |
| **0000** | **0001** | **0** | **1** | **0** |
| **0000** | **0010** | **0** | **1** | **0** |
| **0000** | **0011** | **0** | **1** | **0** |
| **0000** | **0100** | **0** | **1** | **0** |
| **0000** | **0101** | **0** | **1** | **0** |
| **0000** | **0110** | **0** | **1** | **0** |
| **0000** | **0111** | **0** | **1** | **0** |
| **0000** | **1000** | **0** | **1** | **0** |
| **0000** | **1001** | **0** | **1** | **0** |
| **0000** | **1010** | **0** | **1** | **0** |
| **0000** | **1011** | **0** | **1** | **0** |
| **0000** | **1100** | **0** | **1** | **0** |
| **0000** | **1101** | **0** | **1** | **0** |
| **0000** | **1110** | **0** | **1** | **0** |
| **0000** | **1111** | **0** | **1** | **0** |
| **0001** | **0000** | **1** | **0** | **0** |
| **0001** | **0001** | **0** | **0** | **1** |
| **0001** | **0010** | **0** | **1** | **0** |
| **0001** | **0011** | **0** | **1** | **0** |
| **0001** | **0100** | **0** | **1** | **0** |
| **0001** | **0101** | **0** | **1** | **0** |
| **0001** | **0110** | **0** | **1** | **0** |
| **0001** | **0111** | **0** | **1** | **0** |
| **0001** | **1000** | **0** | **1** | **0** |
| **0001** | **1001** | **0** | **1** | **0** |
| **0001** | **1010** | **0** | **1** | **0** |
| **0001** | **1011** | **0** | **1** | **0** |
| **0001** | **1100** | **0** | **1** | **0** |
| **0001** | **1101** | **0** | **1** | **0** |
| **0001** | **1110** | **0** | **1** | **0** |
| **0001** | **1111** | **0** | **1** | **0** |
| **0010** | **0000** | **1** | **0** | **0** |
| **0010** | **0001** | **1** | **0** | **0** |
| **0010** | **0010** | **0** | **0** | **1** |
| **0010** | **0011** | **0** | **1** | **0** |
| **0010** | **0100** | **0** | **1** | **0** |
| **0010** | **0101** | **0** | **1** | **0** |
| **0010** | **0110** | **0** | **1** | **0** |
| **0010** | **0111** | **0** | **1** | **0** |
| **0010** | **1000** | **0** | **1** | **0** |
| **0010** | **1001** | **0** | **1** | **0** |
| **0010** | **1010** | **0** | **1** | **0** |
| **0010** | **1011** | **0** | **1** | **0** |
| **0010** | **1100** | **0** | **1** | **0** |
| **0010** | **1101** | **0** | **1** | **0** |
| **0010** | **1110** | **0** | **1** | **0** |
| **0010** | **1111** | **0** | **1** | **0** |
| **0011** | **0000** | **1** | **0** | **0** |
| **0011** | **0001** | **1** | **0** | **0** |
| **0011** | **0010** | **1** | **0** | **0** |
| **0011** | **0011** | **0** | **0** | **1** |
| **0011** | **0100** | **0** | **1** | **0** |
| **0011** | **0101** | **0** | **1** | **0** |
| **0011** | **0110** | **0** | **1** | **0** |
| **0011** | **0111** | **0** | **1** | **0** |
| **0011** | **1000** | **0** | **1** | **0** |
| **0011** | **1001** | **0** | **1** | **0** |
| **0011** | **1010** | **0** | **1** | **0** |
| **0011** | **1011** | **0** | **1** | **0** |
| **0011** | **1100** | **0** | **1** | **0** |
| **0011** | **1101** | **0** | **1** | **0** |
| **0011** | **1110** | **0** | **1** | **0** |
| **0011** | **1111** | **0** | **1** | **0** |
| **0100** | **0000** | **1** | **0** | **0** |
| **0100** | **0001** | **1** | **0** | **0** |
| **0100** | **0010** | **1** | **0** | **0** |
| **0100** | **0011** | **1** | **0** | **0** |
| **0100** | **0100** | **0** | **0** | **1** |
| **0100** | **0101** | **0** | **1** | **0** |
| **0100** | **0110** | **0** | **1** | **0** |
| **0100** | **0111** | **0** | **1** | **0** |
| **0100** | **1000** | **0** | **1** | **0** |
| **0100** | **1001** | **0** | **1** | **0** |
| **0100** | **1010** | **0** | **1** | **0** |
| **0100** | **1011** | **0** | **1** | **0** |
| **0100** | **1100** | **0** | **1** | **0** |
| **0100** | **1101** | **0** | **1** | **0** |
| **0100** | **1110** | **0** | **1** | **0** |
| **0100** | **1111** | **0** | **1** | **0** |
| **0101** | **0000** | **1** | **0** | **0** |
| **0101** | **0001** | **1** | **0** | **0** |
| **0101** | **0010** | **1** | **0** | **0** |
| **0101** | **0011** | **1** | **0** | **0** |
| **0101** | **0100** | **1** | **0** | **0** |
| **0101** | **0101** | **0** | **0** | **1** |
