1. Design a universal fire hydrant system that has the following properties:
   - When a fire hose is attached, the hydrant valve should automatically open.
   - The hydrant needs to be completely filled with water before its valve can open. (Assume it takes 1 cycle for the hydrant to completely fill)
   - The hydrant should preemptively fill with water if the temperature is at least 5°C.
   - Otherwise, the hydrant should be empty.
   - The fire hydrant should also track how much water is used while valve is open.

Inputs:
- [1 bit] hose: 1 if fire hose is connected, 0 otherwise
- [32 bit] temp: outdoor temperature, in °Celsius

Outputs:
- [1 bit] valve: 1 will open valve, 0 will close it
- [1 bit] fill: 1 will fill fire hydrant, 0 will empty it
- [32 bit] water: how much water was used, in # cycles

Please provide HLSM, datapath, and controller FSM.

2. Convert the following C code to HLSM:
   Inputs: byte a[256], byte b[256], byte cy
   Outputs: byte sumx, byte sumy, byte c[256]

MULT_ADD:
int i = 0;
int sumx = 0;
Int sumy = 0;
while( i < 256 ) {
    if( a[i] > 128 ){
        c[i] = a[i] * b[i]
        sumx = sumx + c[i]
    }else{
        c[i] = a[i] * (b[i] + cy);
        sumy = sumy + c[i];
    }
    i++;
}
3. Design a data-dominated system that computes and outputs the average of the absolute values of 4 separate 32-bit signed numbers which are provided as external data inputs (R0, R1, R2, R3). The computation of the average should be done in a single equation in a single state. The computation should be performed when a single-bit input \textit{go} changes from 0 to 1 and the computed result should be held at the output until the next time \textit{go} changes again from 0 to 1.

Show the following:
   A) HLSM
   B) Datapath
   C) Connect datapath and control

4. Use the RTL design process to design a system that outputs a multi-bit data number based on the following.

The system has a 16-bit unsigned data input \textit{G} and an unsigned output \textit{B}. The data input is sampled and stored in a local register when a single bit input \textit{S} changes from 1 to 0. Right after sampling a new value, the system increases the stored value by 1 at each clock cycle as long as \textit{S} is zero. If \textit{S} is switched back to 1, the system waits. At each clock cycle, the output of the system is:

\[
\text{stored_input + previous_output}[3:0] \ll 4
\]

Where \text{previous_output}[3:0] are the 4 least significant bits of the previous output.
Choose and report the internal bit widths that prevent overflow and present the following in your solution:
   a. Create a datapath and clearly label the control signals of your components.
   b. Connect the datapath to a controller and report the labels.
   c. Derive the controller’s FSM diagram.