

**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY FOR HIGH QUALITY TRAINING DEPARTMENT**

ଓঁশেওঁশে



**HCMUTE**

**VLSI INTEGRATED CIRCUIT DESIGN**

**PROJECT: 8-BIT COMPARATOR CIRCUIT DESIGN**

**VLSI - CMOS**

**NGUYEN HONG DANG**

**Student ID:** **20119127**

**HOANG NGOC HUNG**

**Student ID:** **20119133**

**LE MINH NHAT**

**Student ID:** **20119183**

**LE VAN THINH**

**Student ID:** **20119160**

**Major: COMPUTER ENGINEERING TECHNOLOGY**

**Advisor: PHAM VAN KHOA, PhD**

**Ho Chi Minh City, May 31<sup>st</sup> 2023**

## **Contents**

<b>CHAPTER 1 INTRODUCTION .....</b>	<b>1</b>
<b>CHAPTER 2 OVERVIEW .....</b>	<b>3</b>
2.1 RESEARCH SUBJECTS .....	3
2.2 SUPPORT TOOLS .....	3
2.3 DIVISION OF WORK.....	3
<b>CHAPTER 3 IMPLEMENTATION CONTENT .....</b>	<b>5</b>
3.1 OVERVIEW OF COMPARATOR CIRCUITS .....	5
3.2 1-BIT MAGNITUDE COMPARATOR .....	6
3.3 2-BIT MAGNITUDE COMPARATOR .....	7
3.4 4-BIT MAGNITUDE COMPARATOR .....	9
3.5 CASCADING COMPARATOR .....	11
3.6 DESCRIPTION OF 8 - BIT COMPARATOR.....	11
3.7 STRUCTURAL DIAGRAM .....	12
<b>CHAPTER 4 SIMULATION RESULTS OF 8-BIT COMPARATOR IN CADENCE .....</b>	<b>15</b>
<b>CHAPTER 5 POWER &amp; DELAY.....</b>	<b>21</b>
5.1 POWER COMSUMPTION .....	21
5.2 PROPARATION DELAY.....	23
<b>CHAPTER 6 CONCLUTION &amp; DEVELOPMENT.....</b>	<b>33</b>
<b>CHAPTER 7 SEARCHING A NOVEL DESIGN OF 12-BIT DIGITAL COMPARATOR USING MULTIPLEXER FOR HIGH SPEED APPLICATION IN 32-NM CMOS TECHNOLOGY.....</b>	<b>34</b>
<b>RESEARCH NOTE .....</b>	<b>38</b>
<b>REFERENCES.....</b>	<b>45</b>

# CHAPTER 1 INTRODUCTION

In the world of digital electronics, the need for accurate and high-speed comparison of binary numbers is paramount. An important component to achieve this functionality is an 8-bit comparator. With advances in Very Large Scale Integration (VLSI) technology and Complementary Metal Oxide Semiconductor (CMOS) technology, the design of efficient and reliable 8-bit comparators has become an important mission.

An 8-bit comparator circuit is designed to compare two 8-bit binary numbers and provide an output indicating the relationship between them. It checks each pair of corresponding bits and determines if they are equal or one is greater or less than the other. This capability is important in various applications, such as arithmetic operations, data organization, and decision-making in digital systems.

VLSI technology plays an important role in the design and manufacture of integrated circuits (ICs) by integrating thousands to millions of transistors into a single chip. CMOS, a common technology used in VLSI design, offers advantages such as low power consumption, high noise immunity, and compatibility with modern fabrication processes. Leveraging the benefits of VLSI-CMOS technology, designers can create compact and efficient 8-bit comparator circuits that can be easily integrated into larger digital systems.

The 8-bit comparator circuit design in VLSI-CMOS involves many different stages, including transistor level design, layout generation, and verification. At the transistor level, the circuit is built using CMOS technology, using NMOS (n-channel metal oxide semiconductor) and PMOS (p-channel metal oxide semiconductor) transistors. The layout phase involves placing and routing the transistors for optimal performance and minimizing usable area. Verification ensures that the circuit functions correctly and meets the desired specifications.

Efficient design techniques, such as the use of parallel comparison architecture, advanced transistor sizing, and optimization algorithms, play an important role in achieving low and high-speed operation, power consumption. In addition, designers need to consider factors such as propagation delay, power dissipation, noise tolerance, and scalability while designing an 8-bit comparator.

The field of VLSI-CMOS circuit design continues to grow rapidly due to the increasing demand for high performance digital systems. The 8-bit comparator circuit design plays an important role in enabling accurate and efficient binary number comparison. As technology advances, new design methodologies and optimization techniques will continue to enhance the performance and capabilities of these circuits, further fueling the growth and innovation in the field of digital electronics.

# CHAPTER 2 OVERVIEW

## 2.1 RESEARCH SUBJECTS

- 90 nm technology

The 90 nm technology node revolutionized semiconductor manufacturing by enabling smaller, more powerful, and energy-efficient integrated circuits. It introduced strained silicon and copper interconnects, driving advancements in consumer electronics and other industries.

- IC 74HC85

The IC 74HC85 is a widely used integrated circuit (IC) that functions as an 8-bit magnitude comparator. It offers accurate binary number comparison, low power consumption, and high noise immunity, making it a valuable component in various digital systems and electronic designs.

## 2.2 SUPPORT TOOLS

Cadence is a global leader in electronic design automation (EDA) software, offering a comprehensive suite of tools for designing integrated circuits and electronic systems. With a focus on innovation and collaboration, Cadence enables engineers to streamline the design process and create advanced electronic devices efficiently and reliably.

## 2.3 DIVISION OF WORK

Works/Members	Nguyễn Hồng Đăng	Hoàng Ngọc Hùng	Lê Minh Nhật	Lê Văn Thịnh
Introduction		✓		✓

Overview of Comparator Circuit		✓	✓	
Design 4 – bit comparator	✓		✓	
Design 8 – bit comparator		✓		✓
Simulation and Check the result	✓			✓
Power and Delay	✓			
Research a novel design of 12 – bit digital comparator (using 32 nm technology, 0.7 V)		✓	✓	✓
Research a novel design of 8 – bit digital comparator (using ... nm technology, ... V)	✓		✓	
Report	✓	✓	✓	✓

# CHAPTER 3 IMPLEMENTATION CONTENT

## 3.1 OVERVIEW OF COMPARATOR CIRCUITS

A magnitude digital comparator is a combinational circuit that compares two digital or binary numbers in order to find out whether one binary number is equal, less than, or greater than the other binary number. We logically design a circuit for which we will have two inputs one for A and the other for B and have three output terminals, one for  $A > B$  condition, one for  $A = B$  condition, and one for  $A < B$  condition.



The circuit works by comparing the bits of the two numbers starting from the most significant bit (MSB) and moving toward the least significant bit (LSB). At each bit position, the two corresponding bits of the numbers are compared. If the bit in the first number is greater than the corresponding bit in the second number, the  $A > B$  output is set to 1, and the circuit immediately determines that the first number is greater than the second. Similarly, if the bit in the second number is greater than the corresponding bit in the first number, the  $A < B$  output is set to 1, and the circuit immediately determines that the first number is less than the second.

If the two corresponding bits are equal, the circuit moves to the next bit position and compares the next pair of bits. This process continues until all the bits have been compared. If at any point in the comparison, the circuit determines that the first number is greater or less than the second number, the comparison is terminated, and the appropriate output is generated.

If all the bits are equal, the circuit generates an  $A=B$  output, indicating that the two numbers are equal.

There are different ways to implement a magnitude comparator, such as using a combination of XOR, AND, and OR gates, or by using a cascaded arrangement of full adders. The choice of implementation depends on factors such as speed, complexity, and power consumption.

### 3.2 1-BIT MAGNITUDE COMPARATOR

A comparator used to compare two bits is called a single-bit comparator. It consists of two inputs each for two single-bit numbers and three outputs to generate less than, equal to, and greater than between two binary numbers.

The truth table for a 1-bit comparator is given below:

<b>A</b>	<b>B</b>	<b><math>A &lt; B</math></b>	<b><math>A = B</math></b>	<b><math>A &gt; B</math></b>
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

From the above truth table logical expressions for each output can be expressed as follows:

$$A > B: AB'$$

$$A < B: A'B$$

$$A = B: A'B' + AB$$

From the above expressions we can derive the following formula:

$$(A < B) + (A > B) = \bar{A}B + A\bar{B}$$

Taking complement both sides:

$$\overline{(A < B) + (A > B)} = \overline{\bar{A}B + A\bar{B}}$$

$$\overline{(A < B) + (A > B)} = \overline{(\bar{A}'B)} \overline{(A\bar{B})}$$

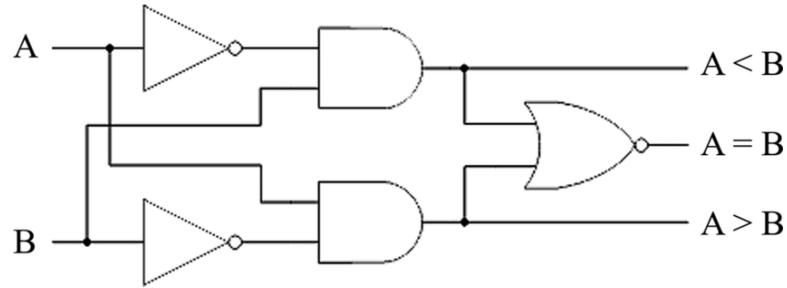
$$\overline{(A < B) + (A > B)} = A\bar{A} + AB + \bar{A}\bar{B} + B\bar{B}$$

$$\overline{(A < B) + (A > B)} = AB + \bar{A}\bar{B}$$

Thus,

$$\overline{(A < B) + (A > B)} = (A = B)$$

By using these Boolean expressions, we can implement a logic circuit for this comparator as given below:



### 3.3 2-BIT MAGNITUDE COMPARATOR

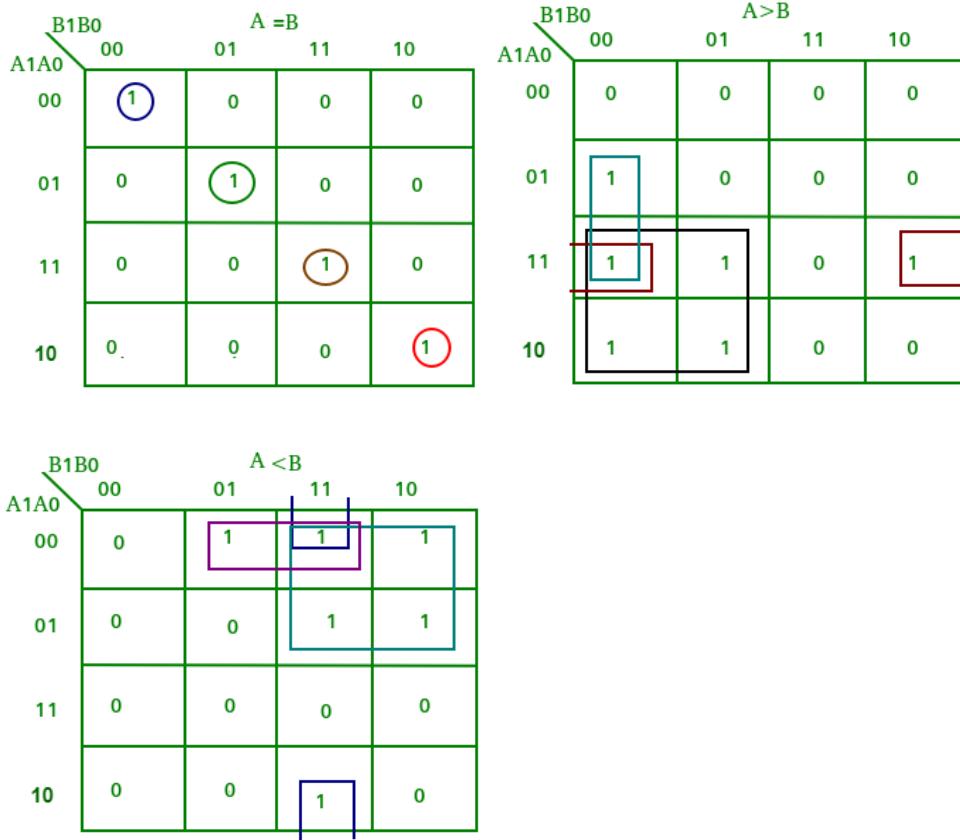
A comparator used to compare two binary numbers each of two bits is called a 2-bit Magnitude Comparator. It consists of four inputs and three outputs to generate less than, equal to, and greater than between two binary numbers.

The truth table for a 2-bit comparator is given below:

INPUT				OUTPUT		
A1	A0	B1	B0	A < B	A = B	A > B
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1

1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

From the above truth table K-map for each output can be drawn as follows:



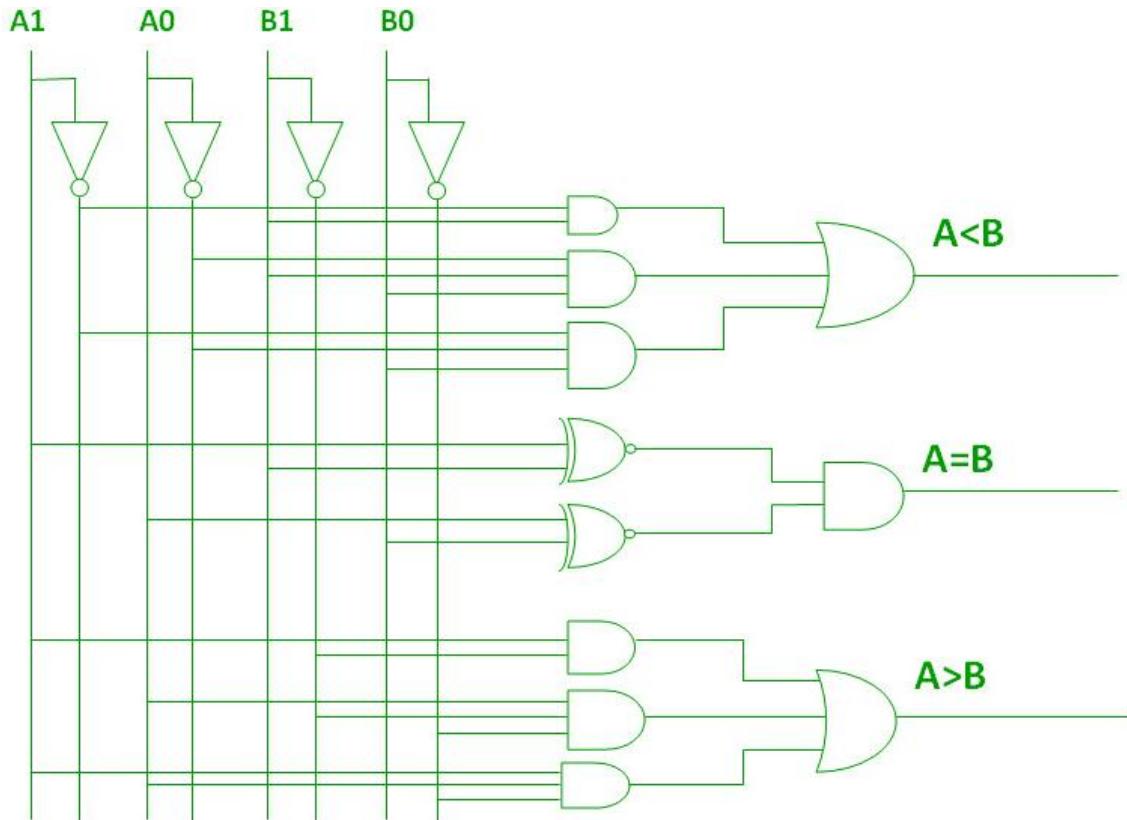
From the above K-maps logical expressions for each output can be expressed as follows:

$$A > B: \quad A1\bar{B}1 + A0\bar{B}1\bar{B}0 + A1A0\bar{B}0$$

$$\begin{aligned} A = B: \quad & \bar{A}1\bar{A}0\bar{B}1\bar{B}0' + \bar{A}1A0\bar{B}1B0 + A1A0B1B0 + A1\bar{A}0B1\bar{B}0 \\ &= \bar{A}1\bar{B}1(\bar{A}0\bar{B}0 + A0B0) + A1B1(A0B0 + \bar{A}0\bar{B}0) \\ &= (A0B0 + \bar{A}0\bar{B}0)(A1B1 + \bar{A}1\bar{B}1) \\ &= (A0 \text{ EX-NOR } B0)(A1 \text{ EX-NOR } B1) \end{aligned}$$

$$A < B: \quad \bar{A}1B1 + \bar{A}0B1B0 + \bar{A}1\bar{A}0B0$$

By using these Boolean expressions, we can implement a logic circuit for this comparator as given below:



### 3.4 4-BIT MAGNITUDE COMPARATOR

A comparator used to compare two binary numbers each of four bits is called a 4-bit magnitude comparator. It consists of eight inputs each for two four-bit numbers and three outputs to generate less than, equal to, and greater than between two binary numbers.

