Problem 1 (5 points)

Draw a block diagram of the execution unit of a circuit capable of storing a set of 128 8-bit signed integers in an internal memory, and then computing the largest and the second largest number in this set and their respective positions within the set. In parallel, the circuit should also compute an average of all 128 numbers stored in the memory.

The circuit should be able to execute the following pseudocode, and should be optimized for minimum latency (i.e., execute as many operations as possible in parallel):

```plaintext
while (s=0)
    initialize internal memory MEM
end while

largest = MIN_SIGNED
second_largest = MIN_SIGNED
largest_pos = 0
second_largest_pos = 0
sum = 0

for i=0 to 127 do
    sum = sum + MEM[i]

    if MEM[i] >= largest then
        second_largest = largest
        second_largest_pos = largest_pos
        largest = MEM[i]
        largest_pos = i
    elseif (MEM[i] >= second_largest) then
        second_largest = MEM[i]
        second_largest_pos = i
    end if

end for

average = sum / 256
```

*Please clearly mark the widths of all buses in your circuit.*
Assume the following interface to your circuit:

<table>
<thead>
<tr>
<th>Port</th>
<th>Width</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk</td>
<td>1</td>
<td>System clock</td>
</tr>
<tr>
<td>Reset</td>
<td>1</td>
<td>System reset – clears internal registers. Active high.</td>
</tr>
<tr>
<td>DataIn</td>
<td>8</td>
<td>Input data bus</td>
</tr>
<tr>
<td>RAdd</td>
<td>7</td>
<td>Address of the internal memory MEM where input data is stored</td>
</tr>
<tr>
<td>WrInit</td>
<td>1</td>
<td>Synchronous write control signal</td>
</tr>
<tr>
<td>s</td>
<td>1</td>
<td>Operating mode: 0 = initialization, 1 = computations.</td>
</tr>
<tr>
<td>Rd</td>
<td>1</td>
<td>Read enable. 0 = high impedance on all output buses, 1 = valid outputs.</td>
</tr>
<tr>
<td>average</td>
<td>8</td>
<td>Average of all 128 numbers</td>
</tr>
<tr>
<td>largest</td>
<td>8</td>
<td>Largest number in the input set</td>
</tr>
<tr>
<td>second_largest</td>
<td>8</td>
<td>Second largest number in the input set</td>
</tr>
<tr>
<td>largest_pos</td>
<td>7</td>
<td>Position of the largest number in the input set</td>
</tr>
<tr>
<td>second_largest_pos</td>
<td>7</td>
<td>Position of the second largest number in the input set</td>
</tr>
<tr>
<td>Done</td>
<td>1</td>
<td>Asserted when all results are ready, zero otherwise.</td>
</tr>
</tbody>
</table>

**Problem 2 (5 points)**

1. Draw a block diagram of a circuit based on the PicoBlaze microcontroller capable of performing computations described in Problem 1.
2. Write a program in the assembly language of PicoBlaze corresponding to the following portion of the pseudocode from Problem 1.

```plaintext
for i=0 to 127 do
  if MEM[i] >= largest then
    second_largest = largest
    second_largest_pos = largest_pos
    largest = MEM[i]
    largest_pos = i
  elseif (MEM[i] >= second_largest) then
    second_largest = MEM[i]
    second_largest_pos = i
  end if
end for
```
Problem 3 (2.5 points)

For a given below circuit implementing sorting of \( k = 2^L = 256 \) N-bit numbers, determine values of all control signals listed in the table during the execution of the instructions given in the first column of the table. Values in the last four columns should correspond to the values of the corresponding signals before the instruction takes effect, and may be marked as UN (“unknown”). Clearly distinguish between a “don’t care” (“-”) value and an “inactive” (“0”) value of the control signals.

<table>
<thead>
<tr>
<th>Instruction</th>
<th>WrInit</th>
<th>Wr</th>
<th>Rd</th>
<th>Bout</th>
<th>Int</th>
<th>Csel</th>
<th>Ain</th>
<th>Bin</th>
<th>LI</th>
<th>EI</th>
<th>LJ</th>
<th>EJ</th>
<th>RAdd</th>
<th>Ci</th>
<th>Cj</th>
</tr>
</thead>
<tbody>
<tr>
<td>M[10] ← DataIn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( i \leftarrow 0 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( i \leftarrow i+1 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( A \leftarrow M[45] )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( B \leftarrow M[89] )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( M[33] \leftarrow B )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( M[254] \leftarrow A )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( j \leftarrow i+1 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( j \leftarrow j+1 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataOut ← M[2]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Assume that after the initialization the circuit performs the following pseudocode:

```
wait for s=1
for i=0 to k-2 do
    A = M[i]
    for j=i+1 to k-1 do
        B = M[j]
        if A > B then
            M[i] = B
            M[j] = A
            A = M[i]
        end if
    end for
end for
Done
wait for s=0
go to the beginning
```

**Problem 4 (5 points)**

Translate the block diagram given in Problem 3 to RTL VHDL, assuming that generic memory, a generic register, and a generic counter have been already defined for you in separate files. Describe all remaining components using dataflow design style. Provide the complete entity and architecture, including the component declarations (if needed in the version of VHDL you use).