| **0101** | **0110** | **0** | **1** | **0** |
| **0101** | **0111** | **0** | **1** | **0** |
| **0101** | **1000** | **0** | **1** | **0** |
| **0101** | **1001** | **0** | **1** | **0** |
| **0101** | **1010** | **0** | **1** | **0** |
| **0101** | **1011** | **0** | **1** | **0** |
| **0101** | **1100** | **0** | **1** | **0** |
| **0101** | **1101** | **0** | **1** | **0** |
| **0101** | **1110** | **0** | **1** | **0** |
| **0101** | **1111** | **0** | **1** | **0** |
| **0110** | **0000** | **1** | **0** | **0** |
| **0110** | **0001** | **1** | **0** | **0** |
| **0110** | **0010** | **1** | **0** | **0** |
| **0110** | **0011** | **1** | **0** | **0** |
| **0110** | **0100** | **1** | **0** | **0** |
| **0110** | **0101** | **1** | **0** | **0** |
| **0110** | **0110** | **0** | **0** | **1** |
| **0110** | **0111** | **0** | **1** | **0** |
| **0110** | **1000** | **0** | **1** | **0** |
| **0110** | **1001** | **0** | **1** | **0** |
| **0110** | **1010** | **0** | **1** | **0** |
| **0110** | **1011** | **0** | **1** | **0** |
| **0110** | **1100** | **0** | **1** | **0** |
| **0110** | **1101** | **0** | **1** | **0** |
| **0110** | **1110** | **0** | **1** | **0** |
| **0110** | **1111** | **0** | **1** | **0** |
| **0111** | **0000** | **1** | **0** | **0** |
| **0111** | **0001** | **1** | **0** | **0** |
| **0111** | **0010** | **1** | **0** | **0** |
| **0111** | **0011** | **1** | **0** | **0** |
| **0111** | **0100** | **1** | **0** | **0** |
| **0111** | **0101** | **1** | **0** | **0** |
| **0111** | **0110** | **1** | **0** | **0** |
| **0111** | **0111** | **0** | **0** | **1** |
| **0111** | **1000** | **0** | **1** | **0** |
| **0111** | **1001** | **0** | **1** | **0** |
| **0111** | **1010** | **0** | **1** | **0** |
| **0111** | **1011** | **0** | **1** | **0** |
| **0111** | **1100** | **0** | **1** | **0** |
| **0111** | **1101** | **0** | **1** | **0** |
| **0111** | **1110** | **0** | **1** | **0** |
| **0111** | **1111** | **0** | **1** | **0** |
| **1000** | **0000** | **1** | **0** | **0** |
| **1000** | **0001** | **1** | **0** | **0** |
| **1000** | **0010** | **1** | **0** | **0** |
| **1000** | **0011** | **1** | **0** | **0** |
| **1000** | **0100** | **1** | **0** | **0** |
| **1000** | **0101** | **1** | **0** | **0** |
| **1000** | **0110** | **1** | **0** | **0** |
| **1000** | **0111** | **1** | **0** | **0** |
| **1000** | **1000** | **0** | **0** | **1** |
| **1000** | **1001** | **0** | **1** | **0** |
| **1000** | **1010** | **0** | **1** | **0** |
| **1000** | **1011** | **0** | **1** | **0** |
| **1000** | **1100** | **0** | **1** | **0** |
| **1000** | **1101** | **0** | **1** | **0** |
| **1000** | **1110** | **0** | **1** | **0** |
| **1000** | **1111** | **0** | **1** | **0** |
| **1001** | **0000** | **1** | **0** | **0** |
| **1001** | **0001** | **1** | **0** | **0** |
| **1001** | **0010** | **1** | **0** | **0** |
| **1001** | **0011** | **1** | **0** | **0** |
| **1001** | **0100** | **1** | **0** | **0** |
| **1001** | **0101** | **1** | **0** | **0** |
| **1001** | **0110** | **1** | **0** | **0** |
| **1001** | **0111** | **1** | **0** | **0** |
| **1001** | **1000** | **1** | **0** | **0** |
| **1001** | **1001** | **0** | **0** | **1** |
| **1001** | **1010** | **0** | **1** | **0** |
| **1001** | **1011** | **0** | **1** | **0** |
| **1001** | **1100** | **0** | **1** | **0** |
| **1001** | **1101** | **0** | **1** | **0** |
| **1001** | **1110** | **0** | **1** | **0** |
| **1001** | **1111** | **0** | **1** | **0** |
| **1010** | **0000** | **1** | **0** | **0** |
| **1010** | **0001** | **1** | **0** | **0** |
| **1010** | **0010** | **1** | **0** | **0** |
| **1010** | **0011** | **1** | **0** | **0** |
| **1010** | **0100** | **1** | **0** | **0** |
| **1010** | **0101** | **1** | **0** | **0** |
| **1010** | **0110** | **1** | **0** | **0** |
| **1010** | **0111** | **1** | **0** | **0** |
| **1010** | **1000** | **1** | **0** | **0** |
| **1010** | **1001** | **1** | **0** | **0** |
| **1010** | **1010** | **0** | **0** | **1** |