In a 4-bit comparator the condition of  $A > B$  can be possible in the following four cases:

- If  $A_3 = 1$  and  $B_3 = 0$
- If  $A_3 = B_3$  and  $A_2 = 1$  and  $B_2 = 0$
- If  $A_3 = B_3$ ,  $A_2 = B_2$  and  $A_1 = 1$  and  $B_1 = 0$

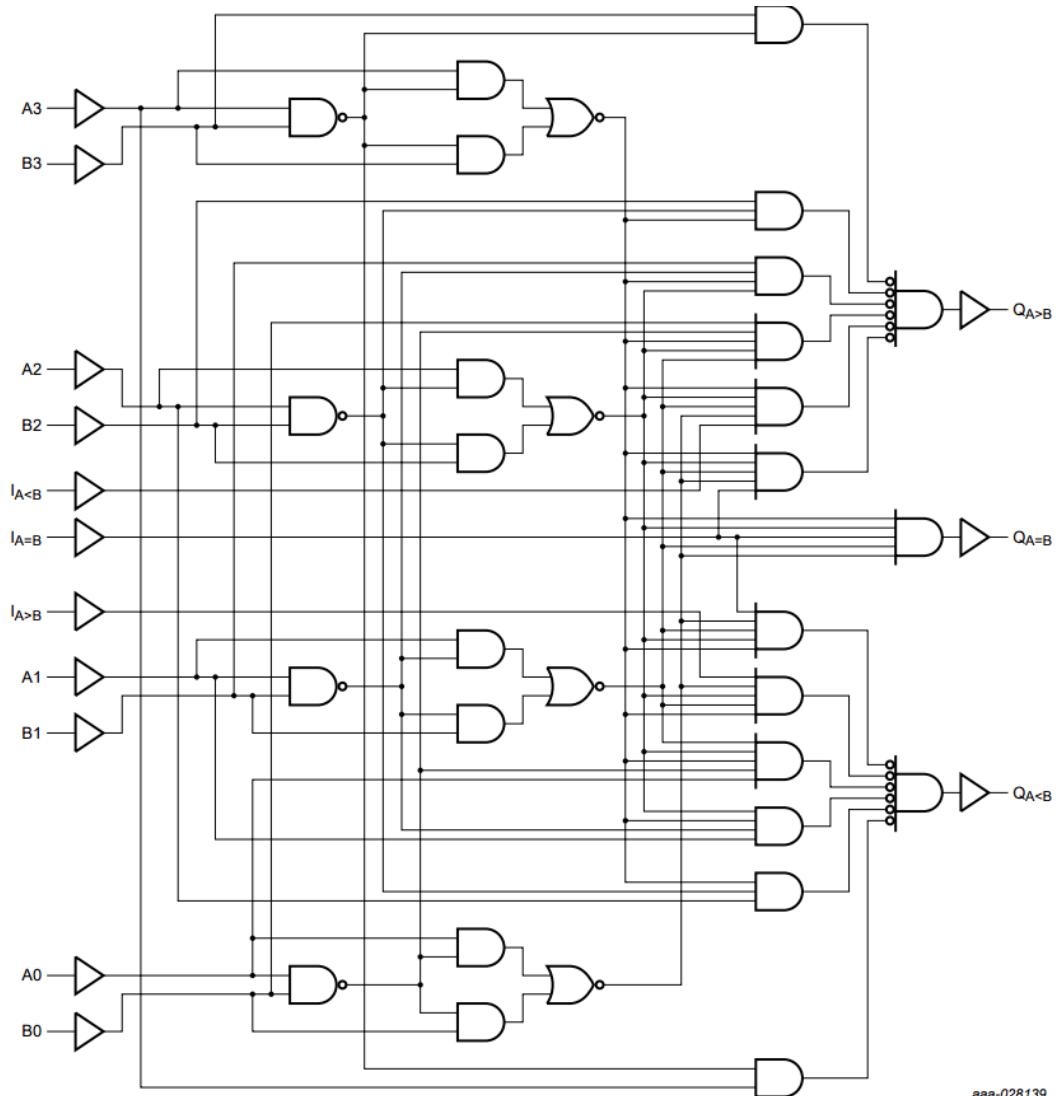
- If  $A_3 = B_3$ ,  $A_2 = B_2$ ,  $A_1 = B_1$  and  $A_0 = 1$  and  $B_0 = 0$

Similarly the condition for  $A < B$  can be possible in the following four cases:

- If  $A_3 = 0$  and  $B_3 = 1$
- If  $A_3 = B_3$  and  $A_2 = 0$  and  $B_2 = 1$
- If  $A_3 = B_3$ ,  $A_2 = B_2$  and  $A_1 = 0$  and  $B_1 = 1$
- If  $A_3 = B_3$ ,  $A_2 = B_2$ ,  $A_1 = B_1$  and  $A_0 = 0$  and  $B_0 = 1$

The condition of  $A = B$  is possible only when all the individual bits of one number exactly coincide with corresponding bits of another number.

A logic circuit for this comparator as given below:

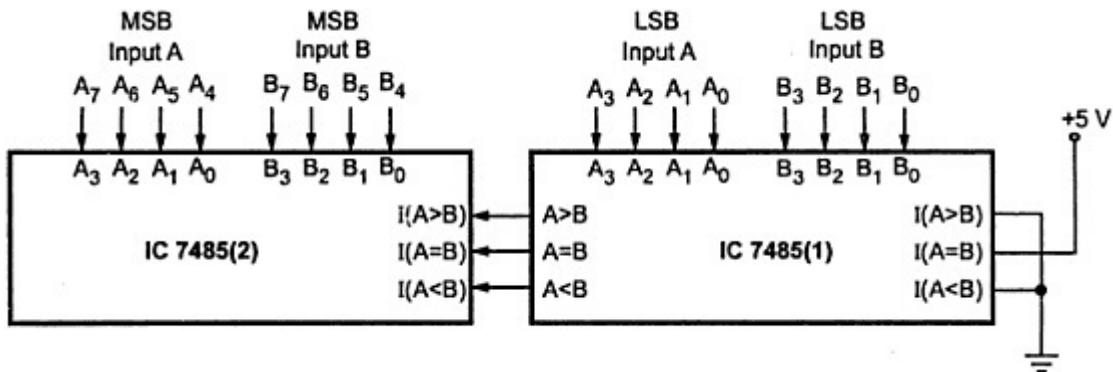


aaa-028139

- NOTE:** For  $n$ - bit comparator then, the number of combinations for which
- $A = B$  is  $2^n$
  - $A > B$  or  $A < B$  is  $(2^{2n} - 2^n)/2$

### 3.5 CASCADING COMPARATOR

A comparator performing the comparison operation to more than four bits by cascading two or more 4-bit comparators is called a cascading comparator. When two comparators are to be cascaded, the outputs of the lower-order comparator are connected to the corresponding inputs of the higher-order comparator.



### 3.6 DESCRIPTION OF 8 - BIT COMPARATOR

An 8-bit comparator can be implemented using two 74HC85 ICs, which are 4-bit magnitude comparators. The operating principle of an 8-bit comparator using two 74HC85 ICs is based on dividing the 8-bit input into two 4-bit inputs and comparing them separately using the two 74HC85 ICs. The outputs of the two 74HC85 ICs are then combined to generate the final outputs of the 8-bit comparator.

Here is the operating principle of an 8-bit comparator using two 74HC85 ICs:

Divide the 8-bit input into two 4-bit inputs, A and B.

Apply the A input to the A<sub>3</sub>-A<sub>0</sub> inputs of the first 74HC85 IC and the B input to the B<sub>3</sub>-B<sub>0</sub> inputs of the same IC.

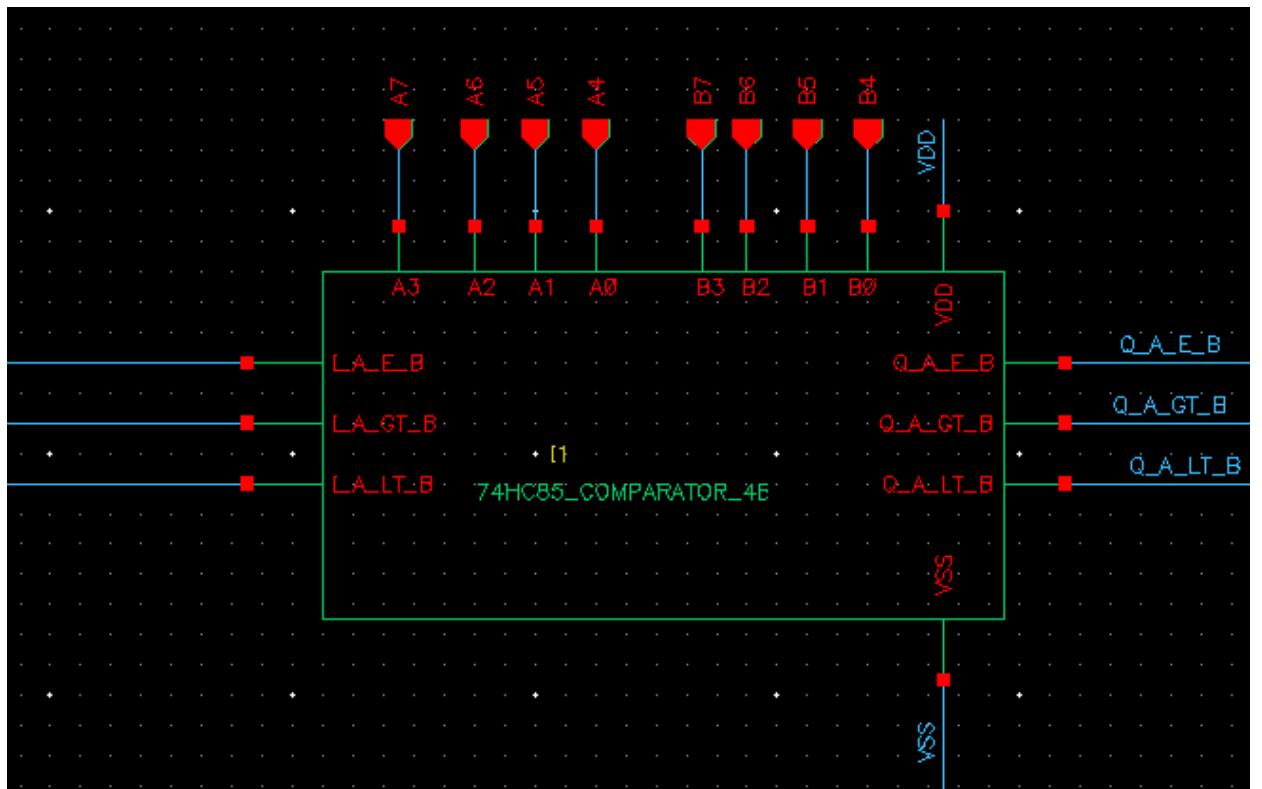
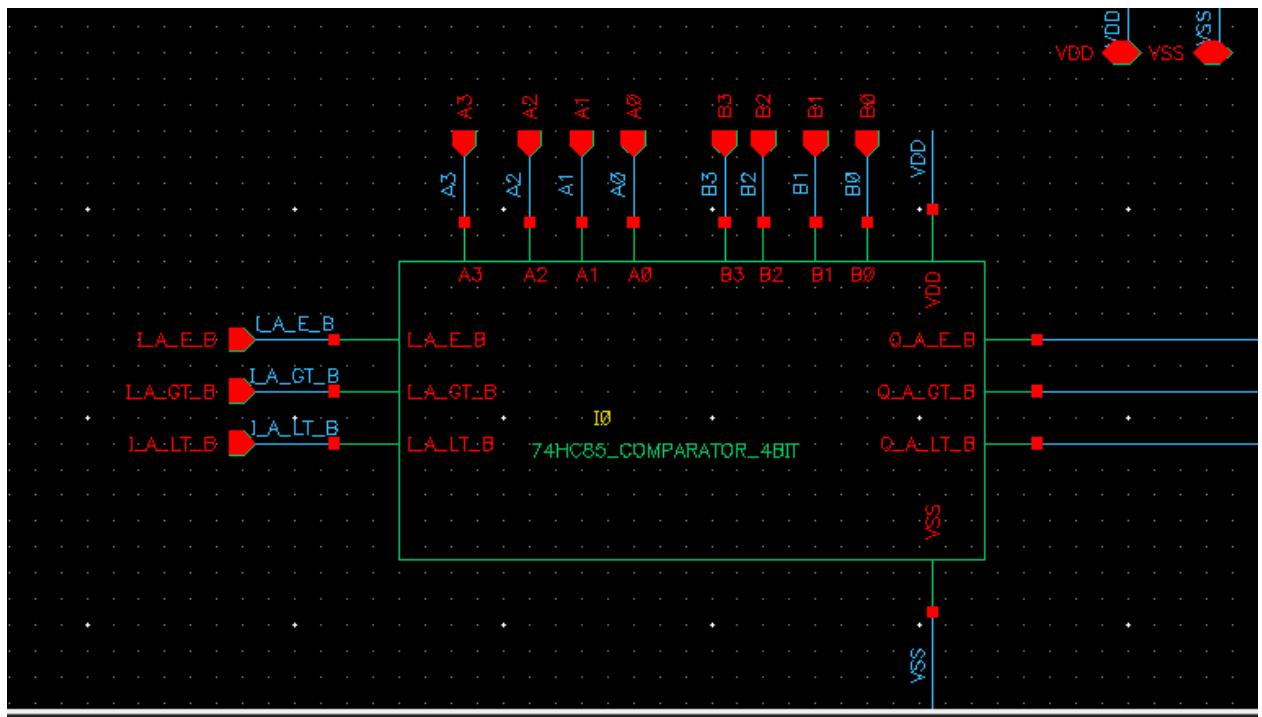
The operating principle of the 8-bit comparator using two 74HC85 ICs is based on dividing the input into two parts and comparing them separately. The outputs of the two comparators are then combined using logic gates to generate the final result. This approach is commonly used when designing digital circuits that require more than 4-bit comparisons.

This is the truth table of IC 7485.

