# 2020302111389\_ 王楚 \_第六次作业

## p3

UDP and TCP use 1s complement for their checksums. Suppose you have the following three 8-bit bytes: 01010011, 01100110, 01110100. What is the 1s complement of the sum of these 8-bit bytes? (Note that although UDP and TCP use 16-bit words in computing the checksum, for this problem you are being asked to consider 8-bit sums.) Show all work. Why is it that UDP takes the 1s complement of the sum; that is, why not just use the sum? With the 1s complement scheme, how does the receiver detect errors? Is it possible that a 1-bit error will go undetected? How about a 2-bit error?

11010001

It is very convenient to use reverse code to connect to the receiver. You just need to add up all the data including the check code and calculate the sum as all ones.  
If it's not all ones, something's wrong. Therefore, in order to detect errors, the receiver should contain four words (three primitive words and a checksum) and be able to detect all one-bit errors, but not necessarily two-bit errors.

## p8

Draw the FSM for the receiver side of protocol rdt3.0

