Faculty of Engineering and Technology

Electrical and Computer Engineering Department

ENCS3310

ADVANCED DIGITAL SYSTEMS DESIGN Project#1

Prepared by:

Ahmad karmi 1220003

Instructor: Elias Khalil

Section: 1

Date: 12/24/2024

Abstract

This project involves the design of a comparator for 6-bit signed and unsigned numbers. The comparator determines if two numbers are equal, greater, or smaller, based on a selection input. It is built using basic logic gates and includes synchronous operation with input and output registers. The design is tested for all input combinations, and error detection is implemented. Key findings and potential improvements are highlighted.

Table of Contents

Table of Figures

1 Theory

A comparator is a fundamental digital circuit used to compare two binary numbers and determine their relationship. In this project, we design a comparator for signed and unsigned 6-bit numbers, considering the following operations: equality, greater-than, and less-than.

1.1 Signed vs. Unsigned Numbers

1.1.1 Signed Numbers

Represented using 2's complement, where the most significant bit (MSB) indicates the sign. A 0 in the MSB denotes a positive number, while a 1 represents a negative number. Comparisons account for the MSB to ensure correct ordering of values.

1.1.2 Unsigned Numbers

All bits represent the magnitude of the value, with no distinction between positive and negative, as the numbers are always non-negative.

1.2 Structural Design

The comparator uses basic logic gates like AND, OR, and XOR to generate outputs (Equal, Greater, Smaller) for both signed and unsigned numbers.

1.3 Selection Mechanism

A one bit selector variable named (S) , used to determine the mode of the comparison S = 0 for unsigned comparison and S = 1 for signed comparison.

1.4 Synchronization and Timing

The circuit is made synchronous by adding input and output registers, which allow it to function with a defined clock signal. The maximum latency is calculated to identify the highest clock frequency the system can support.

1.5 Error Detection

A mechanism is integrated into the design to detect any errors in the system. If an error occurs, it will be logged and displayed in the console for quick identification.

2 Procedures

A 2-bit comparator serves as the foundational building block to construct a 6-bit unsigned comparator, while a 6-bit signed comparator integrates additional logic to handle the sign bit. The design includes a 1-bit comparator dedicated to evaluating the most significant bit (MSB), which determines the sign of the numbers and establishes whether the values are positive or negative. If the MSBs are equal, the comparison shifts to the remaining 5 bits, which are concatenated with a zero and processed through the 6-bit unsigned comparator. A multiplexer (MUX) is employed to switch between signed and unsigned comparison modes, controlled by a selector signal; when the selector is set to 0, the design performs an unsigned comparison, and when set to 1, it executes a signed comparison. To address potential glitches caused by delays in the propagation of signals through basic gates, all inputs and outputs are stabilized using D Flip-Flops. Alongside the structural comparator, a behavioral comparator is implemented to validate the results, and both outputs are routed to an analyzer module. This module compares the results, identifies any discrepancies, and outputs an error message if a mismatch occurs, ensuring reliability and accuracy in the design.

