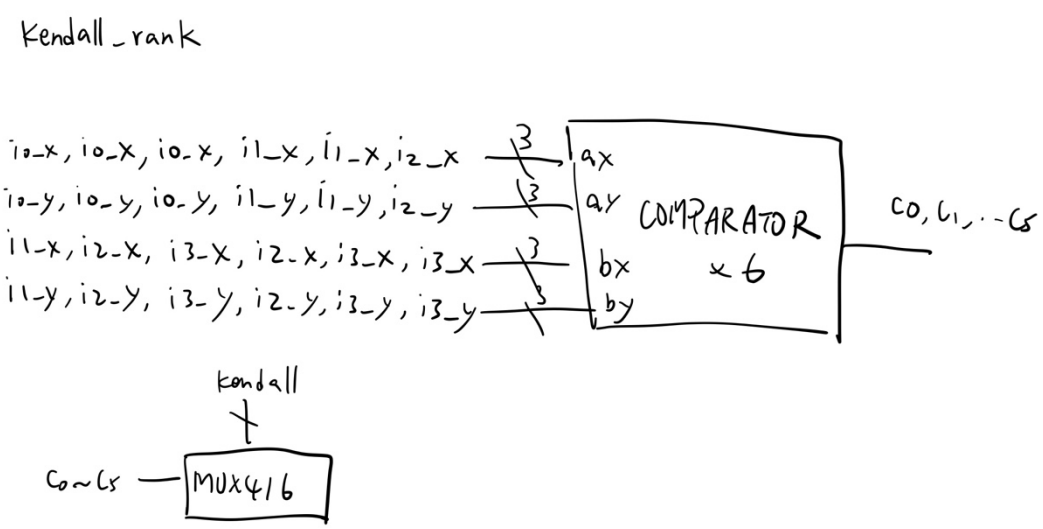
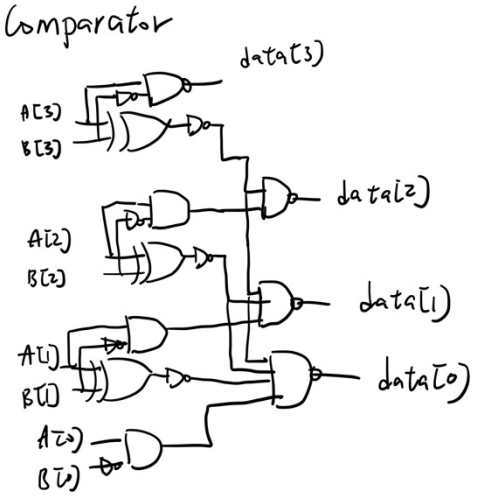
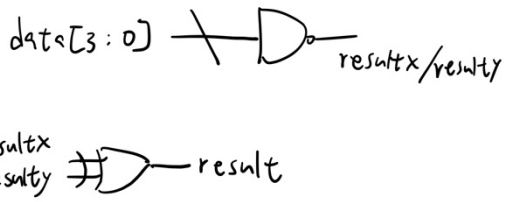
2022ICD HW3 report

B09901089 黃柏穎

Part1.

