CmpE 124 Lab 4: Comparator

Hoang Le, 012679771, CMPEs 124 Spring 2019, Lab Section 03

***Abstract* —** This purpose of this lab is designing signed and unsigned the comparator circuit in 2 bits and 4 bits.

**INTRODUCTION**

The lab manual is providing input and initial condition in 2-bits and 4-bits to design a circuit which is about comparator in larger or equal circuit

.

# Design methodology

For 2-bits comparator, it’s unsigned number. Starting the design by creating the truth table with 16 possible outcomes. Then, creating K-map based on the true table to get simplicity function for 2-bits comparator and build the circuit with 74LS04 and 74LS10

For 4-btis comparator equal and greater than, it’s signed number. It’s more complicated than the two-bits but we do need to get the function which is compare the MSB value first to determine the 2 signed number are greater or equal then moving on to the next value with the similar process. Considering the result with 4 comparing cases : equal, positive vs positive, positive vs negative, and negative vs negative.

## Parts List

* 4 SN74HC04N
* 1 SN74HC86N
* 1 SN74HC63N
* 1 SN74HC163N
* 2 SN74HC10N
* 1 Crystal FS10.00P
* 8 1KΩ Resistors
* 1 100pF capacitor

## True table

**Table I**: Two-bit greater than

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| B1 | B0 | A1 | A0 | A(decimal) | B(decimal) | Z(a>b) | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | |
| 0 | 0 | 1 | 0 | 2 | 0 | 1 | |
| 0 | 0 | 1 | 1 | 3 | 0 | 1 | |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | |
| 0 | 1 | 1 | 0 | 2 | 1 | 1 | |
| 0 | 1 | 1 | 1 | 3 | 1 | 1 | |
| 1 | 0 | 0 | 0 | 0 | 2 | | 0 |
| 1 | 0 | 0 | 1 | 1 | 2 | | 0 |
| 1 | 0 | 1 | 0 | 2 | 2 | | 0 |
| 1 | 0 | 1 | 1 | 3 | 2 | | 1 |
| 1 | 1 | 0 | 0 | 0 | 3 | | 0 |
| 1 | 1 | 0 | 1 | 1 | 3 | | 0 |
| 1 | 1 | 1 | 0 | 2 | 3 | | 0 |
| 1 | 1 | 1 | 1 | 3 | 3 | | 0 |

## Kmap

**Table IV:** Kmap of two-bit greater-than design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a1a0  b1b0 | 00 | 01 | 11 | 10 |
| 00 | 0 | 1 | 1 | 1 |
| 01 | 0 | 0 | 1 | 1 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 1 | 0 |

## Original and Derived Equations

* From K-map, we have

Equation 1: = a0b1’b0’+ a1b1’+ a0a1b0’

* By compare bitwise, we have

E = e3e2e1e0

a3b3+a3’b3’

a2b2+a2’b2’

a1b1+a1’b1’

a0b0+a0’b0’

* Equation 2: The equal function

E = (a3b3+a3’b3’) x (a2b2+a2’b2’) x

(a1b1+a1’b1’) x (a0b0+a0’b0’)

* By compare bitwise, we have

G =

a3’b3

e3b2’a2

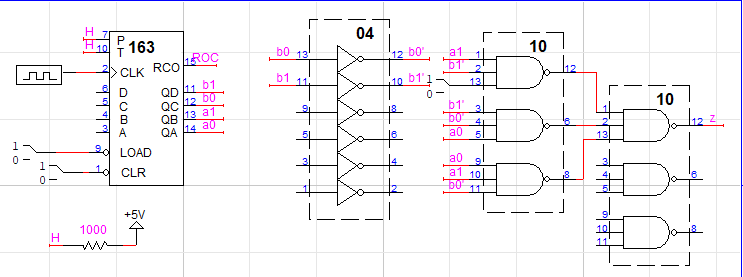
e3 e2b1’a1

e3 e2 e1b0’a0

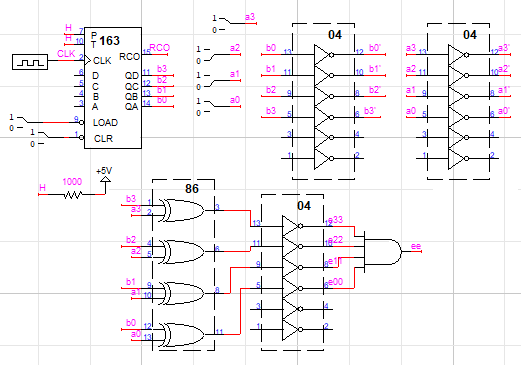
* Equation 3: The greater than function

G = a3’b3 + e3b2’a2e3 e2b1’a1 + e3 e2 e1b0’a0

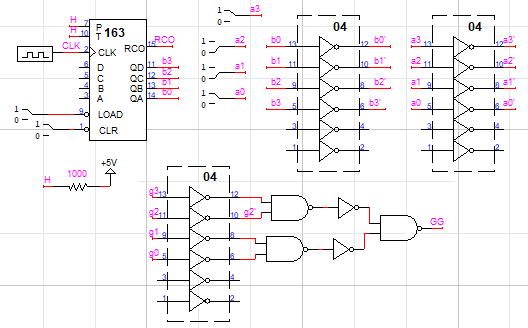
## Schematics



**Figure 1**: The design with two-bits greater than



**Figure 2**: The design with four-bits equal



**Figure 3**: four-bit greater-than circuit design

# testing procedures

Circuit 1:

1. Use output of 163 to the circuit in figure 1.
2. Connecting QD, QC, QB, QA to b1, b0, a1, a0 respectively.
3. Observer the waveform of z in the scope and LogicWork.
4. Compare the waveform between Logic Work and the scope

Circuit 2:

1. Follow the schematic in figure 2 to build the circuit.
2. Connecting QD, QC, QB, QA to b3, b2, b1, b0 respectively. Using dip switch to generate high and low input for a3, a2, a1, a0.
3. Observer when a3210 = LHLH

= HHLL

= HLLH

and record waveforms

Circuit 3:

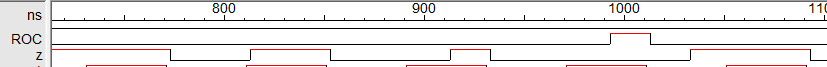
1. Follow the figure 3 to build the circuit.
2. Connecting QD, QC, QB, QA to b1, b0, a1, a0 inputs in order. Using dip switch to generate high and low input for aj.
3. Observer when a3210 = LHLH

= HHLL

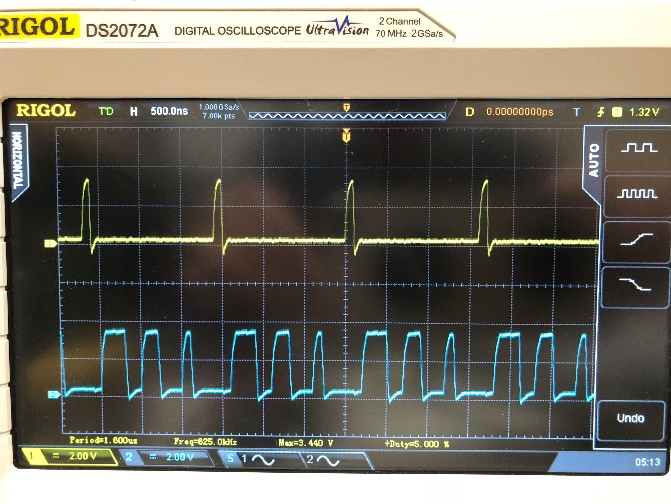
= HLLH

and record waveforms

# testing results



**Figure 4**: The wave form of Roc and output z from Logicwork.



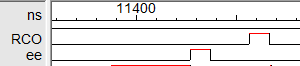
**Figure 5**: The wave forms of Roc (yellow) and output z (blue) from Oscilloscope.



**Figure 6**: The wave form of Roc and output of equal function (e) when a3210 = LHLH from Logicwork.



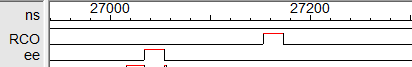
**Figure 7**: The wave form of Roc and output of equal function (e) when a3210 = LHLH from Oscilloscope.



**Figure 8**: The wave form of Roc and output of equal function (e) when a3210 = HHLL from Logicwork.



**Figure 9**: The wave form of Roc and output of equal function (e) when a3210 = HHLL from Oscilloscope.

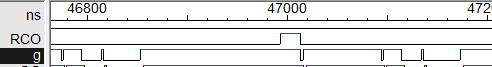


**Figure 10**: The wave form of Roc and output of equal function (e) when a3210 HLLH from Logicwork.

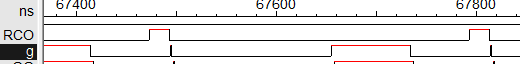


**Figure 11**: The wave form of Roc and output of equal function (e) when a3210 = HLLH from Oscilloscope.

Note: the magnitude of equal function is not as high as Roc from oscilloscope screens. I have not figured it out the reason for this but, the shapes seem match with the wave forms on Logicwork simulator.



**Figure 12**: The wave form of Roc and output of greater than function (g) when a3210 = LHLH from Logicwork.



**Figure 13**: The wave form of Roc and output of greater than function (g) when a3210 = HHLL from Logicwork.



**Figure 14**: The wave form of Roc and output of greater than function (g) when a3210 = HLLH from Logicwork.

Note: My circuit may have errors which does not imply the correct wave forms compare to the wave forms from Logic work. Until now, I have not figured out what are the errors in my circuit.

# Conclusion

Throughout the lab, student gain more experiment and understand about 2-bit greater than and the equal and greater than for 4-bit. They are both building successful.

The four-bit comparison greater than is better building by using the XOR gate because it could save us on money and simply the circuit and easy for student to observe the output by the waveform but does not lost its characteristic.

1. appendices and References

Khalil A.Estell , Mikko Bayabo, Harmander Sihra, Haluk Ozemek. CMPE124 Digital Design I, Course Manual. Khalil A.Estell,