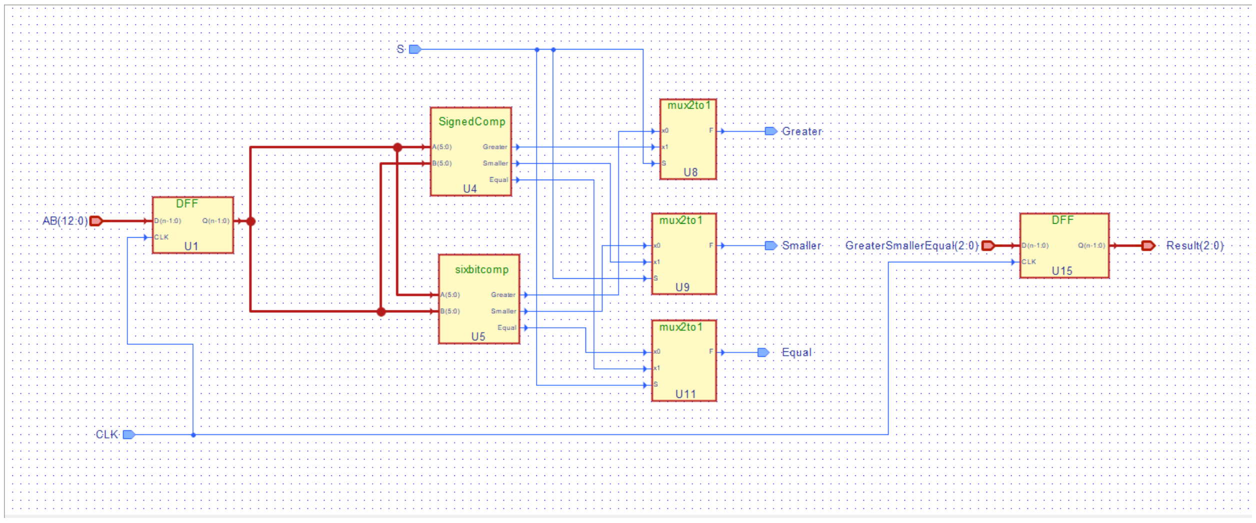


Figure 1: comparator circuit

3 Simulation Results