| **1010** | **1011** | **0** | **1** | **0** |
| **1010** | **1100** | **0** | **1** | **0** |
| **1010** | **1101** | **0** | **1** | **0** |
| **1010** | **1110** | **0** | **1** | **0** |
| **1010** | **1111** | **0** | **1** | **0** |
| **1011** | **0000** | **1** | **0** | **0** |
| **1011** | **0001** | **1** | **0** | **0** |
| **1011** | **0010** | **1** | **0** | **0** |
| **1011** | **0011** | **1** | **0** | **0** |
| **1011** | **0100** | **1** | **0** | **0** |
| **1011** | **0101** | **1** | **0** | **0** |
| **1011** | **0110** | **1** | **0** | **0** |
| **1011** | **0111** | **1** | **0** | **0** |
| **1011** | **1000** | **1** | **0** | **0** |
| **1011** | **1001** | **1** | **0** | **0** |
| **1011** | **1010** | **1** | **0** | **0** |
| **1011** | **1011** | **0** | **0** | **1** |
| **1011** | **1100** | **0** | **1** | **0** |
| **1011** | **1101** | **0** | **1** | **0** |
| **1011** | **1110** | **0** | **1** | **0** |
| **1011** | **1111** | **0** | **1** | **0** |
| **1100** | **0000** | **1** | **0** | **0** |
| **1100** | **0001** | **1** | **0** | **0** |
| **1100** | **0010** | **1** | **0** | **0** |
| **1100** | **0011** | **1** | **0** | **0** |
| **1100** | **0100** | **1** | **0** | **0** |
| **1100** | **0101** | **1** | **0** | **0** |
| **1100** | **0110** | **1** | **0** | **0** |
| **1100** | **0111** | **1** | **0** | **0** |
| **1100** | **1000** | **1** | **0** | **0** |
| **1100** | **1001** | **1** | **0** | **0** |
| **1100** | **1010** | **1** | **0** | **0** |
| **1100** | **1011** | **1** | **0** | **0** |
| **1100** | **1100** | **0** | **0** | **1** |
| **1100** | **1101** | **0** | **1** | **0** |
| **1100** | **1110** | **0** | **1** | **0** |
| **1100** | **1111** | **0** | **1** | **0** |
| **1101** | **0000** | **1** | **0** | **0** |
| **1101** | **0001** | **1** | **0** | **0** |
| **1101** | **0010** | **1** | **0** | **0** |
| **1101** | **0011** | **1** | **0** | **0** |
| **1101** | **0100** | **1** | **0** | **0** |
| **1101** | **0101** | **1** | **0** | **0** |
| **1101** | **0110** | **1** | **0** | **0** |
| **1101** | **0111** | **1** | **0** | **0** |
| **1101** | **1000** | **1** | **0** | **0** |
| **1101** | **1001** | **1** | **0** | **0** |
| **1101** | **1010** | **1** | **0** | **0** |
| **1101** | **1011** | **1** | **0** | **0** |
| **1101** | **1100** | **1** | **0** | **0** |
| **1101** | **1101** | **0** | **0** | **1** |
| **1101** | **1110** | **0** | **1** | **0** |
| **1101** | **1111** | **0** | **1** | **0** |
| **1110** | **0000** | **1** | **0** | **0** |
| **1110** | **0001** | **1** | **0** | **0** |
| **1110** | **0010** | **1** | **0** | **0** |
| **1110** | **0011** | **1** | **0** | **0** |
| **1110** | **0100** | **1** | **0** | **0** |
| **1110** | **0101** | **1** | **0** | **0** |
| **1110** | **0110** | **1** | **0** | **0** |
| **1110** | **0111** | **1** | **0** | **0** |
| **1110** | **1000** | **1** | **0** | **0** |
| **1110** | **1001** | **1** | **0** | **0** |
| **1110** | **1010** | **1** | **0** | **0** |
| **1110** | **1011** | **1** | **0** | **0** |
| **1110** | **1100** | **1** | **0** | **0** |
| **1110** | **1101** | **1** | **0** | **0** |
| **1110** | **1110** | **0** | **0** | **1** |
| **1110** | **1111** | **0** | **1** | **0** |
| **1111** | **0000** | **1** | **0** | **0** |
| **1111** | **0001** | **1** | **0** | **0** |
| **1111** | **0010** | **1** | **0** | **0** |
| **1111** | **0011** | **1** | **0** | **0** |
| **1111** | **0100** | **1** | **0** | **0** |
| **1111** | **0101** | **1** | **0** | **0** |
| **1111** | **0110** | **1** | **0** | **0** |
| **1111** | **0111** | **1** | **0** | **0** |
| **1111** | **1000** | **1** | **0** | **0** |
| **1111** | **1001** | **1** | **0** | **0** |
| **1111** | **1010** | **1** | **0** | **0** |
| **1111** | **1011** | **1** | **0** | **0** |
| **1111** | **1100** | **1** | **0** | **0** |
| **1111** | **1101** | **1** | **0** | **0** |
| **1111** | **1110** | **1** | **0** | **0** |
| **1111** | **1111** | **0** | **0** | **1** |