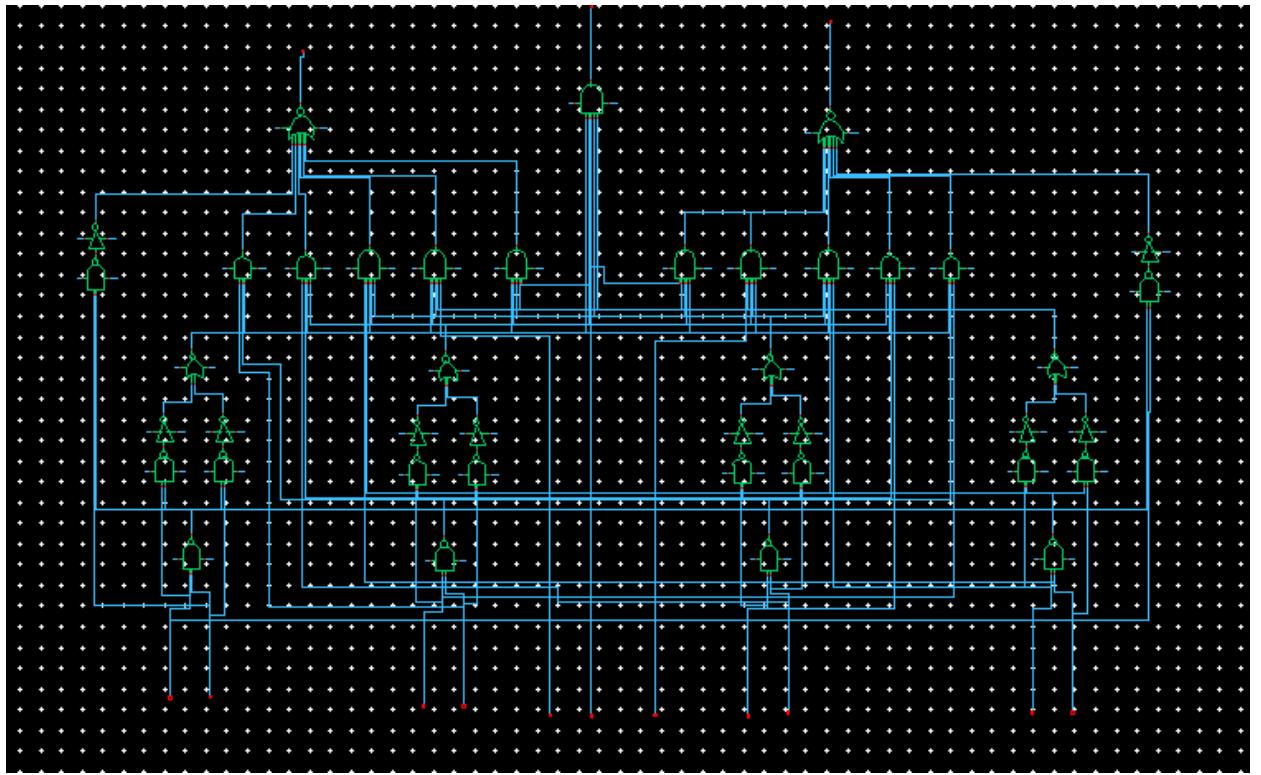
COMPARATOR INPUTS				CASCADING INPUTS			OUTPUTS		
$A_3, B_3$	$A_2, B_2$	$A_1, B_1$	$A_0, B_0$	$I_A > I_B$	$I_A < I_B$	$I_A = I_B$	$O_A > O_B$	$O_A < O_B$	$O_A = O_B$
$A_3 > B_3$	X	X	X	X	X	X	H	L	L
$A_3 < B_3$	X	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X	X	X	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	H	L	L	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	H	L	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	X	X	H	L	H	H
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	H	H	L	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	L	L	H	L	L

### 3.7 STRUCTURAL DIAGRAM

- Inside of a comparator 8bit include 2 comparator 4 bit



- In side a comparator 4 bit:



# CHAPTER 4

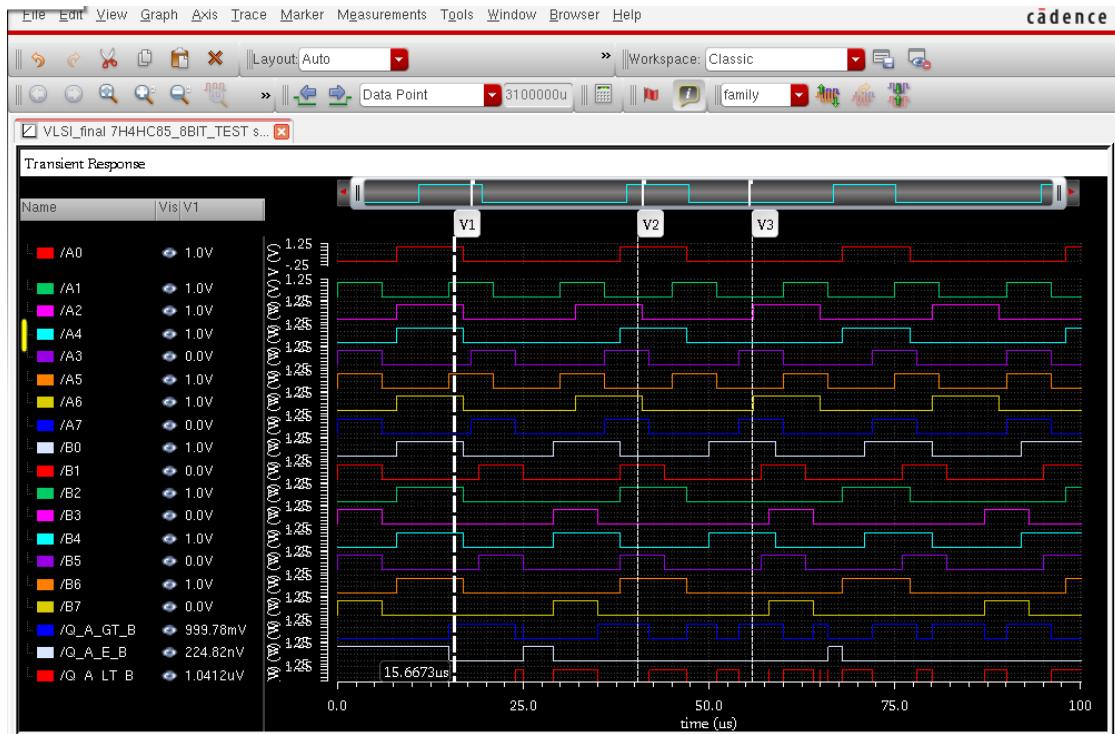
## SIMULATION RESULTS OF 8-BIT COMPARATOR IN CADENCE

a) A greater than B

- Case 1:

A: 01101111 – Decimal: 111

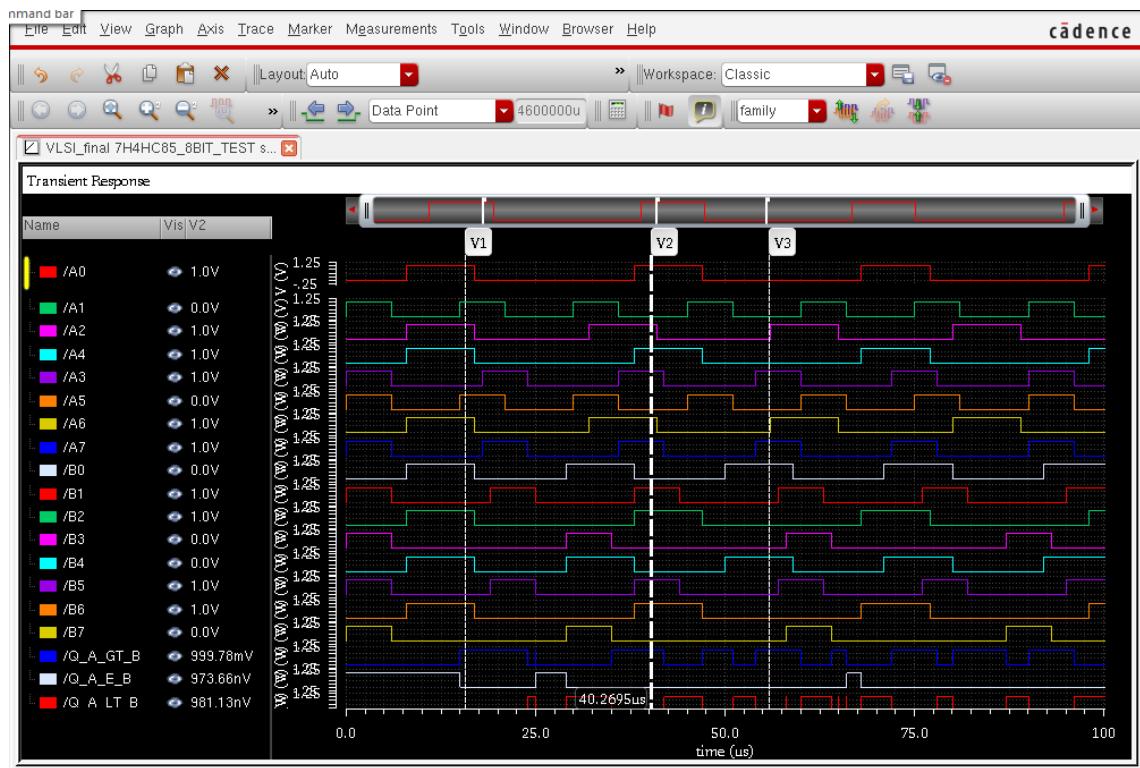
B: 01010101 – Decimal: 85



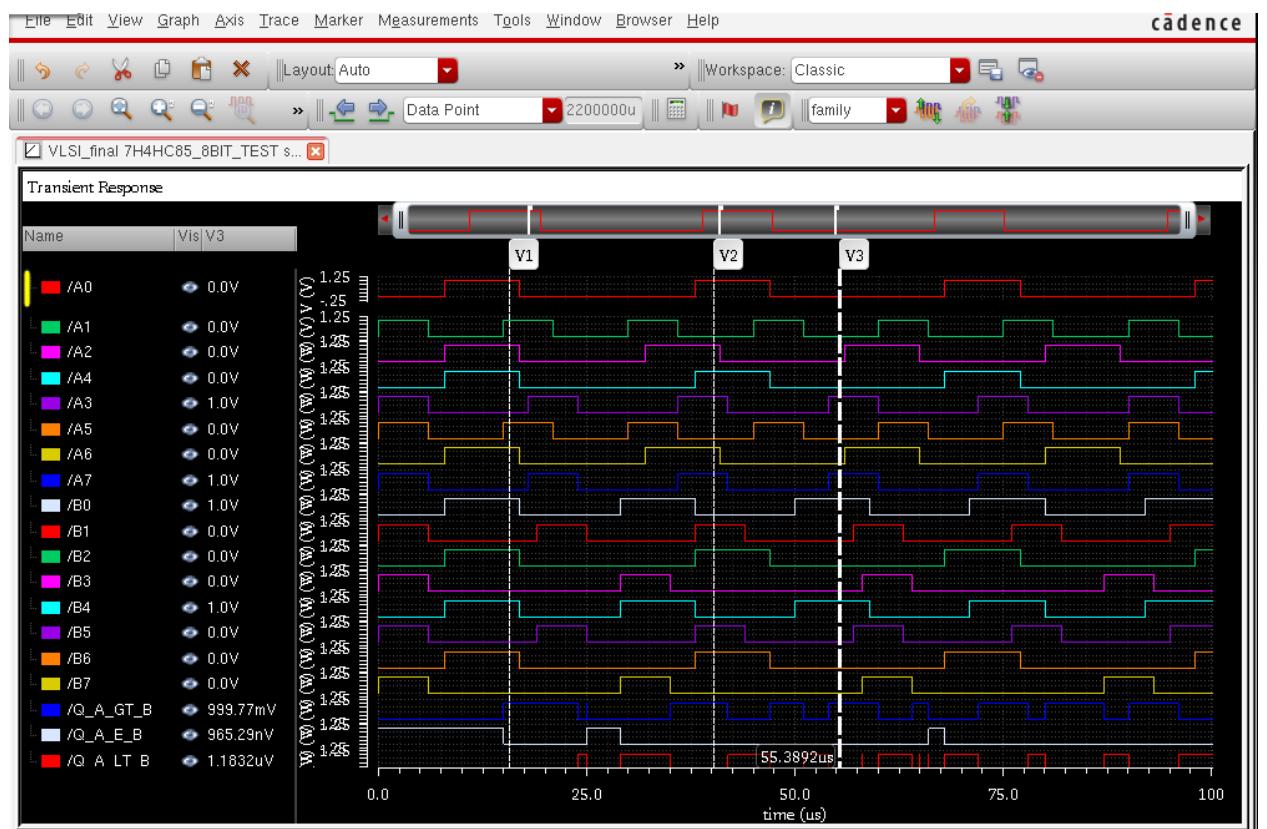
- Case 2:

A: 11011101 – Decimal: 221

B: 01100110 – Decimal: 102



- Case 3:
  - A: 10010000 – Decimal: 144
  - B: 00010001 – Decimal: 17

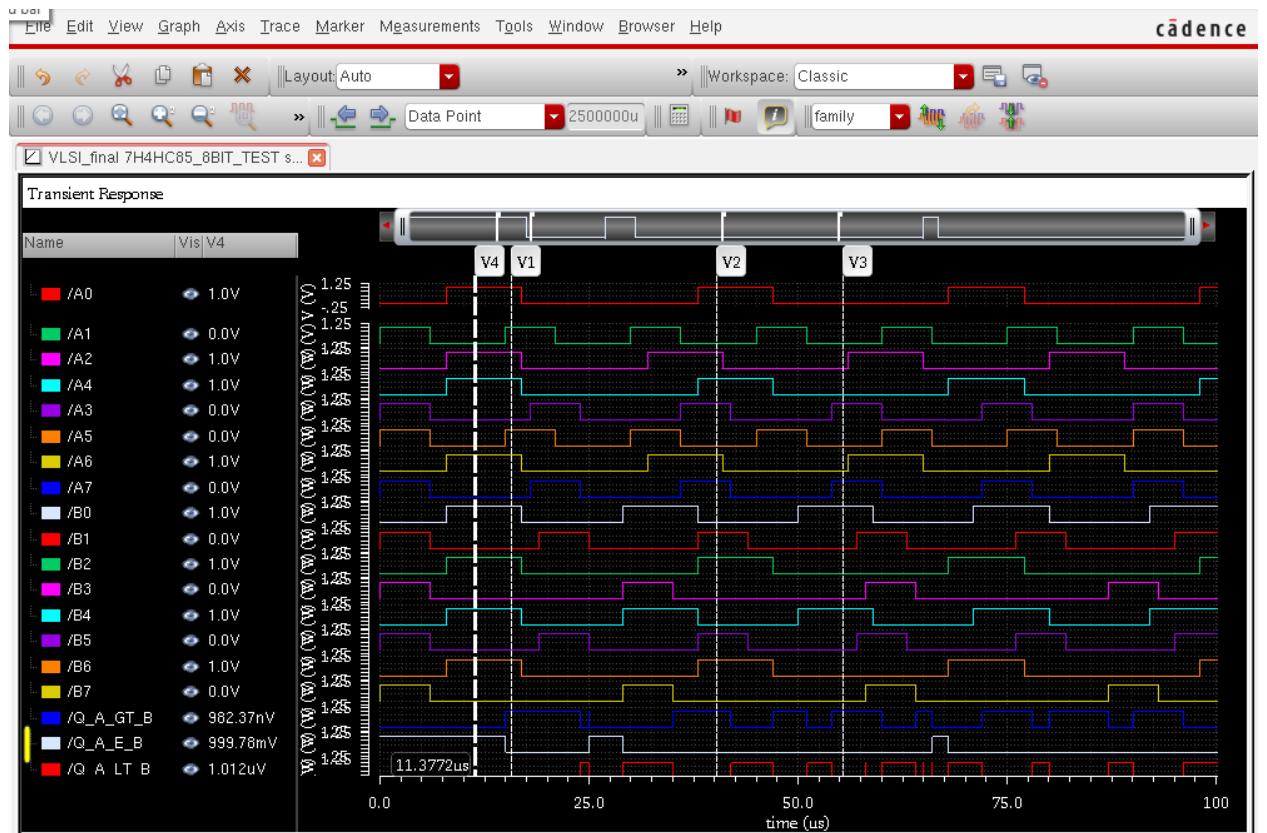


b) A equal to B

Case 1:

A: 01001101 – Decimal: 77

B: 01001101 – Decimal: 77

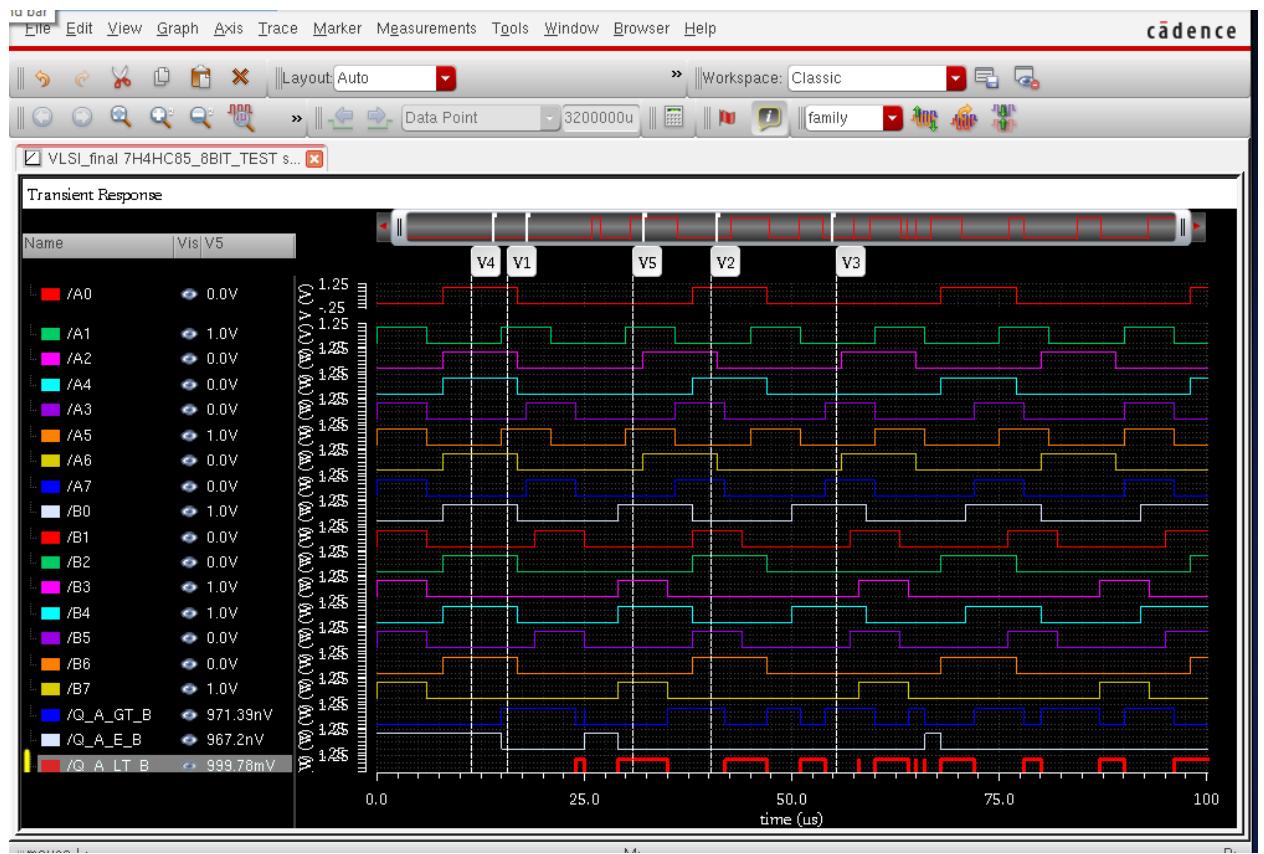


c) A less than B

- Case 1:

A: 00100010 – Decimal: 34

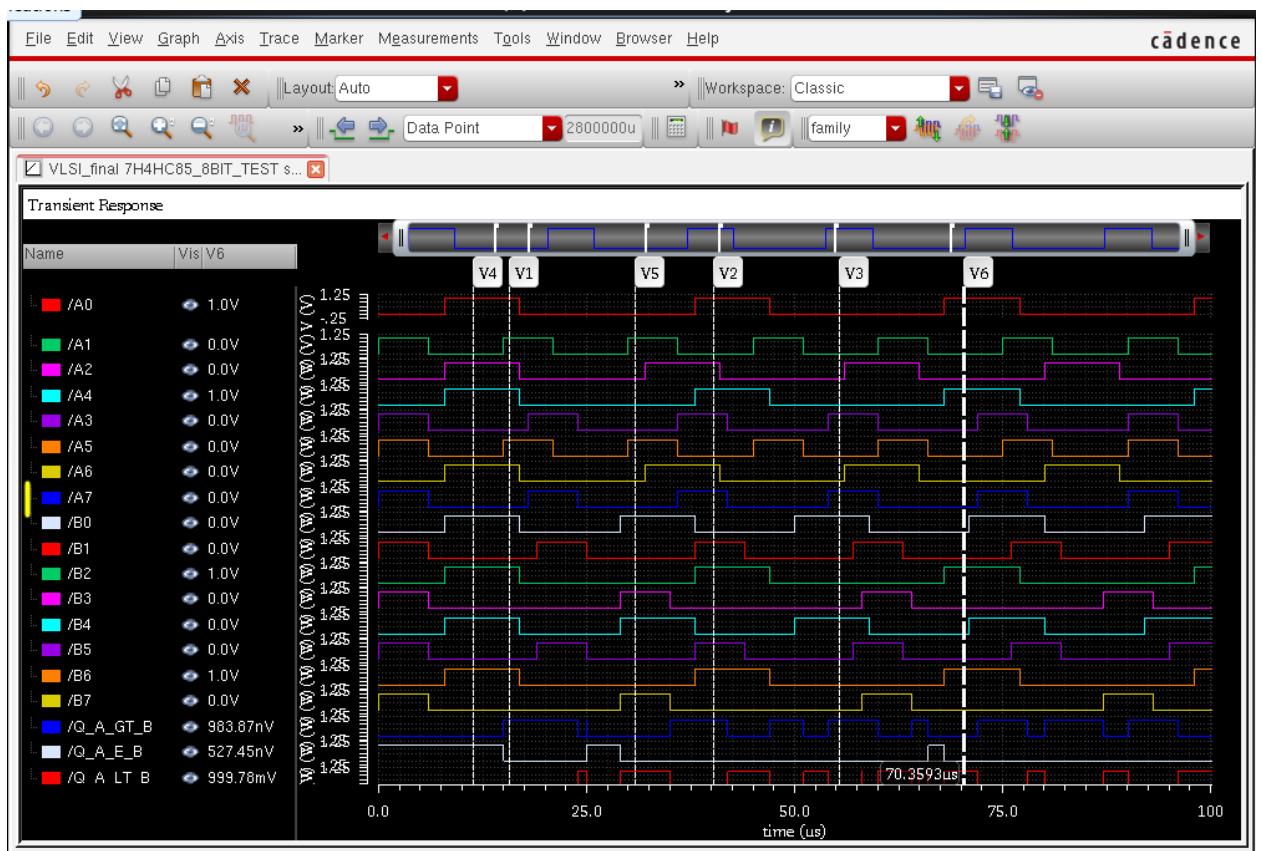
B: 10011001 – Decimal: 153



- Case 2:

A: 00001001 – Decimal: 9

B: 01000100 – Decimal: 68



- Case 3:

A: 00100010 – Decimal: 34

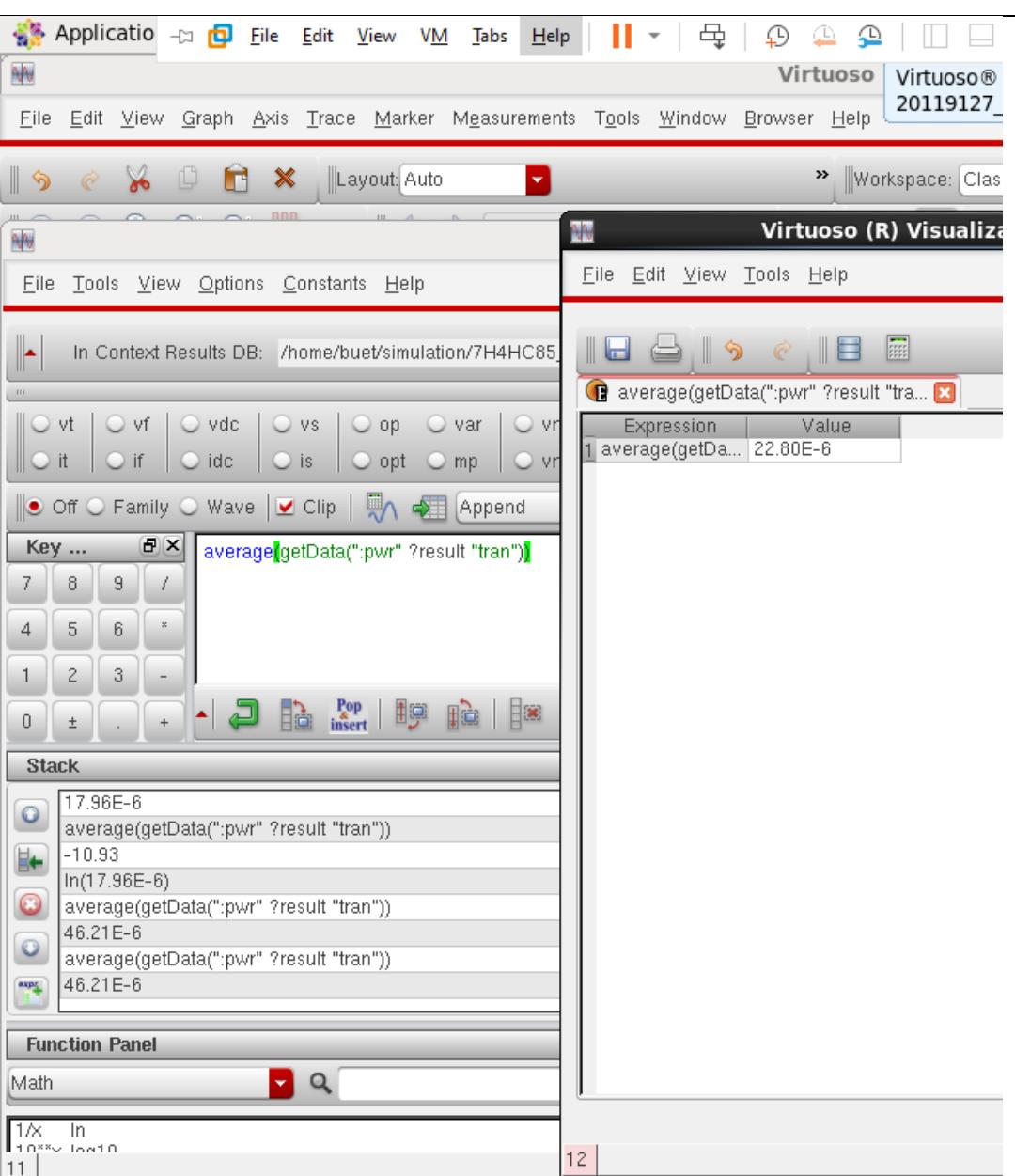
B: 00110011 – Decimal: 51

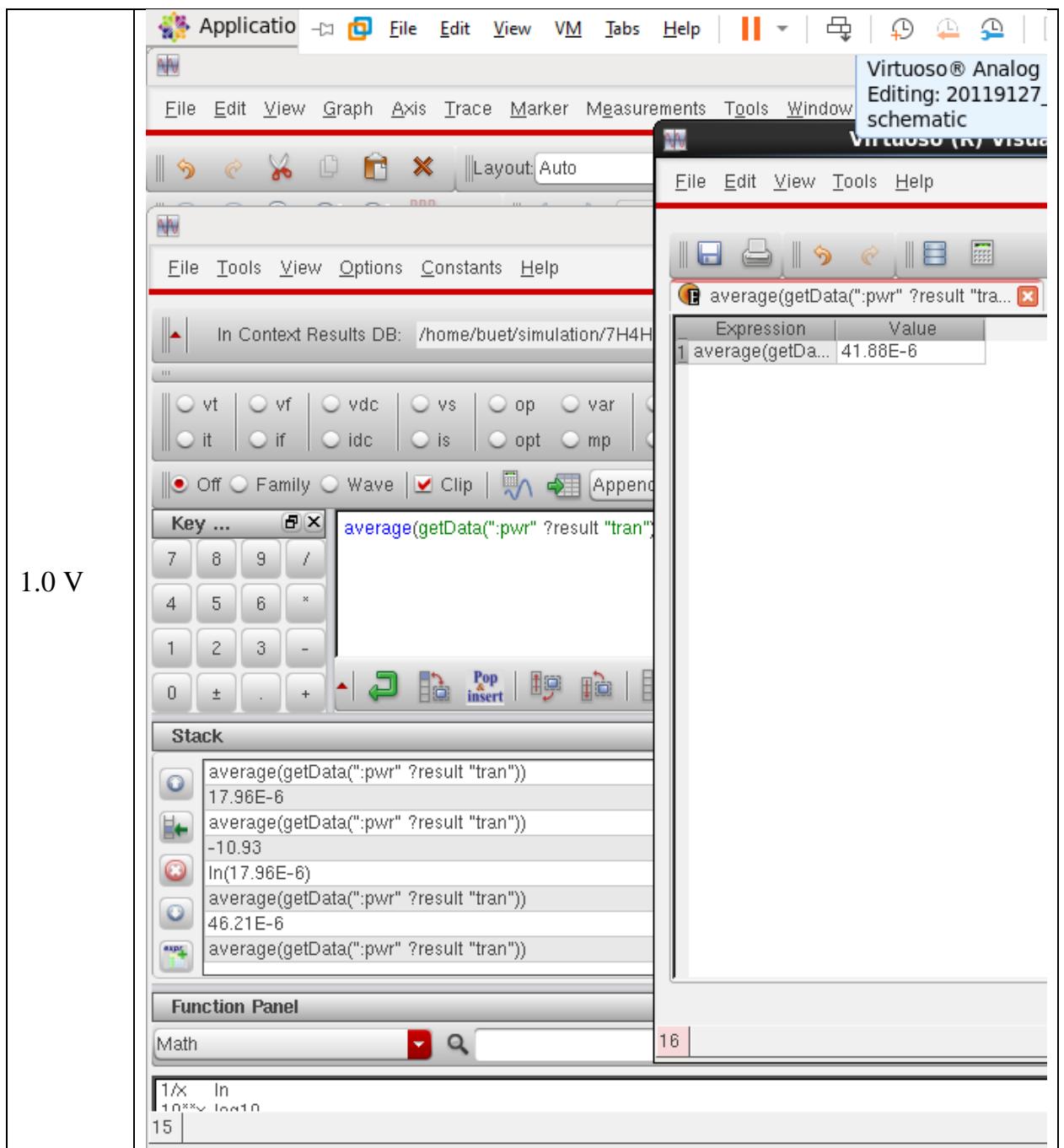


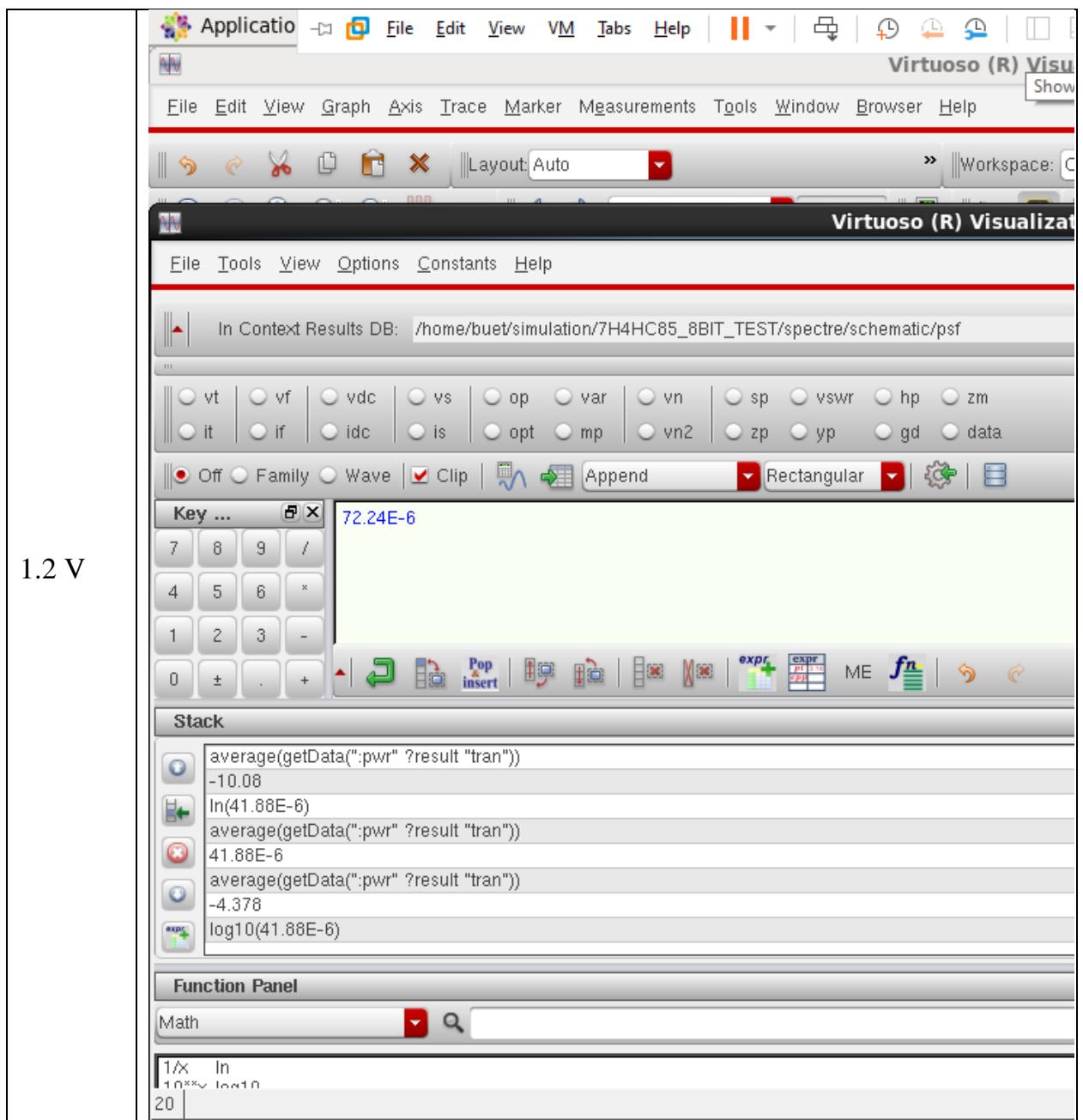
# CHAPTER 5 POWER & DELAY

## 5.1 POWER COMSUMPTION

The frequency 100MHz, 90 nm technology

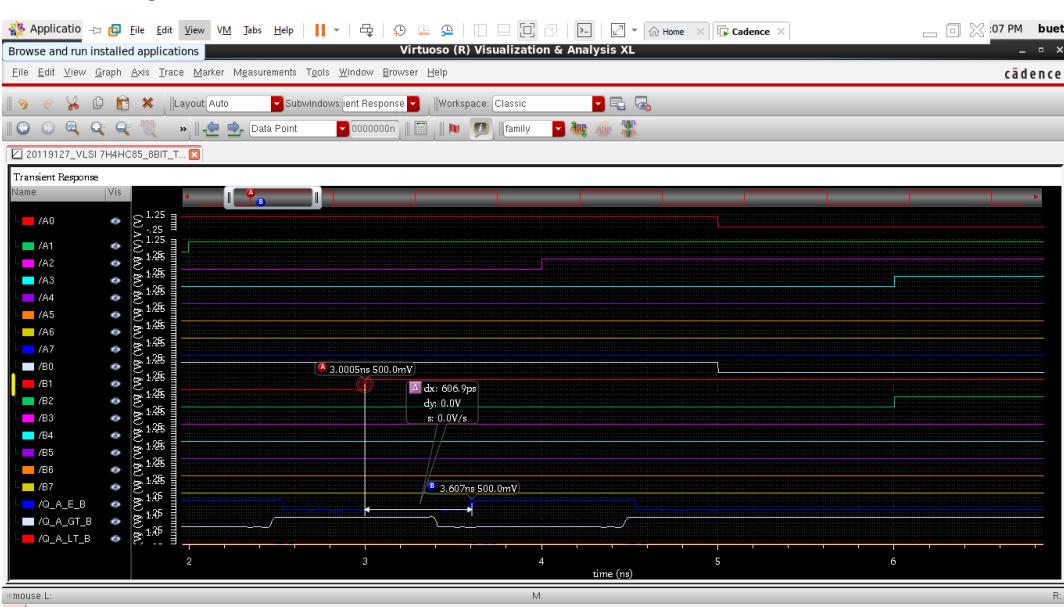
Voltage					
0.8 V	 <p>The screenshot shows the Virtuoso software interface with two main windows. The left window is titled 'Virtuoso (R) Visualizer' and displays a table with one row:</p> <table border="1"><thead><tr><th>Expression</th><th>Value</th></tr></thead><tbody><tr><td>average(getData(":pwr" ?result "tran"))</td><td>22.80E-6</td></tr></tbody></table> <p>The right window shows the 'Application' menu bar and various toolbars. A search bar at the top contains the query 'average(getData(":pwr" ?result "tran"))'. Below it is a stack of recent command history:</p> <ul style="list-style-type: none"><li>average(getData(":pwr" ?result "tran"))</li><li>-10.93</li><li>In(17.96E-6)</li><li>average(getData(":pwr" ?result "tran"))</li><li>46.21E-6</li><li>average(getData(":pwr" ?result "tran"))</li><li>46.21E-6</li></ul> <p>At the bottom, there is a function panel with a 'Math' dropdown and a search bar.</p>	Expression	Value	average(getData(":pwr" ?result "tran"))	22.80E-6
Expression	Value				
average(getData(":pwr" ?result "tran"))	22.80E-6				

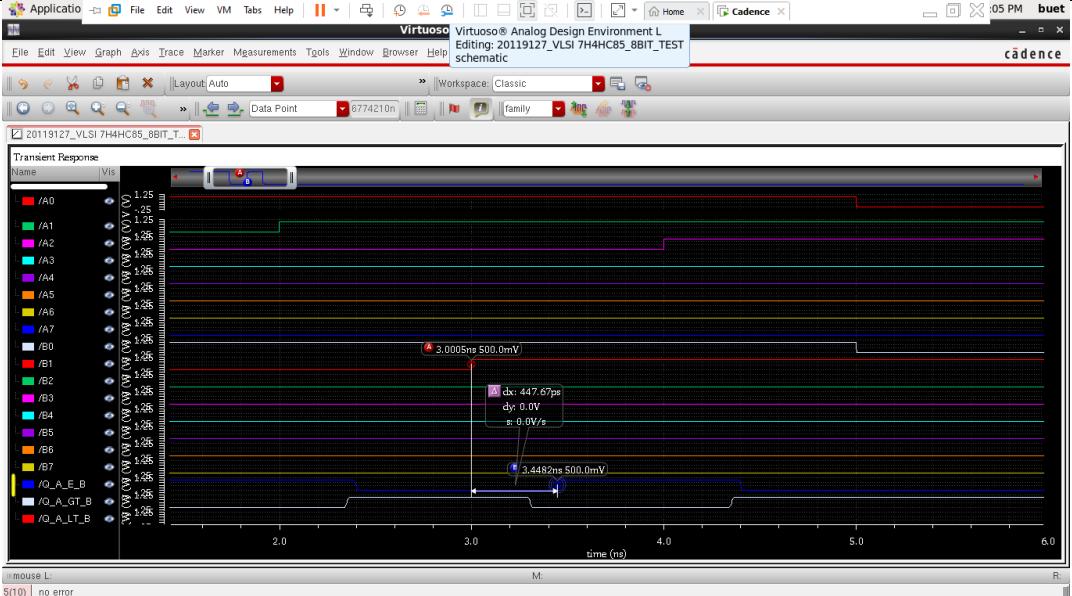
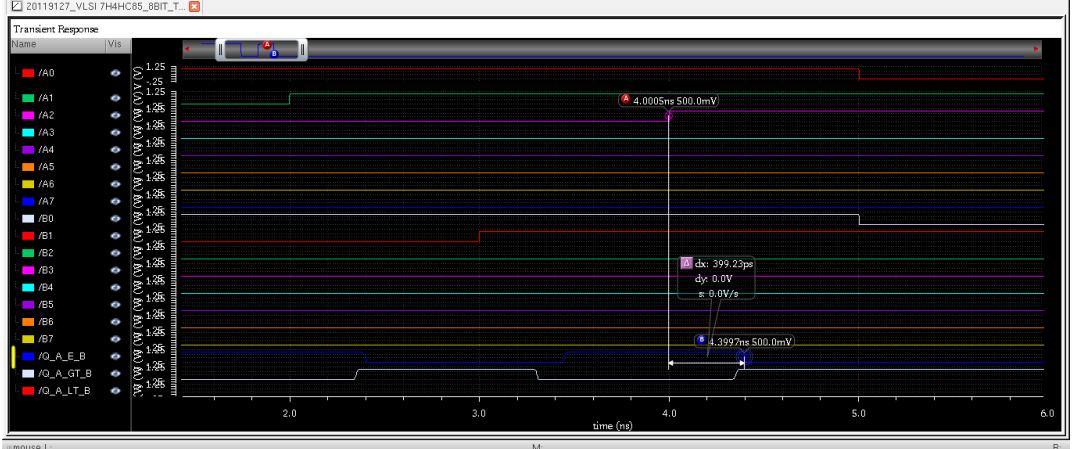


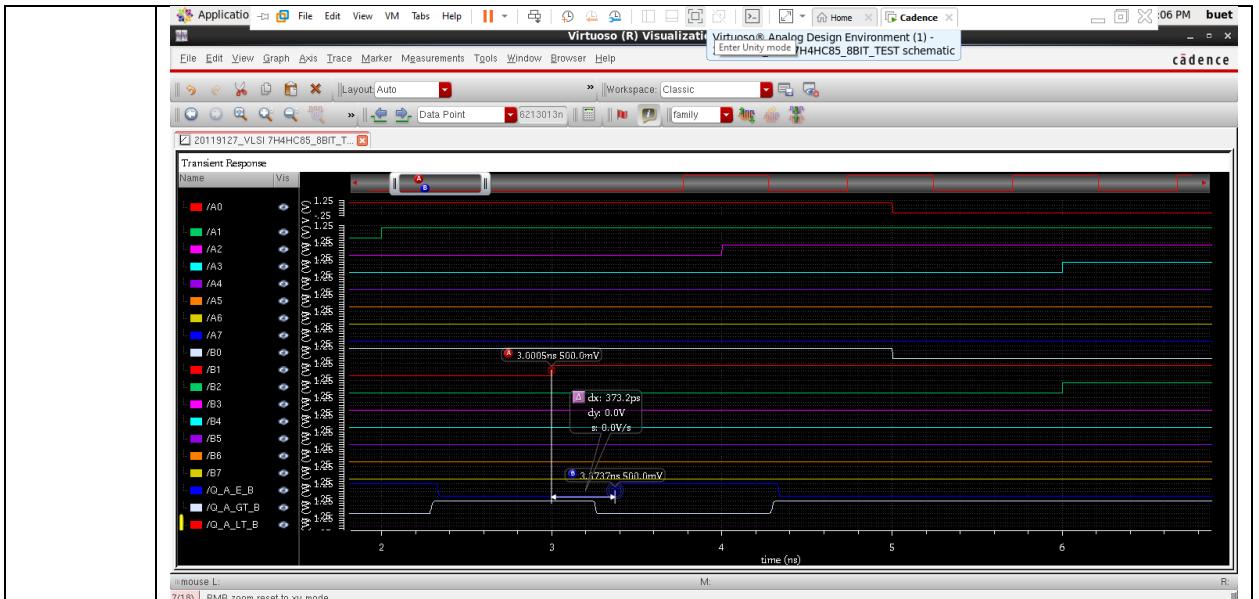


## 5.2 PROPARATION DELAY

The frequency 100MHz, 90 nm technology

Voltage	A is equal to B
0.8 V	<p>Low to High</p> 
1.0 V	<p>High to Low</p> 
1.0 V	Low to High

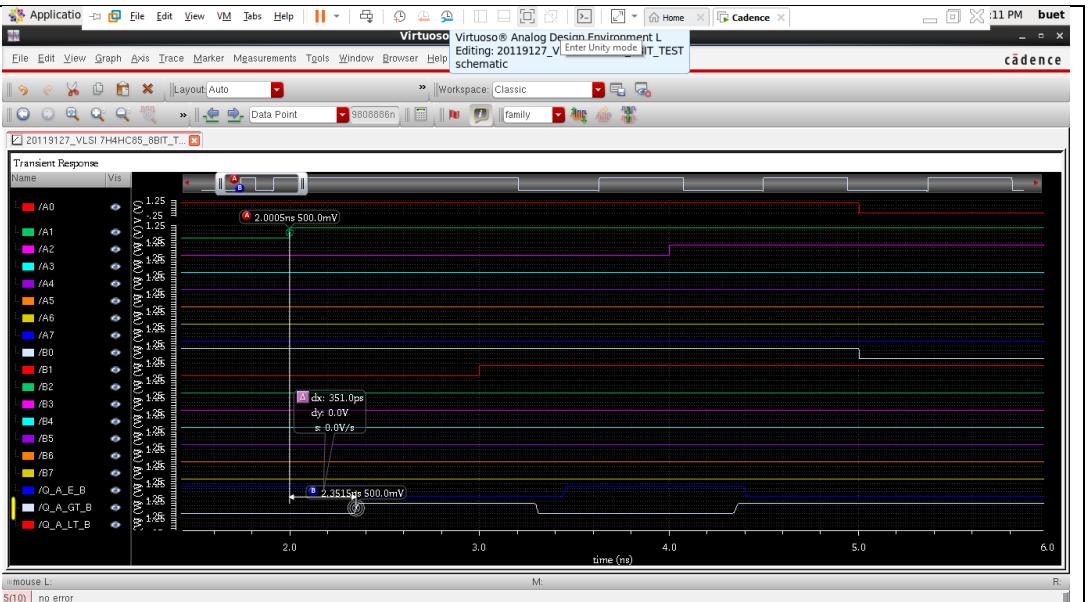
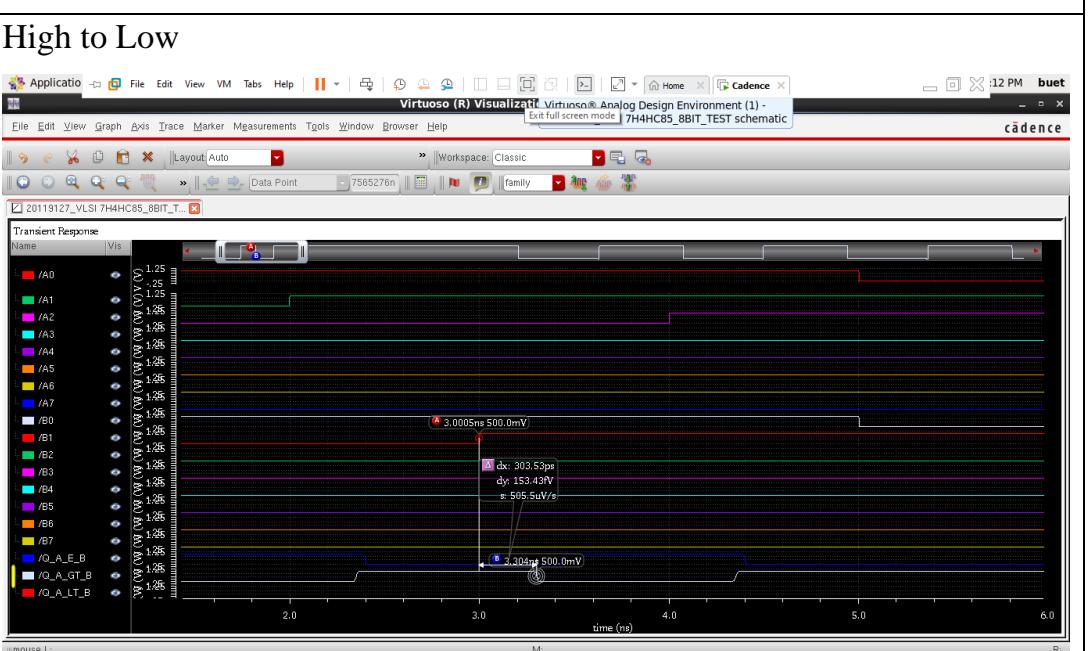
	