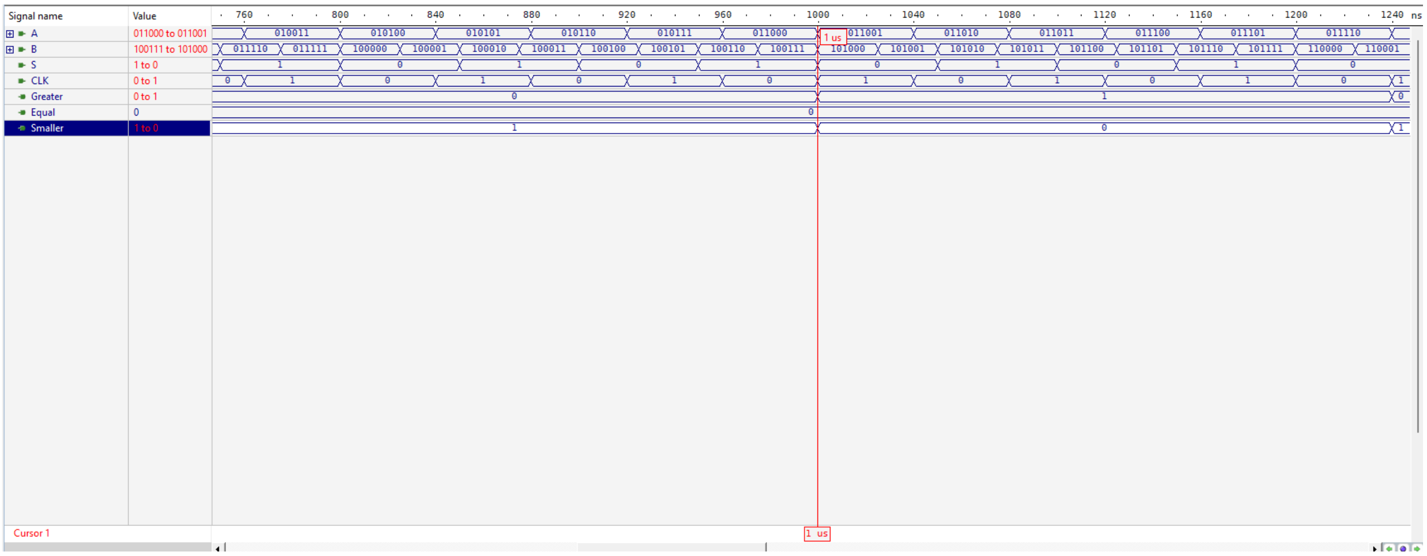
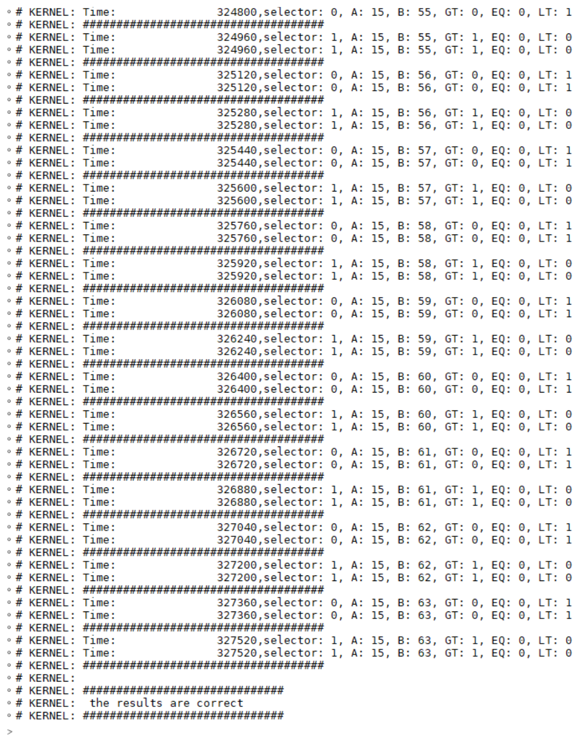