**TRUTH TABLE FOR S COMPARATOR (SIGNED)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C** | **D** | **GCD** | **GDC** | **E** |
| **0000** | **0000** | **0** | **0** | **1** |
| **0000** | **0001** | **0** | **1** | **0** |
| **0000** | **0010** | **0** | **1** | **0** |
| **0000** | **0011** | **0** | **1** | **0** |
| **0000** | **0100** | **0** | **1** | **0** |
| **0000** | **0101** | **0** | **1** | **0** |
| **0000** | **0110** | **0** | **1** | **0** |
| **0000** | **0111** | **0** | **1** | **0** |
| **0000** | **1000** | **1** | **0** | **0** |
| **0000** | **1001** | **1** | **0** | **0** |
| **0000** | **1010** | **1** | **0** | **0** |
| **0000** | **1011** | **1** | **0** | **0** |
| **0000** | **1100** | **1** | **0** | **0** |
| **0000** | **1101** | **1** | **0** | **0** |
| **0000** | **1110** | **1** | **0** | **0** |
| **0000** | **1111** | **1** | **0** | **0** |
| **0001** | **0000** | **1** | **0** | **0** |
| **0001** | **0001** | **0** | **0** | **1** |
| **0001** | **0010** | **0** | **1** | **0** |
| **0001** | **0011** | **0** | **1** | **0** |
| **0001** | **0100** | **0** | **1** | **0** |
| **0001** | **0101** | **0** | **1** | **0** |
| **0001** | **0110** | **0** | **1** | **0** |
| **0001** | **0111** | **0** | **1** | **0** |
| **0001** | **1000** | **1** | **0** | **0** |
| **0001** | **1001** | **1** | **0** | **0** |
| **0001** | **1010** | **1** | **0** | **0** |
| **0001** | **1011** | **1** | **0** | **0** |
| **0001** | **1100** | **1** | **0** | **0** |
| **0001** | **1101** | **1** | **0** | **0** |
| **0001** | **1110** | **1** | **0** | **0** |
| **0001** | **1111** | **1** | **0** | **0** |
| **0010** | **0000** | **1** | **0** | **0** |
| **0010** | **0001** | **1** | **0** | **0** |
| **0010** | **0010** | **0** | **0** | **1** |
| **0010** | **0011** | **0** | **1** | **0** |
| **0010** | **0100** | **0** | **1** | **0** |
| **0010** | **0101** | **0** | **1** | **0** |
| **0010** | **0110** | **0** | **1** | **0** |
| **0010** | **0111** | **0** | **1** | **0** |
| **0010** | **1000** | **1** | **0** | **0** |
| **0010** | **1001** | **1** | **0** | **0** |
| **0010** | **1010** | **1** | **0** | **0** |
| **0010** | **1011** | **1** | **0** | **0** |
| **0010** | **1100** | **1** | **0** | **0** |
| **0010** | **1101** | **1** | **0** | **0** |
| **0010** | **1110** | **1** | **0** | **0** |
| **0010** | **1111** | **1** | **0** | **0** |
| **0011** | **0000** | **1** | **0** | **0** |
| **0011** | **0001** | **1** | **0** | **0** |
| **0011** | **0010** | **1** | **0** | **0** |
| **0011** | **0011** | **0** | **0** | **1** |
| **0011** | **0100** | **0** | **1** | **0** |
| **0011** | **0101** | **0** | **1** | **0** |
| **0011** | **0110** | **0** | **1** | **0** |
| **0011** | **0111** | **0** | **1** | **0** |
| **0011** | **1000** | **1** | **0** | **0** |
| **0011** | **1001** | **1** | **0** | **0** |
| **0011** | **1010** | **1** | **0** | **0** |
| **0011** | **1011** | **1** | **0** | **0** |
| **0011** | **1100** | **1** | **0** | **0** |
| **0011** | **1101** | **1** | **0** | **0** |
| **0011** | **1110** | **1** | **0** | **0** |
| **0011** | **1111** | **1** | **0** | **0** |
| **0100** | **0000** | **1** | **0** | **0** |
| **0100** | **0001** | **1** | **0** | **0** |
| **0100** | **0010** | **1** | **0** | **0** |
| **0100** | **0011** | **1** | **0** | **0** |
| **0100** | **0100** | **0** | **0** | **1** |
| **0100** | **0101** | **0** | **1** | **0** |
| **0100** | **0110** | **0** | **1** | **0** |
| **0100** | **0111** | **0** | **1** | **0** |
| **0100** | **1000** | **1** | **0** | **0** |
| **0100** | **1001** | **1** | **0** | **0** |
| **0100** | **1010** | **1** | **0** | **0** |
| **0100** | **1011** | **1** | **0** | **0** |
| **0100** | **1100** | **1** | **0** | **0** |
| **0100** | **1101** | **1** | **0** | **0** |
| **0100** | **1110** | **1** | **0** | **0** |
| **0100** | **1111** | **1** | **0** | **0** |
| **0101** | **0000** | **1** | **0** | **0** |
| **0101** | **0001** | **1** | **0** | **0** |
| **0101** | **0010** | **1** | **0** | **0** |