High to Low	
1.2 V	Low to High

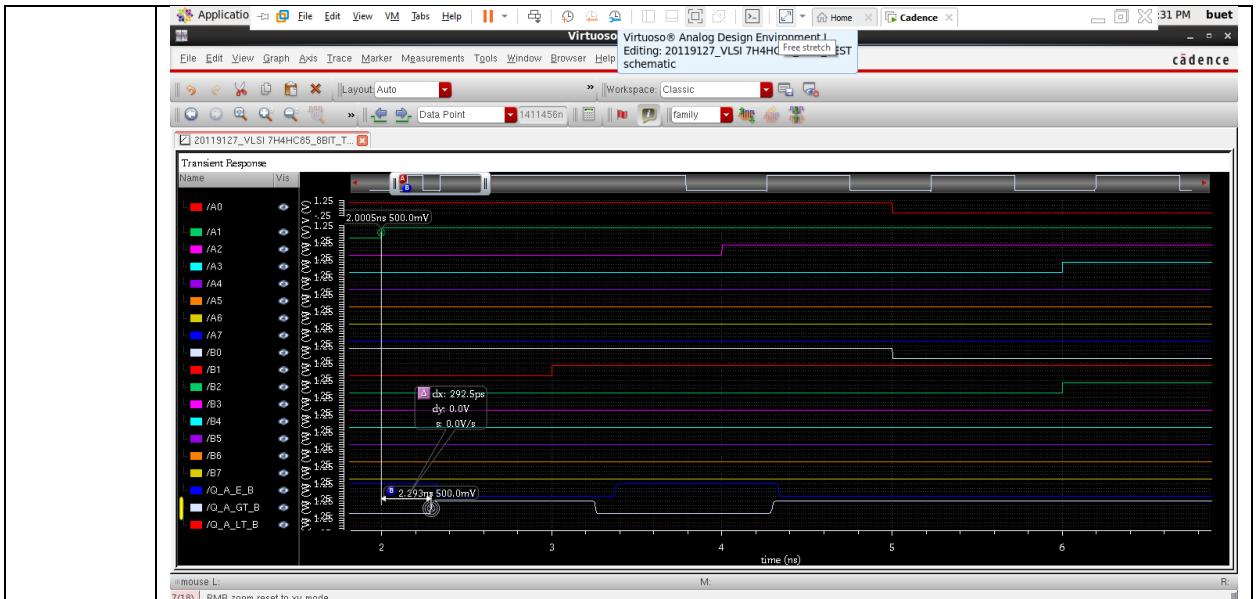


## High to Low

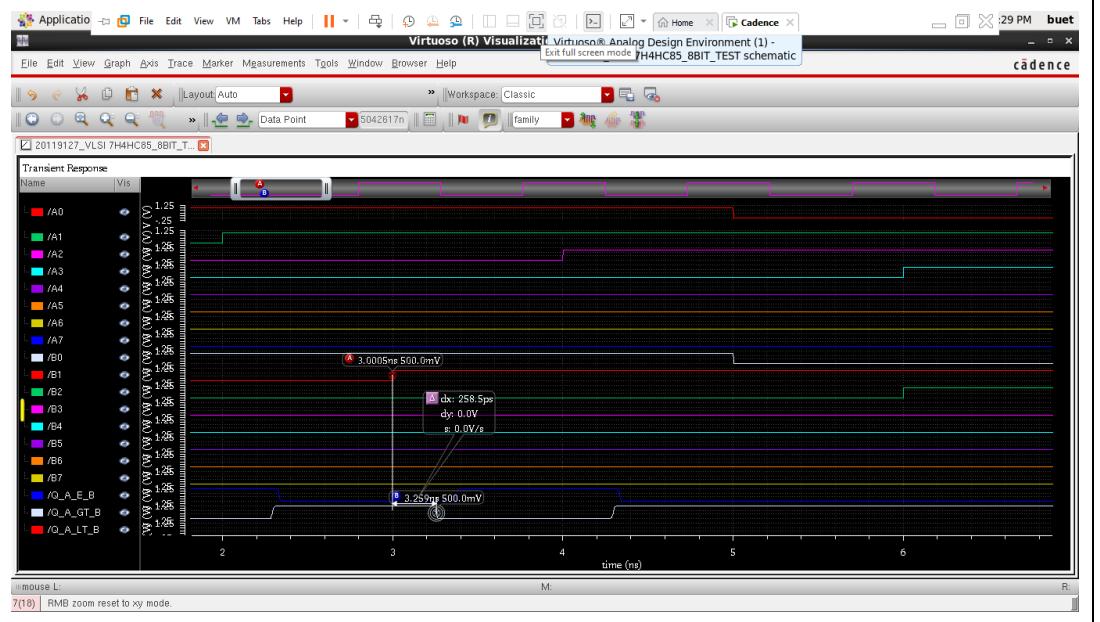


Voltage	A is greater than B
Low to High	
0.8 V	
1.0 V	Low to High

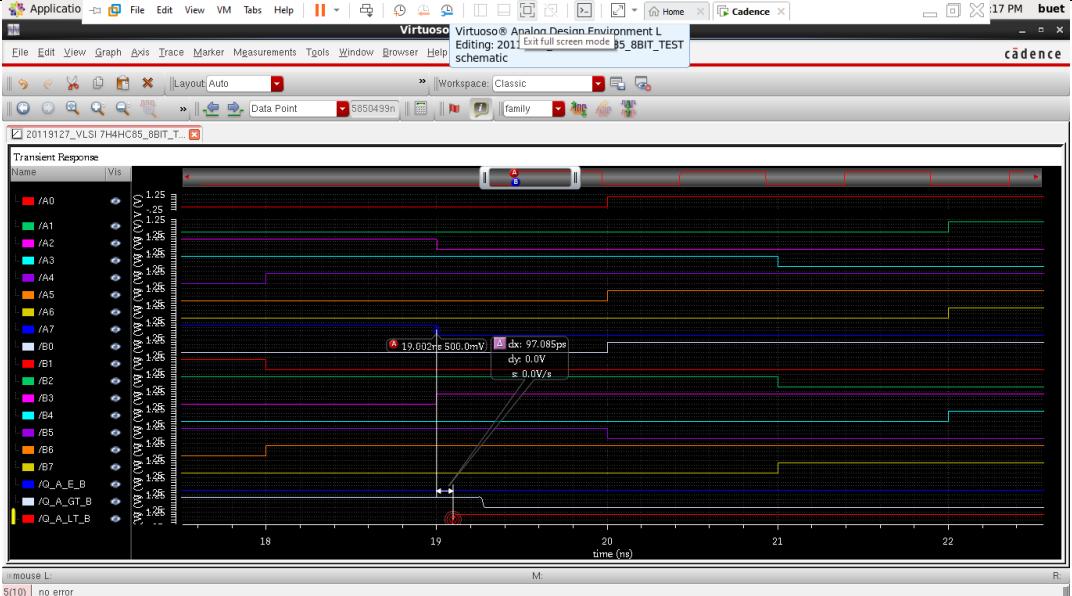
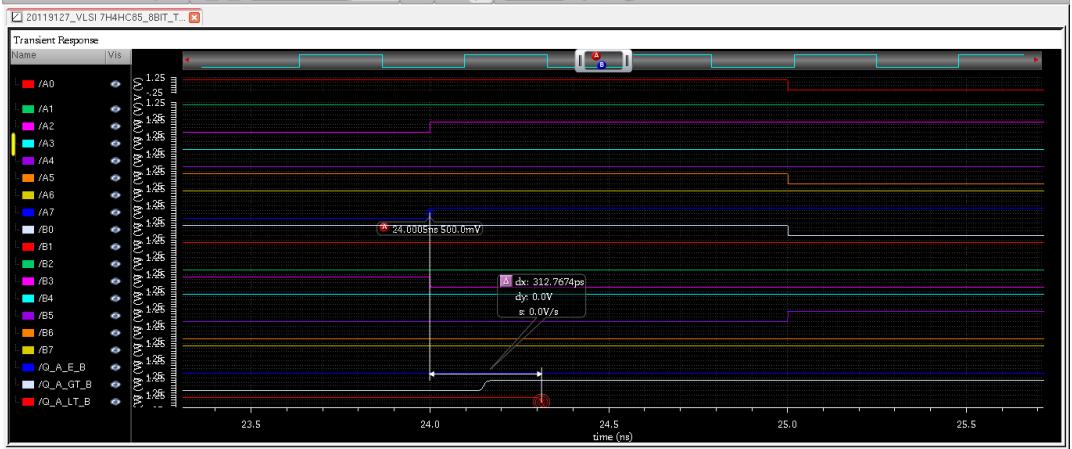
	
	
1.2 V	Low to High

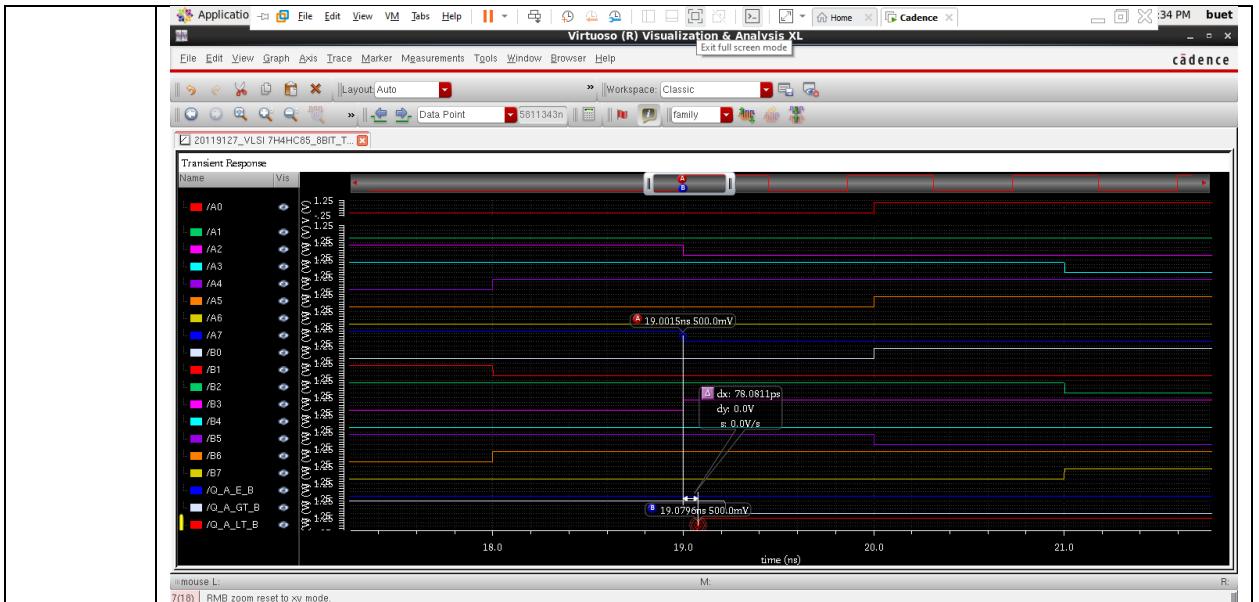


## High to Low



Voltage	A is less than B
Low to High	 <p>This screenshot shows a transient response plot in Virtuoso's Analog Design Environment. The plot displays multiple waveforms over time (ns). The legend lists various nodes: /A0, /A1, /A2, /A3, /A4, /A5, /A6, /A7, /B0, /B1, /B2, /B3, /B4, /B5, /B6, /B7, /Q_A_E_B, /Q_A_GT_B, and /Q_A_LT_B. Most waveforms are at 1.25V, except for /Q_A_LT_B which starts at 0.0V and rises to 1.25V. A cursor highlights a point on the /Q_A_LT_B waveform at approximately 19.1 ns, with a value of 500.0 mV. Another cursor highlights a point on the same waveform at approximately 18.0 ns, with a value of 500.0 mV. The plot title is "20119127_VLSI 7H4HC85_8BIT_T...".</p>
0.8 V	 <p>This screenshot shows a transient response plot in Virtuoso's Analog Design Environment. The plot displays multiple waveforms over time (ns). The legend lists the same nodes as the previous plot. In this case, most waveforms are at 1.25V, while /Q_A_LT_B starts at 1.25V and drops to 0.0V. A cursor highlights a point on the /Q_A_LT_B waveform at approximately 24.0 ns, with a value of 500.0 mV. Another cursor highlights a point on the same waveform at approximately 21.4 ns, with a value of 500.0 mV. The plot title is "20119127_VLSI 7H4HC85_8BIT_TEST".</p>
1.0 V	Low to High

	
	
1.2 V	Low to High



## High to Low



## CHAPTER 6 CONCLUSION & DEVELOPMENT

The 74HC85 is a 4-bit magnitude comparator IC, while an 8-bit comparator can be constructed by combining two 74HC85 ICs. By using the 74HC85 in pairs, a complete 8-bit binary comparison can be made, where the inputs are split into two groups of four bits each, and the outputs of each 74HC85 are combined to produce a result.

The 8-bit comparator constructed using two 74HC85 ICs can be used in various digital applications where binary comparison is required for 8-bit binary numbers, such as in digital signal processing, arithmetic operations, and data encoding. For example, it can be used to compare two 8-bit numbers in a microcontroller and control the flow of a program based on the comparison result.

One common application of the 8-bit comparator is in designing digital circuits that perform arithmetic operations on 8-bit numbers. In such circuits, the 8-bit comparator can be used to compare the two operands and determine their relative magnitudes, which is important for deciding the appropriate operation to be performed, such as addition or subtraction.

Another application of the 8-bit comparator is in data encoding, where it can be used to encode the relative magnitudes of two or more 8-bit binary numbers into a smaller number of bits. For example, in a priority encoder, the 8-bit comparator can be used to compare the binary values of several 8-bit inputs and output the highest-priority input as a binary code.

In conclusion, an 8-bit comparator constructed using two 74HC85 ICs is a versatile IC that can be used in various digital applications where binary comparison is required for 8-bit binary numbers. Its compact size, low power consumption, and fast operation make it a popular choice for many digital circuit designs.

# **CHAPTER 7 SEARCHING A NOVEL DESIGN OF 12-BIT DIGITAL COMPARATOR USING MULTIPLEXER FOR HIGH SPEED APPLICATION IN 32-NM CMOS TECHNOLOGY**

The paper discusses the importance of power consumption in digital CMOS circuits and the use of different methodologies for designing low-power circuits with small size and high-speed interfaces. It emphasizes the impact of wiring complexity on the area of an integrated circuit and the significance of selecting the right logic style for circuit performance.

The paper presents the design of a 12-bit comparator with low power consumption and improved packing densities using a multiplexer-based approach and a novel technique. The design is implemented in 32-nanometer technology with a supply voltage of 0.7 V, targeting applications such as Digital Signal Processing, Central Processing Unit, and Microcontroller.

The paper discusses the optimization of power consumption, speed, and area in CMOS comparators, considering the challenges posed by Moore's Law, which dictates the doubling of transistors in a chip every 18 months. Various logic techniques, such as typical CMOS, PLT, TG, GDI, and hybrid logic, have been used in different comparator designs with different transistor counts to achieve low power consumption, high speed, and small area.

The paper mentions specific examples of two-bit and one-bit comparators designed using different logic styles, such as PTL, GDI, and CMOS, with varying power consumption, area, and transistor counts. The proposed multiplexer-based two-bit comparator design is highlighted as having fewer transistors compared to other comparator designs using CMOS, PTL, and TG logic techniques.

The paper proposes a two-stage structure for designing a 12-bit comparator, where in the first stage, three 4-bit comparators based on multiplexers are used instead of a single 12-bit comparator to minimize the number of transistors and gate levels. Each 4-bit comparator produces individual outputs indicating whether

"A" is less than "B" or equal to "B". The output of a 4-bit comparator is determined using specific equations (8) and (9).

The second stage has been implemented by Equations (5–7) to compute the final outputs of a 12-bit comparator.

$$A < B := L_{0-3} + E_{0-3}(L_{4-7} + E_{4-7} * L_{8-11}) \quad (5)$$

$$A = B := E_{0-3}*E_{4-7}*E_{8-11} \quad (6)$$

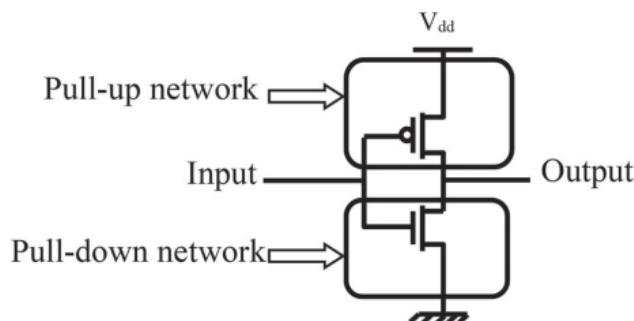
$$A > B := (A < B) * (A > B) \quad (7)$$

where  $L_{0-3}$ ,  $L_{4-7}$  and  $L_{8-11}$  represents that the signal A is less than the signal B in these comparator blocks. Terms  $E_{0-3}$ ,  $E_{4-7}$  and  $E_{8-11}$  shows that the signal A is equal to the signal B in these comparator blocks.

$$L_{0-3} = \overline{A}_0 B_0 + \sum_{K=1}^3 [\overline{A}_K B_K \prod_{M=K-1}^0 (A_m \odot B_m)] \quad (8)$$

$$E_{0-3} = \prod_{m=0}^3 (A_m \odot B_m) \quad (9)$$

The paper also proposes a way to optimize the 12-bit comparator circuit by using Transmission Gate Logic. The transmission gate logic technique is one of the most prominent technique to design a comparator, which has low power consumption and less transistor count than conventional CMOS logic-based comparator. In this technique one NMOS and one PMOS transistors are connected in parallel with complementary inputs at their gates..