Figure 2: wave form

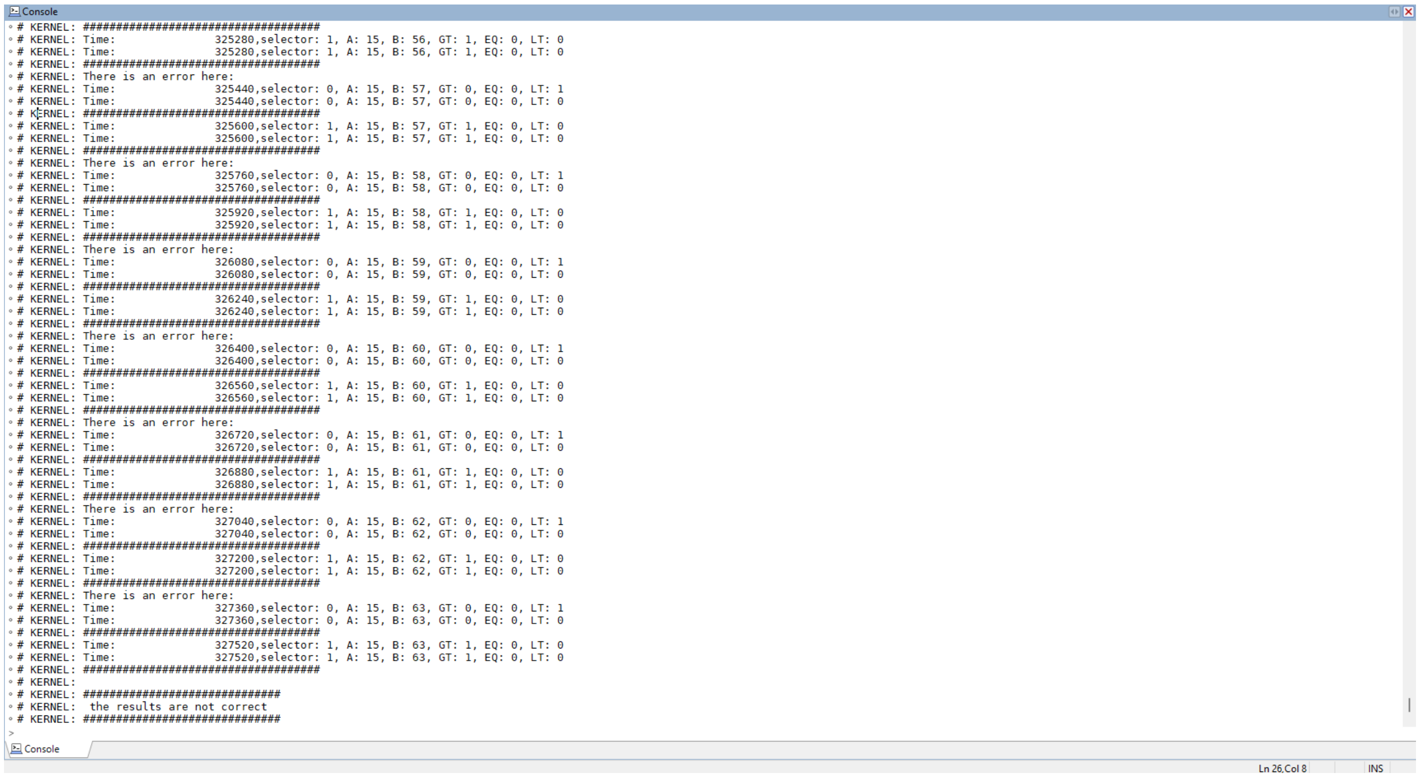
This is what print’s when the results are correct



```
◦ # KERNEL: Time:              324800,selector: 0, A: 15, B: 55, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              324960,selector: 1, A: 15, B: 55, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: Time:              324960,selector: 1, A: 15, B: 55, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              325120,selector: 0, A: 15, B: 56, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: Time:              325120,selector: 0, A: 15, B: 56, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              325280,selector: 1, A: 15, B: 56, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: Time:              325280,selector: 1, A: 15, B: 56, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              325440,selector: 0, A: 15, B: 57, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: Time:              325440,selector: 0, A: 15, B: 57, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              325600,selector: 1, A: 15, B: 57, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: Time:              325600,selector: 1, A: 15, B: 57, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              325760,selector: 0, A: 15, B: 58, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: Time:              325760,selector: 0, A: 15, B: 58, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              325920,selector: 1, A: 15, B: 58, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: Time:              325920,selector: 1, A: 15, B: 58, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              326080,selector: 0, A: 15, B: 59, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: Time:              326080,selector: 0, A: 15, B: 59, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              326240,selector: 1, A: 15, B: 59, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: Time:              326240,selector: 1, A: 15, B: 59, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              326400,selector: 0, A: 15, B: 60, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: Time:              326400,selector: 0, A: 15, B: 60, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              326560,selector: 1, A: 15, B: 60, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: Time:              326560,selector: 1, A: 15, B: 60, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              326720,selector: 0, A: 15, B: 61, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: Time:              326720,selector: 0, A: 15, B: 61, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              326880,selector: 1, A: 15, B: 61, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: Time:              326880,selector: 1, A: 15, B: 61, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              327040,selector: 0, A: 15, B: 62, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: Time:              327040,selector: 0, A: 15, B: 62, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              327200,selector: 1, A: 15, B: 62, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: Time:              327200,selector: 1, A: 15, B: 62, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              327360,selector: 0, A: 15, B: 63, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: Time:              327360,selector: 0, A: 15, B: 63, GT: 0, EQ: 0, LT: 1
◦ # KERNEL: ####################################
◦ # KERNEL: Time:              327520,selector: 1, A: 15, B: 63, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: Time:              327520,selector: 1, A: 15, B: 63, GT: 1, EQ: 0, LT: 0
◦ # KERNEL: ####################################
◦ # KERNEL:
◦ # KERNEL: ##############################
◦ # KERNEL:  the results are correct
◦ # KERNEL: ##############################
>
```

