Problem Statement

Fig.1 shows the top level block diagram. In this lab we will use the priority arbiter which we completed in last lab with few modification and will add few extra blocks to demonstrate the schedulling.

Instead of assigning weights in the priority arbiter now we have a different module named Weight Assignment. Based on the *Req* input, priority arbiter will generate the grant (gnt) signal. This *gnt* signal is fed to Weigth Assignment and Virtual Output Queue modules.

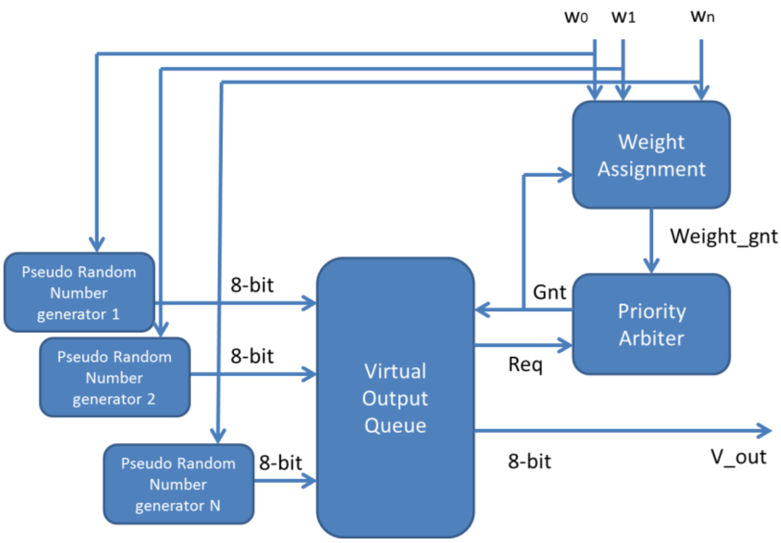

Figure 1 System Block Diagram

Based on *gnt* signal Weight Assignment module will make *Weight_gnt* signal high for number of clock period equal to expected weight of the corrsponding request. w0, w1,…wn are the weight input to the Weight Assignment module corresponding to each of *req* signal i.e for req0 weight is w0, for req1 weight is w1 and so on.

Pseudo Random number generator block will generate 8-bit pseudo random number and generated number will be based on the weight corresponding to that pseudo random number generator block. i.e block1 will generate w0 pseudo random numbers in n cycles, block2 will generate w1 pseudo random numbers in n cycle and so on.

Virtual Output Queue (VOQ) block get the input from *n* pseudo random number generator blocks and stores them in *n* FIFO, one corresponding to each pseudo random number generator block. VOQ block generate n *req* signal one from each FIFO. Req signal become high only when there is data in FIFO, otherwise it remians LOW. Whenever VOQ get a gnt signal for a FIFO, for every cycle *gnt* is HIGH it will read a data from the respective FIFO and gives as output to v_out.

Note: You may add extra interface signals to blocks in addition to interface signal shown in fig.1.