| **0101** | **0011** | **1** | **0** | **0** |
| **0101** | **0100** | **1** | **0** | **0** |
| **0101** | **0101** | **0** | **0** | **1** |
| **0101** | **0110** | **0** | **1** | **0** |
| **0101** | **0111** | **0** | **1** | **0** |
| **0101** | **1000** | **1** | **0** | **0** |
| **0101** | **1001** | **1** | **0** | **0** |
| **0101** | **1010** | **1** | **0** | **0** |
| **0101** | **1011** | **1** | **0** | **0** |
| **0101** | **1100** | **1** | **0** | **0** |
| **0101** | **1101** | **1** | **0** | **0** |
| **0101** | **1110** | **0** | **0** | **0** |
| **0101** | **1111** | **1** | **0** | **0** |
| **0110** | **0000** | **1** | **0** | **0** |
| **0110** | **0001** | **1** | **0** | **0** |
| **0110** | **0010** | **1** | **0** | **0** |
| **0110** | **0011** | **1** | **0** | **0** |
| **0110** | **0100** | **1** | **0** | **0** |
| **0110** | **0101** | **1** | **0** | **0** |
| **0110** | **0110** | **0** | **0** | **1** |
| **0110** | **0111** | **0** | **1** | **0** |
| **0110** | **1000** | **1** | **0** | **0** |
| **0110** | **1001** | **1** | **0** | **0** |
| **0110** | **1010** | **1** | **0** | **0** |
| **0110** | **1011** | **1** | **0** | **0** |
| **0110** | **1100** | **1** | **0** | **0** |
| **0110** | **1101** | **1** | **0** | **0** |
| **0110** | **1110** | **1** | **0** | **0** |
| **0110** | **1111** | **1** | **0** | **0** |
| **0111** | **0000** | **1** | **0** | **0** |
| **0111** | **0001** | **1** | **0** | **0** |
| **0111** | **0010** | **1** | **0** | **0** |
| **0111** | **0011** | **1** | **0** | **0** |
| **0111** | **0100** | **1** | **0** | **0** |
| **0111** | **0101** | **1** | **0** | **0** |
| **0111** | **0110** | **1** | **0** | **0** |
| **0111** | **0111** | **0** | **0** | **1** |
| **0111** | **1000** | **1** | **0** | **0** |
| **0111** | **1001** | **1** | **0** | **0** |
| **0111** | **1010** | **1** | **0** | **0** |
| **0111** | **1011** | **1** | **0** | **0** |
| **0111** | **1100** | **1** | **0** | **0** |
| **0111** | **1101** | **1** | **0** | **0** |
| **0111** | **1110** | **1** | **0** | **0** |
| **0111** | **1111** | **1** | **0** | **0** |
| **1000** | **0000** | **0** | **1** | **0** |
| **1000** | **0001** | **0** | **1** | **0** |
| **1000** | **0010** | **0** | **1** | **0** |
| **1000** | **0011** | **0** | **1** | **0** |
| **1000** | **0100** | **0** | **1** | **0** |
| **1000** | **0101** | **0** | **1** | **0** |
| **1000** | **0110** | **0** | **1** | **0** |
| **1000** | **0111** | **0** | **1** | **0** |
| **1000** | **1000** | **0** | **0** | **1** |
| **1000** | **1001** | **0** | **1** | **0** |
| **1000** | **1010** | **0** | **1** | **0** |
| **1000** | **1011** | **0** | **1** | **0** |
| **1000** | **1100** | **0** | **1** | **0** |
| **1000** | **1101** | **0** | **1** | **0** |
| **1000** | **1110** | **0** | **1** | **0** |
| **1000** | **1111** | **0** | **1** | **0** |
| **1001** | **0000** | **0** | **1** | **0** |
| **1001** | **0001** | **0** | **1** | **0** |
| **1001** | **0010** | **0** | **1** | **0** |
| **1001** | **0011** | **0** | **1** | **0** |
| **1001** | **0100** | **0** | **1** | **0** |
| **1001** | **0101** | **0** | **1** | **0** |
| **1001** | **0110** | **0** | **1** | **0** |
| **1001** | **0111** | **0** | **1** | **0** |
| **1001** | **1000** | **1** | **0** | **0** |
| **1001** | **1001** | **0** | **0** | **1** |
| **1001** | **1010** | **0** | **1** | **0** |
| **1001** | **1011** | **0** | **1** | **0** |
| **1001** | **1100** | **0** | **1** | **0** |
| **1001** | **1101** | **0** | **1** | **0** |
| **1001** | **1110** | **0** | **1** | **0** |
| **1001** | **1111** | **0** | **1** | **0** |
| **1010** | **0000** | **0** | **1** | **0** |
| **1010** | **0001** | **0** | **1** | **0** |
| **1010** | **0010** | **0** | **1** | **0** |
| **1010** | **0011** | **0** | **1** | **0** |
| **1010** | **0100** | **0** | **1** | **0** |
| **1010** | **0101** | **0** | **1** | **0** |
| **1010** | **0110** | **0** | **1** | **0** |
| **1010** | **0111** | **0** | **1** | **0** |
| **1010** | **1000** | **1** | **0** | **0** |
| **1010** | **1001** | **1** | **0** | **0** |