*Figure 1 CMOS logic technique-based inverter.*

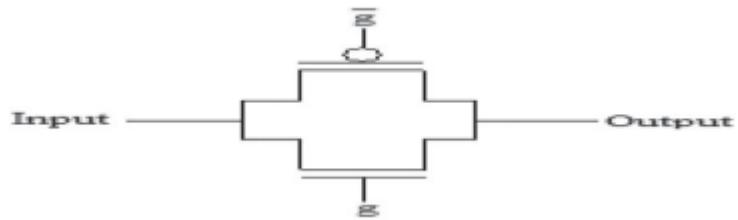


Figure 2 CMOS transmission gate

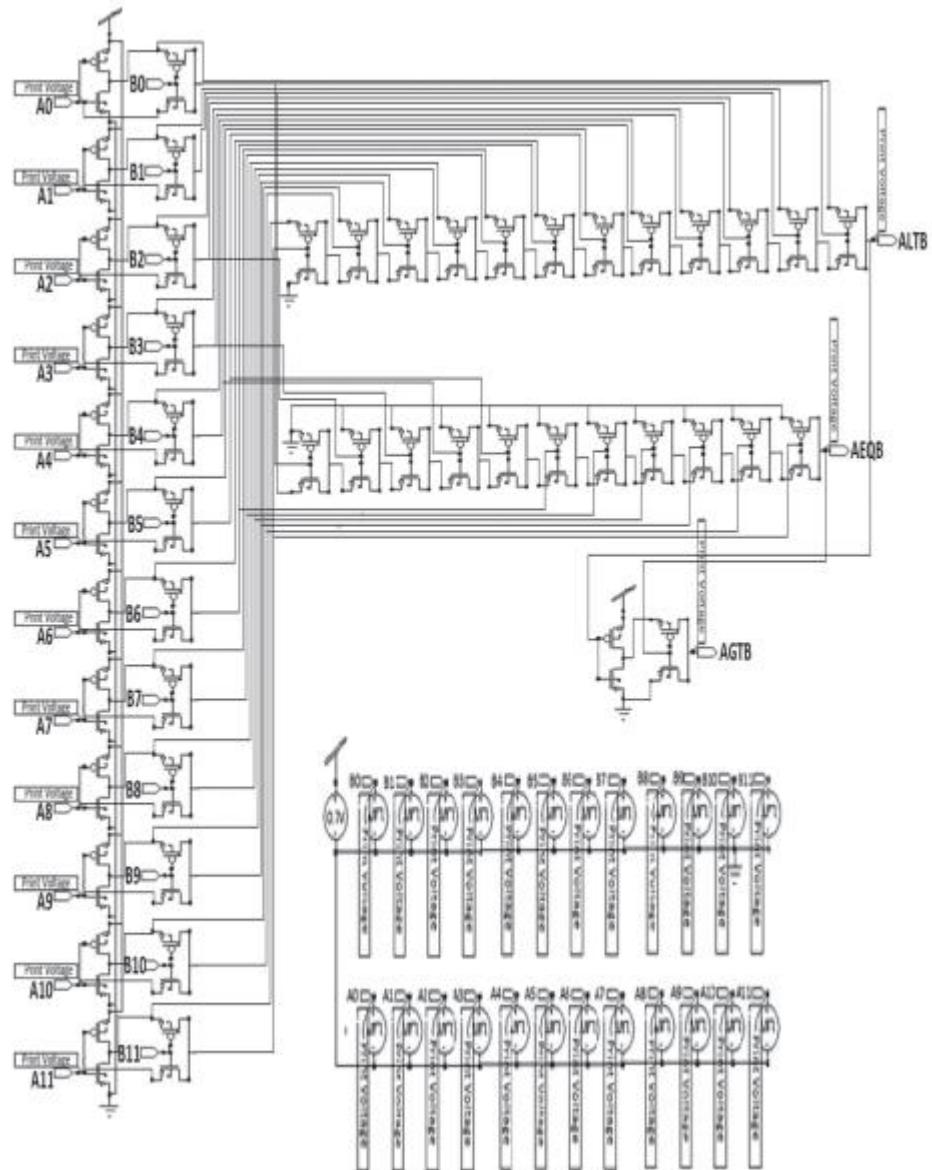
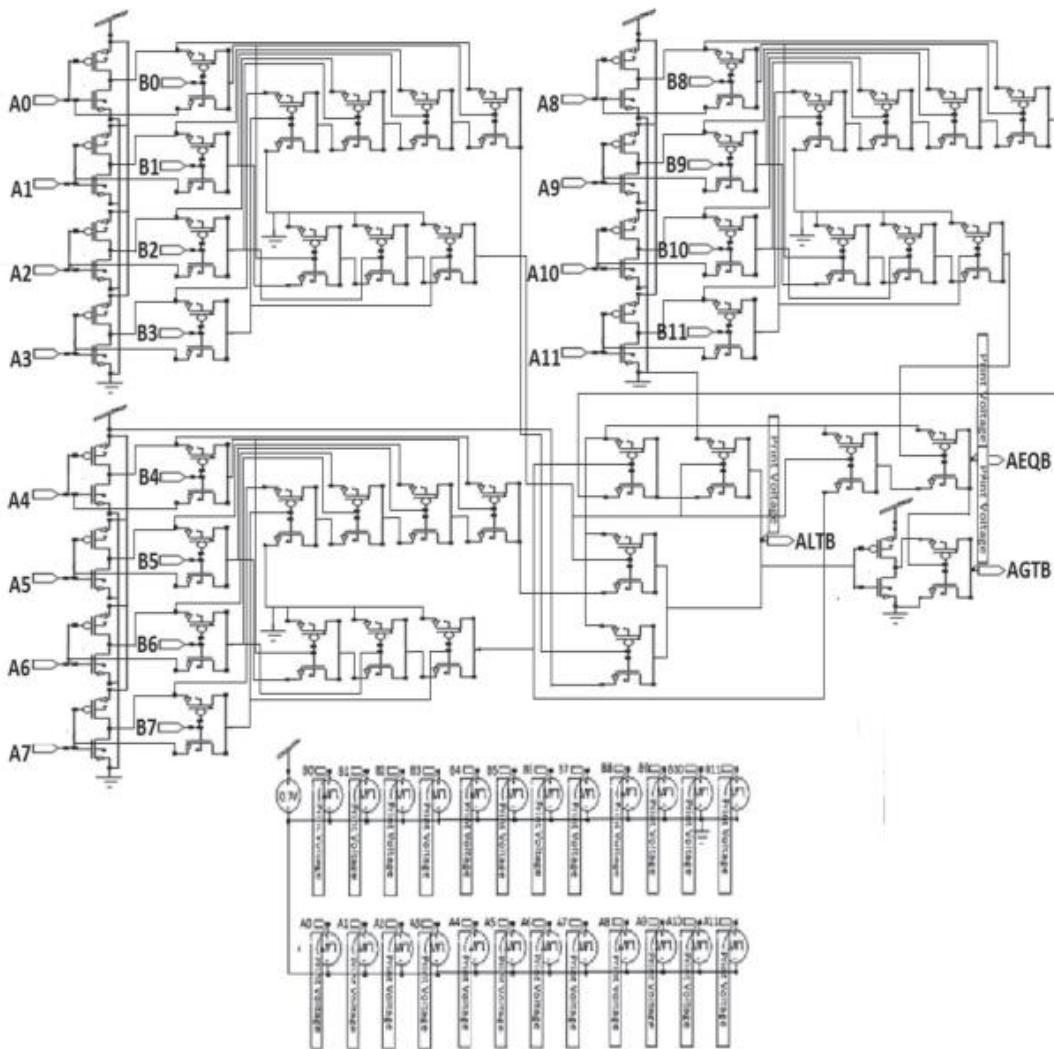


Figure 3 Schematic of the 12-bit digital comparator using a multiplexer.



*Figure 4 Schematic of 12-bit digital comparator using a novel technique.*

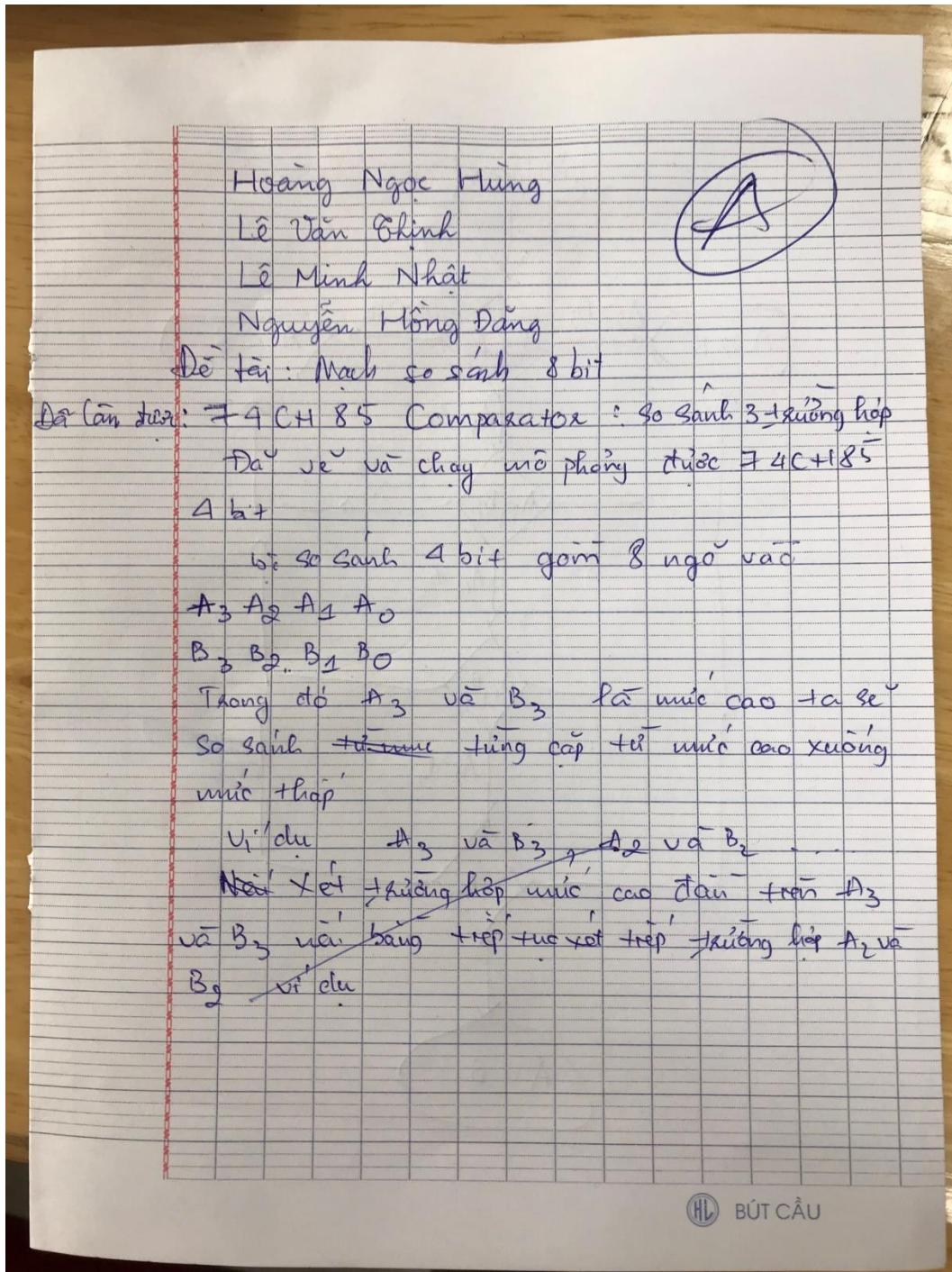
Simulation result from this novel design

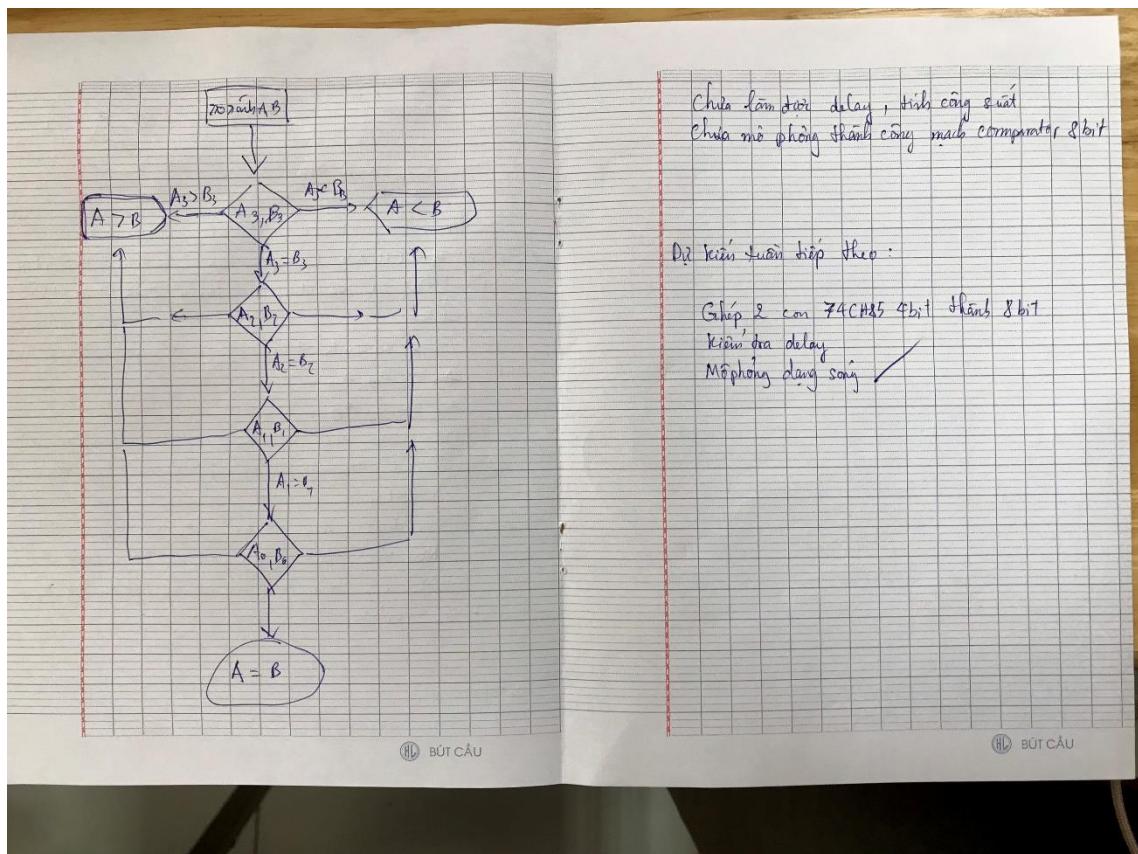
**Table 1: Performance comparison of a digital comparator.**

Parameters	Conventional 2-bit comparator	Transmission Gate logic [12] based 2-bit comparator	Transmission Gate logic [1] based 2-bit comparator	Multiplexer-based 2-bit comparator	Multiplexer-based 12-bit comparator	Novel technique-based 12-bit comparator
Power consumption ( $\mu\text{W}$ )	0.53	0.41	0.23	0.09	0.54	0.56
Number of transistor	54	74	30	18	98	106
Propagation delay (n-sec)	6.97	7.13	5.32	4.52	7.38	5.15
Power delay Product ( $\mu\text{-nJ}$ )	3.69	2.92	1.22	0.40	3.98	2.88