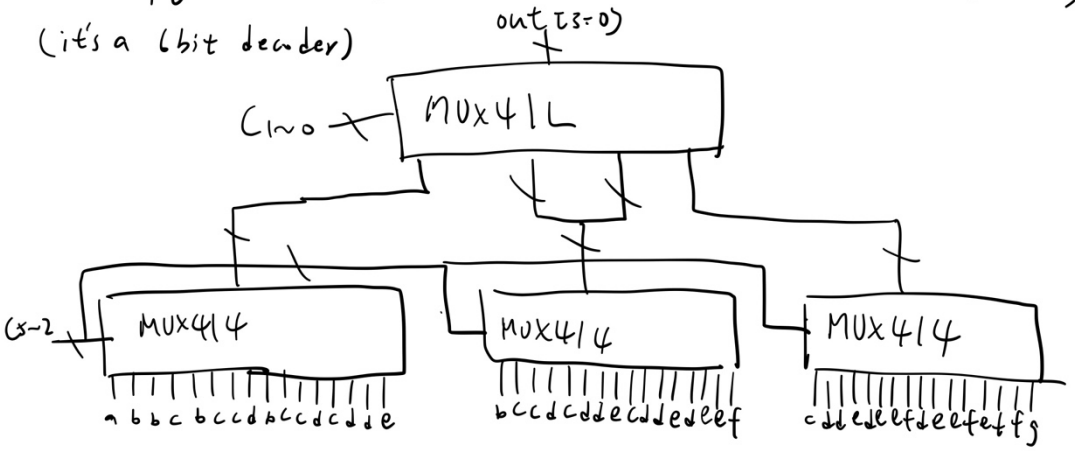






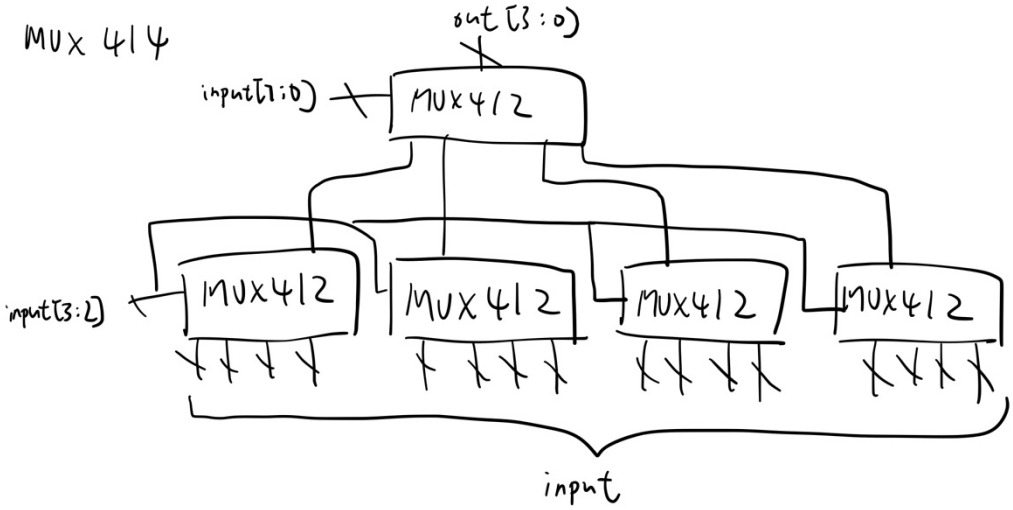
MUX 416 (a= 0100 b= 0011 c= 0001 d=0000 e= 1111 f=1101 g= 1100)

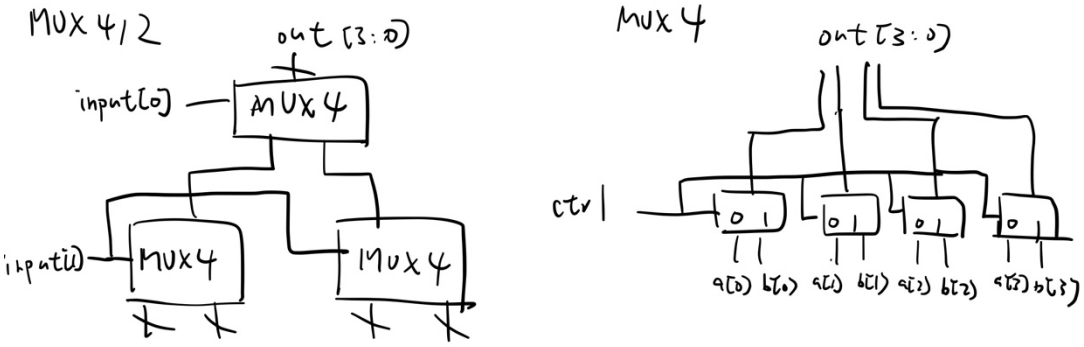
(it's a 6bit decoder)