| **1010** | **1010** | **0** | **0** | **1** |
| **1010** | **1011** | **0** | **1** | **0** |
| **1010** | **1100** | **0** | **1** | **0** |
| **1010** | **1101** | **0** | **1** | **0** |
| **1010** | **1110** | **0** | **1** | **0** |
| **1010** | **1111** | **0** | **1** | **0** |
| **1011** | **0000** | **0** | **1** | **0** |
| **1011** | **0001** | **0** | **1** | **0** |
| **1011** | **0010** | **0** | **1** | **0** |
| **1011** | **0011** | **0** | **1** | **0** |
| **1011** | **0100** | **0** | **1** | **0** |
| **1011** | **0101** | **0** | **1** | **0** |
| **1011** | **0110** | **0** | **1** | **0** |
| **1011** | **0111** | **0** | **1** | **0** |
| **1011** | **1000** | **1** | **0** | **0** |
| **1011** | **1001** | **1** | **0** | **0** |
| **1011** | **1010** | **1** | **0** | **0** |
| **1011** | **1011** | **0** | **0** | **1** |
| **1011** | **1100** | **0** | **1** | **0** |
| **1011** | **1101** | **0** | **1** | **0** |
| **1011** | **1110** | **0** | **1** | **0** |
| **1011** | **1111** | **0** | **1** | **0** |
| **1100** | **0000** | **0** | **1** | **0** |
| **1100** | **0001** | **0** | **1** | **0** |
| **1100** | **0010** | **0** | **1** | **0** |
| **1100** | **0011** | **0** | **1** | **0** |
| **1100** | **0100** | **0** | **1** | **0** |
| **1100** | **0101** | **0** | **1** | **0** |
| **1100** | **0110** | **0** | **1** | **0** |
| **1100** | **0111** | **0** | **1** | **0** |
| **1100** | **1000** | **1** | **0** | **0** |
| **1100** | **1001** | **1** | **0** | **0** |
| **1100** | **1010** | **1** | **0** | **0** |
| **1100** | **1011** | **1** | **0** | **0** |
| **1100** | **1100** | **0** | **0** | **1** |
| **1100** | **1101** | **0** | **1** | **0** |
| **1100** | **1110** | **0** | **1** | **0** |
| **1100** | **1111** | **0** | **1** | **0** |
| **1101** | **0000** | **0** | **1** | **0** |
| **1101** | **0001** | **0** | **1** | **0** |
| **1101** | **0010** | **0** | **1** | **0** |
| **1101** | **0011** | **0** | **1** | **0** |
| **1101** | **0100** | **0** | **1** | **0** |
| **1101** | **0101** | **0** | **1** | **0** |
| **1101** | **0110** | **0** | **1** | **0** |
| **1101** | **0111** | **0** | **1** | **0** |
| **1101** | **1000** | **1** | **0** | **0** |
| **1101** | **1001** | **1** | **0** | **0** |
| **1101** | **1010** | **1** | **0** | **0** |
| **1101** | **1011** | **1** | **0** | **0** |
| **1101** | **1100** | **1** | **0** | **0** |
| **1101** | **1101** | **0** | **0** | **1** |
| **1101** | **1110** | **0** | **1** | **0** |
| **1101** | **1111** | **0** | **1** | **0** |
| **1110** | **0000** | **0** | **1** | **0** |
| **1110** | **0001** | **0** | **1** | **0** |
| **1110** | **0010** | **0** | **1** | **0** |
| **1110** | **0011** | **0** | **1** | **0** |
| **1110** | **0100** | **0** | **1** | **0** |
| **1110** | **0101** | **0** | **1** | **0** |
| **1110** | **0110** | **0** | **1** | **0** |
| **1110** | **0111** | **0** | **1** | **0** |
| **1110** | **1000** | **1** | **0** | **0** |
| **1110** | **1001** | **1** | **0** | **0** |
| **1110** | **1010** | **1** | **0** | **0** |
| **1110** | **1011** | **1** | **0** | **0** |
| **1110** | **1100** | **1** | **0** | **0** |
| **1110** | **1101** | **1** | **0** | **0** |
| **1110** | **1110** | **0** | **0** | **1** |
| **1110** | **1111** | **0** | **1** | **0** |
| **1111** | **0000** | **0** | **1** | **0** |
| **1111** | **0001** | **0** | **1** | **0** |
| **1111** | **0010** | **0** | **1** | **0** |
| **1111** | **0011** | **0** | **1** | **0** |
| **1111** | **0100** | **0** | **1** | **0** |
| **1111** | **0101** | **0** | **1** | **0** |
| **1111** | **0110** | **0** | **1** | **0** |
| **1111** | **0111** | **0** | **1** | **0** |
| **1111** | **1000** | **1** | **0** | **0** |
| **1111** | **1001** | **1** | **0** | **0** |
| **1111** | **1010** | **1** | **0** | **0** |
| **1111** | **1011** | **1** | **0** | **0** |
| **1111** | **1100** | **1** | **0** | **0** |
| **1111** | **1101** | **1** | **0** | **0** |
| **1111** | **1110** | **1** | **0** | **0** |
| **1111** | **1111** | **0** | **0** | **1** |