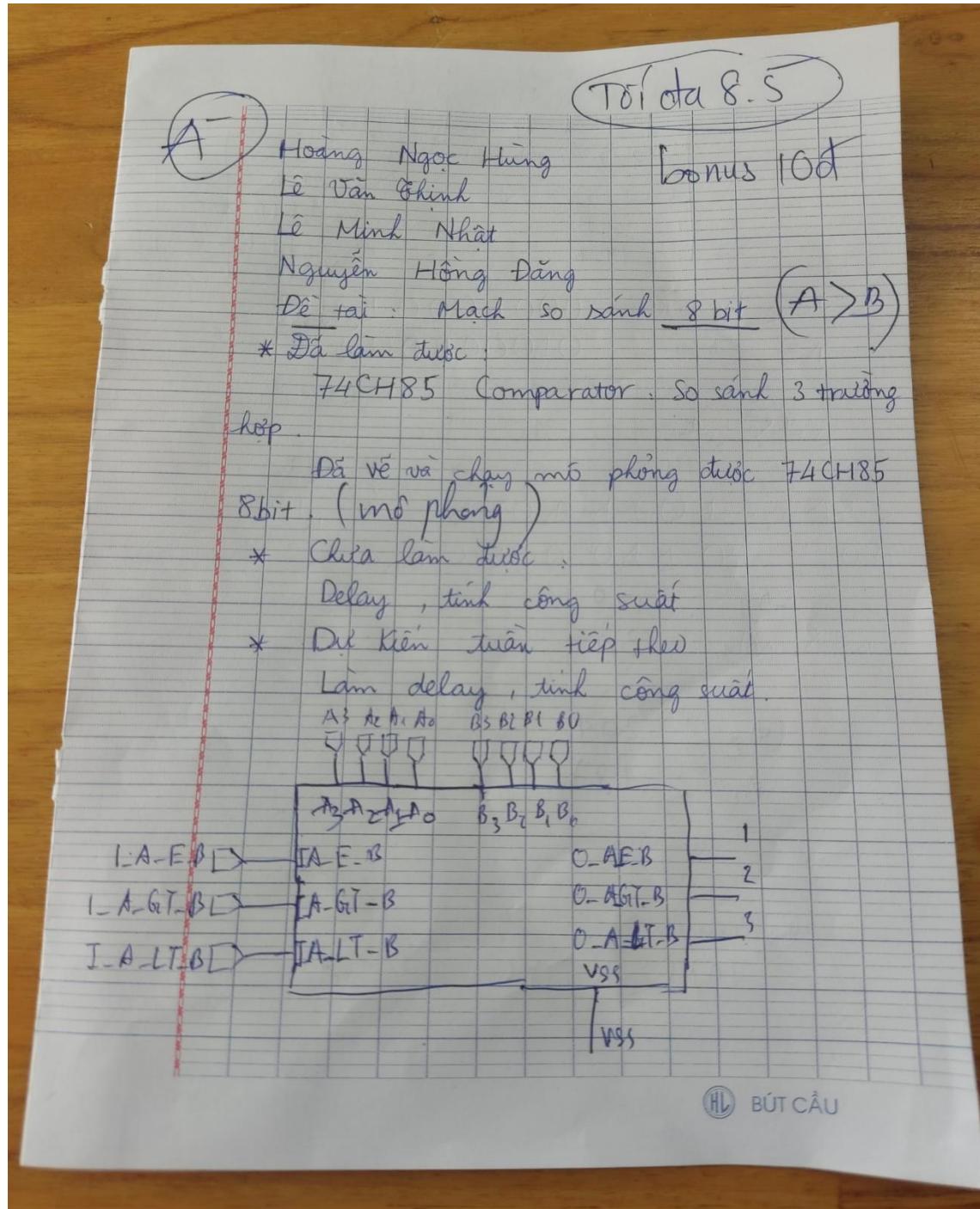
RESEARCH NOTE

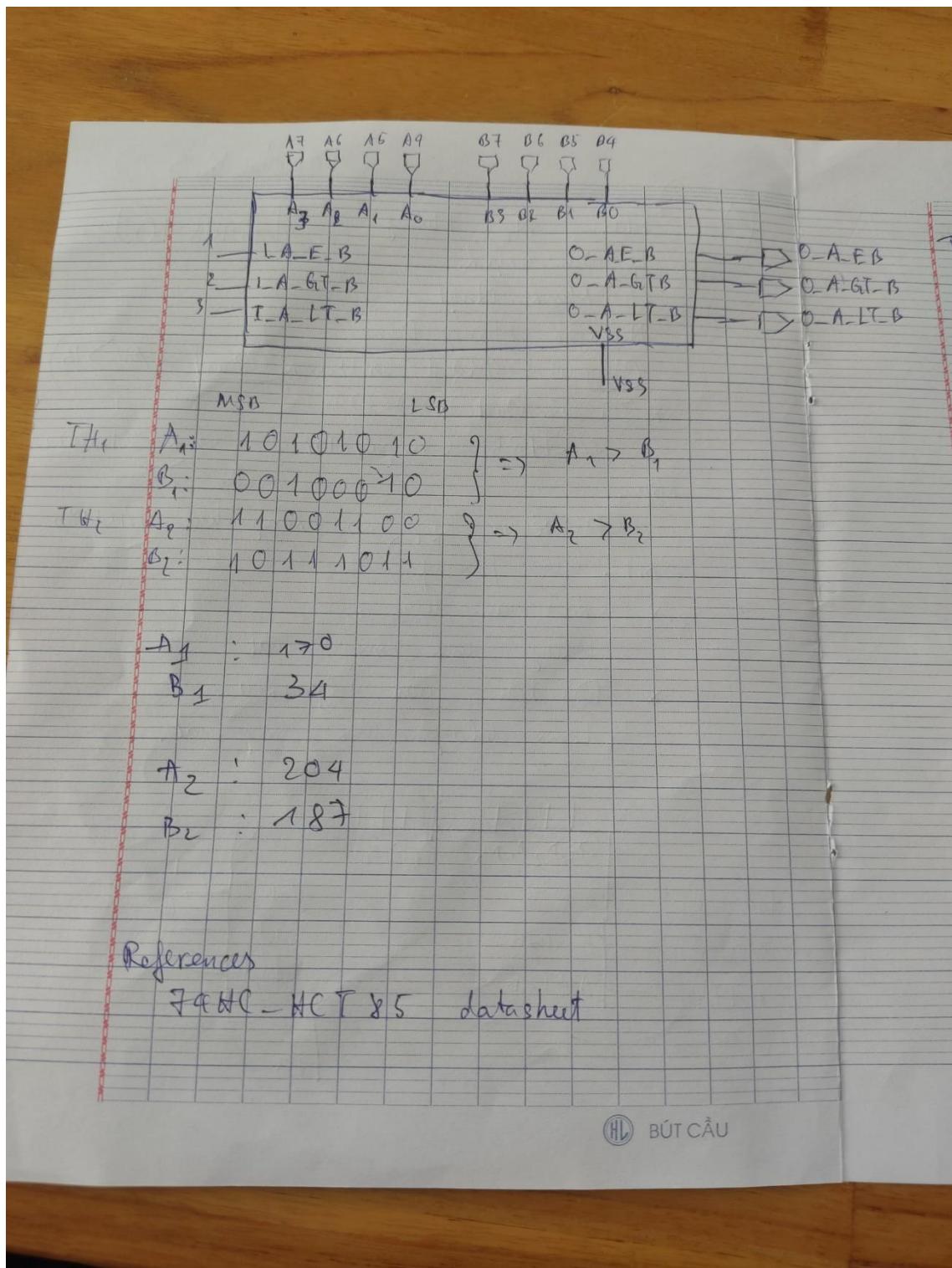
The first time

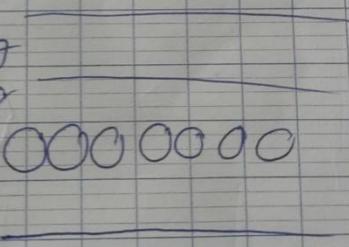




The second time

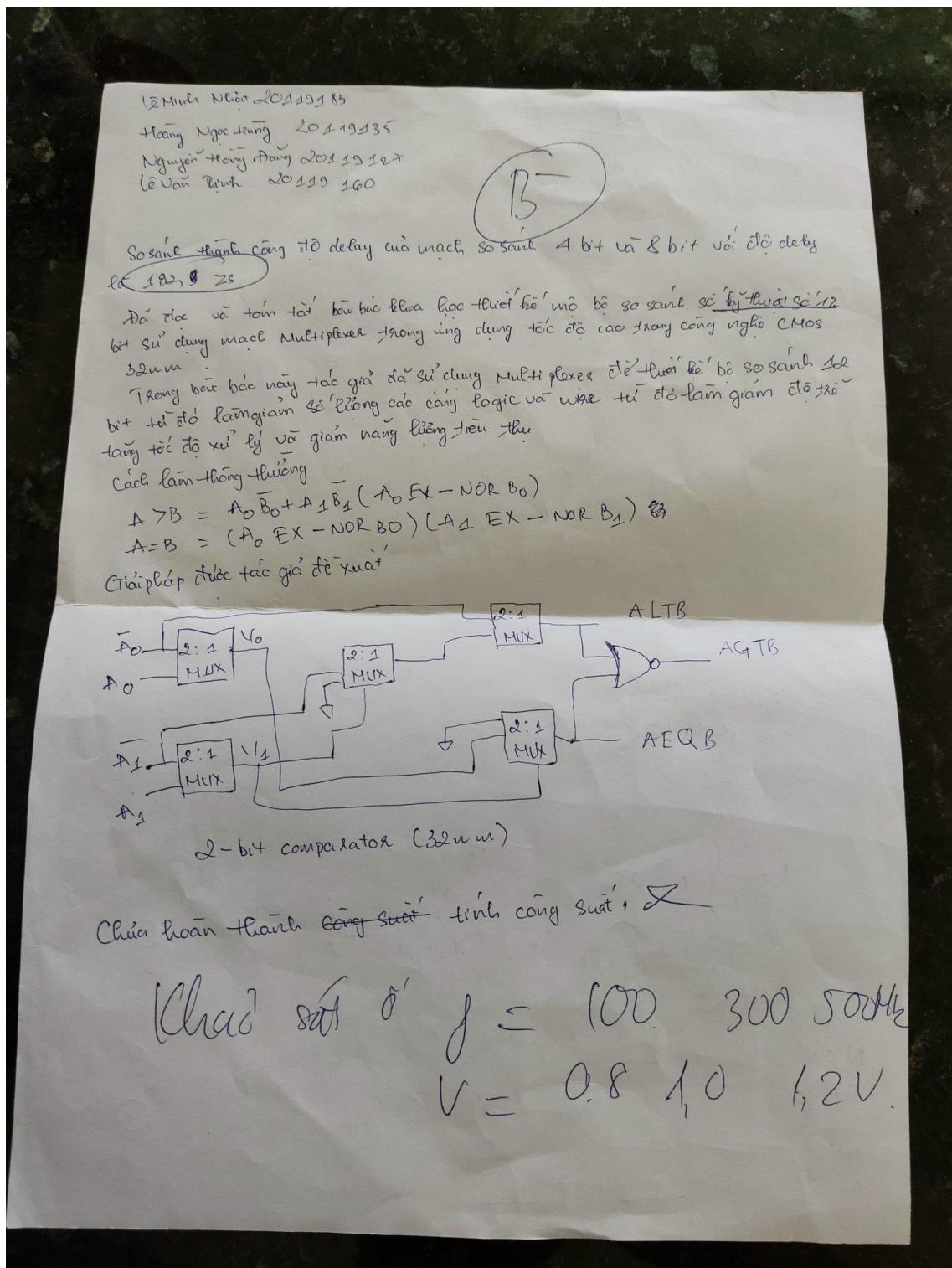




<ul style="list-style-type: none"> <li>→ O_A_EB</li> <li>→ O_A_GT-B</li> <li>→ O_A_LT-B</li> </ul>	$T_{H_3}$ <del><math>A_3 \quad 01010101</math></del> <del><math>B_3 \quad 01010101</math></del>	$T_{H_3} A_3 10001000$ $B_3 00100010$ $\qquad\qquad\qquad$ Decimal 136 $\qquad\qquad\qquad$ 34 $A_3 > B_3$
	$T_{H_4}$ <del><math>A_4 \quad 11011101</math></del> <del><math>B_4 \quad 01100110</math></del>	$A_4 > B_4$
	$A \quad A^7$ $B \quad B^7$ $\qquad\qquad\qquad$ 128 $O \quad$	

(HL) BÚT CẦU

### The third time



$$P_{\text{switching}} = d \times C \times V_{\text{dd}}^2 \times f$$

grain cog. logic gates (accessing) Memory

memory logic gates

$$\left\{ \begin{array}{l} d = 0,1 \\ C_{\text{total}} = 50 \cdot 10^6 \cdot 32 \cdot 1,025 \cdot 1,8 \cdot 10^{-9} \end{array} \right. = 2,7 \mu F$$

$$\left\{ \begin{array}{l} d = 0,02 \\ C_{\text{total}} = 350 \cdot 10^6 \cdot 1 \cdot 0,025 \cdot 1,8 \cdot 10^{-9} \end{array} \right. = 1,71 \mu F$$

$$P_{\text{switching}} = 0,02 \times 1,71 \times 10^{-9} \times 1^2 + 1 \cdot 10^{-9}$$

-

$$0,1 \times 2,7 + 1^2 \times 10^{-9}$$

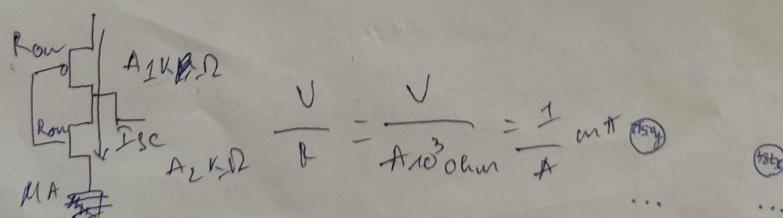
$$10^{-9} (C_0 \cdot 0,2 \times 1,1 \cdot 10^{-9} + 0,1 \times 2,7 \cdot 10^{-9})$$

$$= 6,1 \mu W$$

Power dissipation in CMOS

Dynamic dissipation

Static dissipation



$$\frac{V}{R} = \frac{V}{A \cdot 10^3 \text{ Ohm}} = \frac{1}{A} \text{ mW}$$

(85a)

...

...

(85b)

...

...

(85c)

...

...

(85d)

...

(85e)

...

(85f)

...

(85g)

...

(85h)

...

(85i)

...

(85j)

...

(85k)

...

(85l)

...

(85m)

...

(85n)

...

(85o)

...

(85p)

...

(85q)

...

(85r)

...

(85s)

...

(85t)

...

(85u)

...

(85v)

...

(85w)

...

(85x)

...

(85y)

...

(85z)

...

(85aa)

...

(85ab)

...

(85ac)

...

(85ad)

...

(85ae)

...

(85af)

...

(85ag)

...

(85ah)

...

(85ai)

...

(85aj)

...

(85ak)

...

(85al)

...

(85am)

...

(85an)

...

(85ao)

...

(85ap)

...

(85aq)

...

(85ar)

...

(85as)

...

(85at)

...

(85au)

...

(85av)

...

(85aw)

...

(85ax)

...

(85ay)

...

(85az)

...

## REFERENCES

1. Neil H. E. Weste, David Money Harris, “*CMOS VLSI Design - A Circuits and Systems Perspective, Fourth Edition*”.
2. “*2:1 MUX using transmission gate*”  
<https://www.electronics-tutorial.net/Digital-CMOS-Design/Pass-Transistor-Logic/2-1-MUX-using-transmission-gate/>
3. “*Pass – Transistor - Logic*”  
<https://www.electronics-tutorial.net/Digital-CMOS-Design/Pass-Transistor-Logic/2-1-MUX-using-transmission-gate/>
4. D. N. Mukherjee, S. Panda & B. Maji, “*A Novel Design of 12-bit Digital Comparator Using Multiplexer for High Speed Application in 32-nm CMOS Technology*”, 08 Aug 2019.
5. Geetanjali Sharma, Uma Nirmal, Yogesh Misra, “*A Low Power 8-bit Magnitude Comparator with Small Transistor Count using Hybrid PTL/CMOS Logic*”, *IJCEM International Journal of Computational Engineering & Management*, Vol. 12, April 2011.