it's

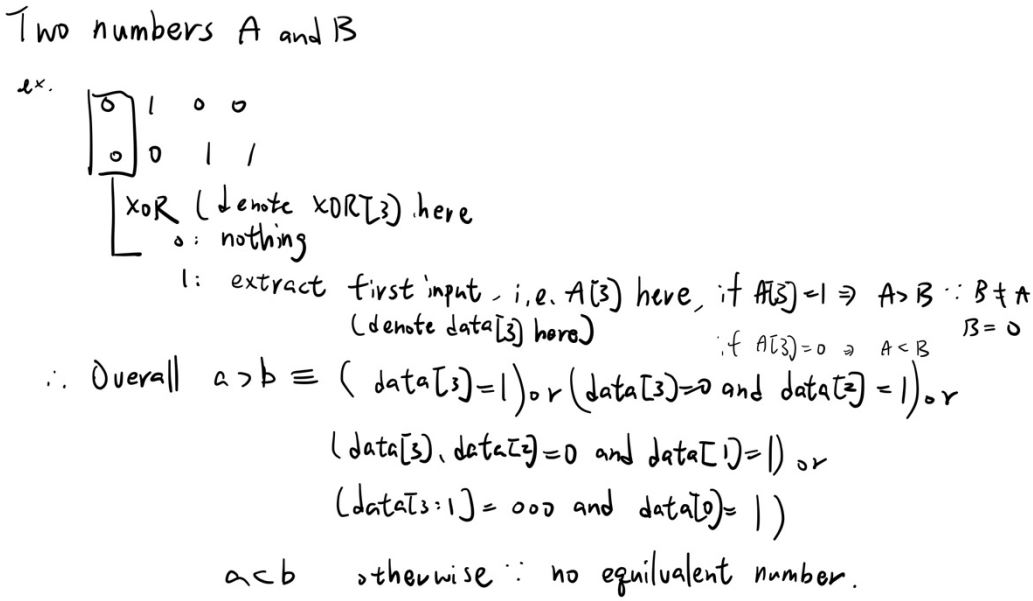
MUX 414





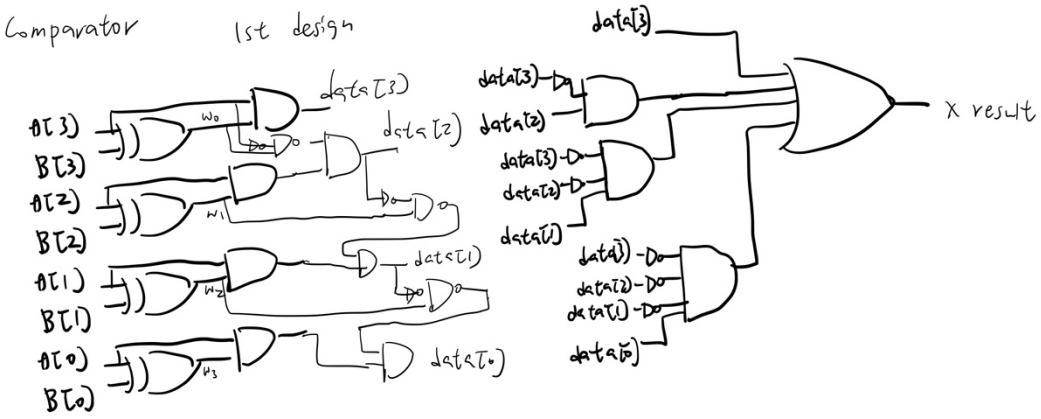
Part2.

To design an comparator, one naïve thought is simply using a 4bits substractor. But the performance of concatenated adder is way too bad. Thus I figured out a way to design comparator without using adder, see following figure.

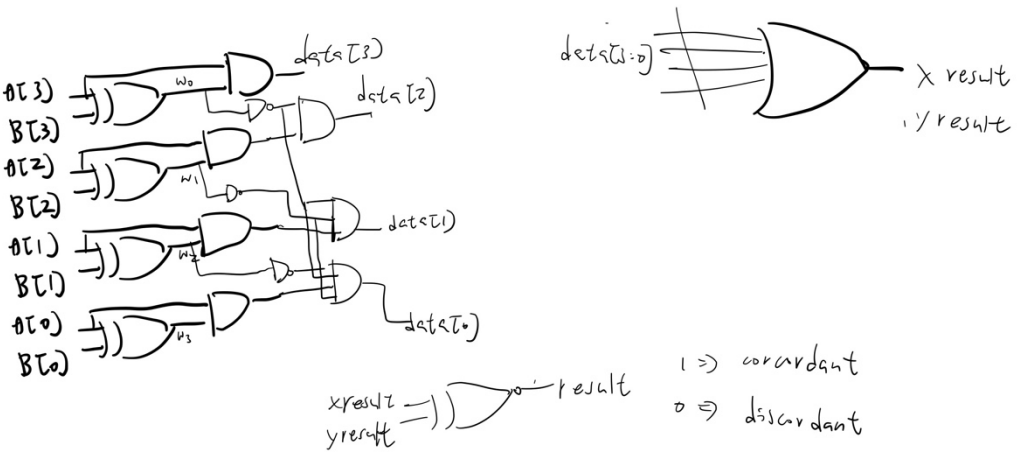


I subsequently made my first design, the core idea is that once the XOR output of higher bit turned out to be 1, It means two numbers can be distinguish in that bit, so the circuit will set the output bits of lower bits to zero, which is like masking them

because their result is meaningless.



Then I removed some redundant circuit and have a more concise version.



But the performance wasn't good enough, after discussed with classmates, I realized that we can take inversion of b instead of a xor b as the second parameter of the 4 two input and gate, and can change the gate used to improve. In the end designed the final version.

As for the 6bit decoder, I tried to add up the concordant pairs (the result ranging

from 0 to 6), then decode the three bits result at first. However, the performance is the same problem of design using adder. I realized that the last step is actually a 6 bit decoder, we need to categorize 64 possibility into 7 cases, and I don’t think there exists method that not using adder other than directly construct a binary decision tree. And I implement it by construct MUXs bigger and bigger to prevent too much wires.