1. **Problems encountered, errors and warnings resolved**

Explain what problems you encountered while writing your code.

**I have not encountered with any problems while writing my code.**

Explain what synthesis errors and warnings you observed.

**I have not encountered with any synthesis errors and I have not observed any warnings.**

Explain what problems you had to solve (or could not) on your board even if your code could be synthesized successfully.

**Since it is not necessary, I have not used the board in the lab. So, I haven’t got any problems I had to solve on my board .**

1. **Conclusion**

Provide a 1 paragraph summary of the lab and explain what you learned from this lab.

**In the lab-2, firstly, I have created project which has a name “MComparator”. Then, I have created vhdl module which has a name “MComparator.vhd”. After that, I have created vhdl test bench which has a name “MComparator.sim”. After the creations, I wrote the simulation code to MComparator.sim and my implementation of code to MComparator.vhd. I have applied same general procedure to the project which has a name “SComparator”. I have tried to give 3 different pairs of inputs to the simulation codes which are “SComparator.sim and MComparator.sim”. The logic of the implementation of MComparator.vhd and SComparator.vhd comes from thinking all of the possible cases for the bits of A,B,C, and D where (A>B or A=B or A<B) and (C>D or C=D or C<D). I have run the simulation codes and observed the output values. In the timing diagram, when A is equal to B; I have observed that E is 1, TAB is 0, and TBA is 0. When A is less than B; I have observed that E is 0, TAB is 0, and TBA is 1. When A is greater than B; I have observed that E is 0, TAB is 1, and TBA is 0. When C is equal to D; I have observed that E is 1, TCD is 0, and TDC is zero. When C is greater than D; I have observed that E is 0, TCD is 1, and TDC is 0. When C is smaller than D; I have observed that E is 0, TDC is 1, and TCD is 0.**

**I have learned how to implement the code of the 4-bit magnitude comparator for unsigned binary numbers. I have learned how to implement the VHDL code of the 4-bit comparator for signed binary numbers. I have learned how to implement the VHDL code of the 4-bit magnitude comparator for unsigned binary numbers. I have learned how to draw and design an equality comparator circuit for 4-bit binary numbers. I have also learned how to draw and design a 4-bit magnitude comparator circuit for unsigned binary numbers. In addition to these, I have learned how to observe different output values for different pairs of inputs.**

References

1. Please cite any resource (web site, book, youtube video) you used for this lab.

**Youtube video:** [**https://www.youtube.com/watch?v=s-WLncsM-oE**](https://www.youtube.com/watch?v=s-WLncsM-oE)

**Web site:** [**https://stackoverflow.com/questions/49793354/incrementing-a-std-logic-vector-in-vhdl**](https://stackoverflow.com/questions/49793354/incrementing-a-std-logic-vector-in-vhdl)

**Youtube video:** [**https://www.youtube.com/watch?v=sQGlD3NRBuw**](https://www.youtube.com/watch?v=sQGlD3NRBuw)

**MComparator Source Code**

----------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date:    13:16:25 03/18/2021

-- Design Name:

-- Module Name:    MComparator - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

----------------------------------------------------------------------------------

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC\_STD.ALL;

-- Uncomment the following library declaration if instantiating

-- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity MComparator is

    Port ( A : in  STD\_LOGIC\_VECTOR (3 downto 0);

           B : in  STD\_LOGIC\_VECTOR (3 downto 0);

           GAB : out  STD\_LOGIC;

           GBA : out  STD\_LOGIC;

           E : out  STD\_LOGIC);

end MComparator;

architecture Behavioral of MComparator is

signal e0:std\_logic;

signal e1:std\_logic;

signal e2:std\_logic;

signal e3:std\_logic;

signal NA0: std\_logic;

signal NA1: std\_logic;

signal NA2: std\_logic;

signal NA3: std\_logic;

signal NB0: std\_logic;

signal NB1: std\_logic;

signal NB2: std\_logic;

signal NB3: std\_logic;

signal E123:std\_logic;

signal E23: std\_logic;

begin

NA0<= not A(0);

NA1<= not A(1);

NA2<= not A(2);

NA3<= not A(3);

NB0<= not B(0);

NB1<= not B(1);

NB2<= not B(2);

NB3<= not B(3);

e0<=(not (A(0) xor B(0)));

e1<=(not (A(1) xor B(1)));

e2<=(not (A(2) xor B(2)));

e3<=(not (A(3) xor B(3)));

E123<= e3 and e2 and e1;

E23<= e3 and e2;

E <= e0 and e1 and e2 and e3;

GAB <= (A(3) and (NB3)) or (e3 and A(2) and (NB2)) or (E23 and A(1) and (NB1))

or (E123 and A(0) and (NB0));

GBA <= ((NA3) and B(3)) or (e3 and (NA2) and B(2)) or (E23 and (NA1) and B(1))

or (E123 and (NA0) and B(0));

end Behavioral;

**MComparatorSim Code**

--------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date:   15:12:43 03/18/2021

-- Design Name:

-- Module Name:   C:/Test/MComparator/MComparatorSim.vhd

-- Project Name:  MComparator

-- Target Device:

-- Tool versions:

-- Description:

--

-- VHDL Test Bench Created by ISE for module: MComparator

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

-- Notes:

-- This testbench has been automatically generated using types std\_logic and

-- std\_logic\_vector for the ports of the unit under test.  Xilinx recommends

-- that these types always be used for the top-level I/O of a design in order

-- to guarantee that the testbench will bind correctly to the post-implementation

-- simulation model.