Figure 3: Analyzer

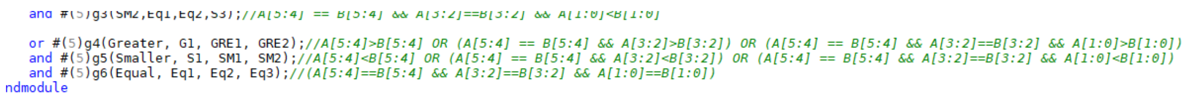
This is what print's when the results are not correct



```
# KERNEL: ####################################
# KERNEL: Time:                   325280,selector: 1, A: 15, B: 56, GT: 1, EQ: 0, LT: 0
# KERNEL: Time:                   325280,selector: 1, A: 15, B: 56, GT: 1, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: There is an error here:
# KERNEL: Time:                   325440,selector: 0, A: 15, B: 57, GT: 0, EQ: 0, LT: 1
# KERNEL: Time:                   325440,selector: 0, A: 15, B: 57, GT: 0, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: Time:                   325600,selector: 1, A: 15, B: 57, GT: 1, EQ: 0, LT: 0
# KERNEL: Time:                   325600,selector: 1, A: 15, B: 57, GT: 1, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: There is an error here:
# KERNEL: Time:                   325760,selector: 0, A: 15, B: 58, GT: 0, EQ: 0, LT: 1
# KERNEL: Time:                   325760,selector: 0, A: 15, B: 58, GT: 0, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: Time:                   325920,selector: 1, A: 15, B: 58, GT: 1, EQ: 0, LT: 0
# KERNEL: Time:                   325920,selector: 1, A: 15, B: 58, GT: 1, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: There is an error here:
# KERNEL: Time:                   326080,selector: 0, A: 15, B: 59, GT: 0, EQ: 0, LT: 1
# KERNEL: Time:                   326080,selector: 0, A: 15, B: 59, GT: 0, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: Time:                   326240,selector: 1, A: 15, B: 59, GT: 1, EQ: 0, LT: 0
# KERNEL: Time:                   326240,selector: 1, A: 15, B: 59, GT: 1, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: There is an error here:
# KERNEL: Time:                   326400,selector: 0, A: 15, B: 60, GT: 0, EQ: 0, LT: 1
# KERNEL: Time:                   326400,selector: 0, A: 15, B: 60, GT: 0, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: Time:                   326560,selector: 1, A: 15, B: 60, GT: 1, EQ: 0, LT: 0
# KERNEL: Time:                   326560,selector: 1, A: 15, B: 60, GT: 1, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: There is an error here:
# KERNEL: Time:                   326720,selector: 0, A: 15, B: 61, GT: 0, EQ: 0, LT: 1
# KERNEL: Time:                   326720,selector: 0, A: 15, B: 61, GT: 0, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: Time:                   326880,selector: 1, A: 15, B: 61, GT: 1, EQ: 0, LT: 0
# KERNEL: Time:                   326880,selector: 1, A: 15, B: 61, GT: 1, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: There is an error here:
# KERNEL: Time:                   327040,selector: 0, A: 15, B: 62, GT: 0, EQ: 0, LT: 1
# KERNEL: Time:                   327040,selector: 0, A: 15, B: 62, GT: 0, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: Time:                   327200,selector: 1, A: 15, B: 62, GT: 1, EQ: 0, LT: 0
# KERNEL: Time:                   327200,selector: 1, A: 15, B: 62, GT: 1, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: There is an error here:
# KERNEL: Time:                   327360,selector: 0, A: 15, B: 63, GT: 0, EQ: 0, LT: 1
# KERNEL: Time:                   327360,selector: 0, A: 15, B: 63, GT: 0, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL: Time:                   327520,selector: 1, A: 15, B: 63, GT: 1, EQ: 0, LT: 0
# KERNEL: Time:                   327520,selector: 1, A: 15, B: 63, GT: 1, EQ: 0, LT: 0
# KERNEL: ####################################
# KERNEL:
# KERNEL: ##############################
# KERNEL:  the results are not correct
# KERNEL: ##############################
>
```

Figure 4:Analyzer error detection

And that's happened because I changed the gate for output Smaller from or to and



```
   and #(5)g3(SM2,Eq1,Eq2,S3);//A[5:4] == B[5:4] && A[3:2]==B[3:2] && A[1:0]<B[1:0]

   or #(5)g4(Greater, G1, GRE1, GRE2);//A[5:4]>B[5:4] OR (A[5:4] == B[5:4] && A[3:2]>B[3:2]) OR (A[5:4] == B[5:4] && A[3:2]==B[3:2] && A[1:0]>B[1:0])
   and #(5)g5(Smaller, S1, SM1, SM2);//A[5:4]<B[5:4] OR (A[5:4] == B[5:4] && A[3:2]<B[3:2]) OR (A[5:4] == B[5:4] && A[3:2]==B[3:2] && A[1:0]<B[1:0])
   and #(5)g6(Equal, Eq1, Eq2, Eq3);//(A[5:4]==B[5:4] && A[3:2]==B[3:2] && A[1:0]==B[1:0])
ndmodule
```

Figure 5: Error introduce

Conclusion and Future works

This project successfully designed a 6-bit comparator for signed and unsigned numbers. The comparator performs equality, greater-than, and less-than operations accurately using basic logic gates and synchronous registers. Testing and error detection confirmed its reliability.

Future works:

- Expand the comparator to handle larger numbers.
- Optimize the design for better speed and efficiency.
- Add support for more number formats, like floating-point.
- Use the comparator in larger systems, such as processors.
- Explore error correction for improved reliability.