--------------------------------------------------------------------------------

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--USE ieee.numeric\_std.ALL;

ENTITY MComparatorSim IS

END MComparatorSim;

ARCHITECTURE behavior OF MComparatorSim IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT MComparator

    PORT(

         A : IN  std\_logic\_vector(3 downto 0);

         B : IN  std\_logic\_vector(3 downto 0);

         GAB : OUT  std\_logic;

         GBA : OUT  std\_logic;

         E : OUT  std\_logic

        );

    END COMPONENT;

   --Inputs

   signal A : std\_logic\_vector(3 downto 0) := (others => '0');

   signal B : std\_logic\_vector(3 downto 0) := (others => '0');

  --Outputs

   signal GAB : std\_logic;

   signal GBA : std\_logic;

   signal E : std\_logic;

   -- No clocks detected in port list. Replace <clock> below with

   -- appropriate port name

BEGIN

-- Instantiate the Unit Under Test (UUT)

   uut: MComparator PORT MAP (

          A => A,

          B => B,

          GAB => GAB,

          GBA => GBA,

          E => E

        );

   -- Stimulus process

   stim\_proc: process

   begin

A<="0010";

B<="1001";

      wait for 20 ns;

A<="1001";

       B<="0010";

      wait for 20 ns;

      A<="0101";

      B<="0101";

      -- insert stimulus here

      wait;

   end process;

END;

**SComparator Source Code**

-- Company:

-- Engineer:

--

-- Create Date:    13:16:25 03/18/2021

-- Design Name:

-- Module Name:    SComparator - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

----------------------------------------------------------------------------------

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC\_STD.ALL;

-- Uncomment the following library declaration if instantiating

-- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity SComparator is

    Port ( C : in  STD\_LOGIC\_VECTOR (3 downto 0);

           D : in  STD\_LOGIC\_VECTOR (3 downto 0);

           GCD : out  STD\_LOGIC;

           GDC : out  STD\_LOGIC;

           E : out  STD\_LOGIC);

end SComparator;

architecture Behavioral of SComparator is

signal e0:std\_logic;

signal e1:std\_logic;

signal e2:std\_logic;

signal e3:std\_logic;

signal NC0: std\_logic;

signal NC1: std\_logic;

signal NC2: std\_logic;

signal NC3: std\_logic;

signal ND0: std\_logic;

signal ND1: std\_logic;

signal ND2: std\_logic;

signal ND3: std\_logic;

signal E123:std\_logic;

signal E23: std\_logic;

begin

NC0<= not C(0);

NC1<= not C(1);

NC2<= not C(2);

NC3<= not C(3);

ND0<= not D(0);

ND1<= not D(1);

ND2<= not D(2);

ND3<= not D(3);

e0<=(not (C(0) xor D(0)));

e1<=(not (C(1) xor D(1)));

e2<=(not (C(2) xor D(2)));

e3<=(not ((NC3) xor (ND3)));

E123<= e3 and e2 and e1;

E23<= e3 and e2;

E <= e0 and e1 and e2 and e3;

GCD <= ((NC3) and (D(3))) or (e3 and C(2) and (ND2)) or (E23 and C(1) and (ND1))

or (E123 and C(0) and (ND0));

GDC <= ((C(3)) and (ND3)) or (e3 and (NC2) and D(2)) or (E23 and (NC1) and D(1))

or (E123 and (NC0) and D(0));

end Behavioral;

**SComparatorSim Code**

--------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date:   17:31:20 03/18/2021

-- Design Name:

-- Module Name:   C:/Test/SComparator/SComparatorSim.vhd

-- Project Name:  SComparator

-- Target Device:

-- Tool versions:

-- Description:

--

-- VHDL Test Bench Created by ISE for module: SComparator

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

-- Notes:

-- This testbench has been automatically generated using types std\_logic and

-- std\_logic\_vector for the ports of the unit under test.  Xilinx recommends

-- that these types always be used for the top-level I/O of a design in order

-- to guarantee that the testbench will bind correctly to the post-implementation

-- simulation model.

--------------------------------------------------------------------------------

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--USE ieee.numeric\_std.ALL;

ENTITY SComparatorSim IS

END SComparatorSim;

ARCHITECTURE behavior OF SComparatorSim IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT SComparator

    PORT(

         C : IN  std\_logic\_vector(3 downto 0);

         D : IN  std\_logic\_vector(3 downto 0);

         GCD : OUT  std\_logic;

         GDC : OUT  std\_logic;

         E : OUT  std\_logic

        );

    END COMPONENT;

   --Inputs

   signal C : std\_logic\_vector(3 downto 0) := (others => '0');

   signal D : std\_logic\_vector(3 downto 0) := (others => '0');

  --Outputs

   signal GCD : std\_logic;

   signal GDC : std\_logic;

   signal E : std\_logic;

   -- No clocks detected in port list. Replace <clock> below with

   -- appropriate port name

BEGIN

-- Instantiate the Unit Under Test (UUT)

   uut: SComparator PORT MAP (

          C => C,

          D => D,

          GCD => GCD,

          GDC => GDC,

          E => E

        );

   -- Stimulus process

    stim\_proc: process

   begin

C<="0010";

D<="1001";

      wait for 20 ns;

C<="1001";

      D<="0010";

      wait for 20 ns;

      C<="1111";

      D<="0000";

      wait;

   end process;

END;