Homework 2

due Thursday, September 25, 4:30pm (in class)

All solutions should be developed individually!
The GMU Honor Code applies.

Problem 1A

Draw a block diagram of the datapath unit of a circuit capable of executing the pseudocode given below.
In this code, MEM_D represents a single-port memory of the size of 2048 x 10. The circuit takes as an input a stream of 8-bit ASCII characters representing a document written in English. It calculates the lengths of 2048 first sentences (i.e., strings ending with ‘.’, ‘?’, or ‘!’) and stores them in MEM_D. In parallel, it calculates the minimum, maximum, and average length of a sentence in the input text. The document is assumed to contain at least 2048 sentences. All sentences are assumed to be shorter than 1024 characters.

```
begin:
  wait for s=1
  done = 0
  i=0; count = 0
  start = 0
  sum=0; max=0; min=1023

  while (count < 2048) do
    next = din
    if ((next = '.') or (next = '!') or (next = '?')) then
      length = i - start
      start = i+1
      sum = sum + length
      if length > max then
        max = length
      end if;
      if length < min then
        min = length
      end if;
      MEM_D[count] = length
      count ++;
    end if;
    i++
  end while

  avr = sum/2048

  done = 1
  wait for s=0
  // when s=0, an external circuit can read data from memory MEM_D, one number at a time,
  // using ports mem_addr and mem_dout
  go to begin
```

*Please clearly mark 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 all internal registers and counters. Active high.</td>
</tr>
<tr>
<td>din</td>
<td>8</td>
<td>Input data bus.</td>
</tr>
<tr>
<td>s</td>
<td>1</td>
<td>Operating mode: 0 = waiting for data/reading results, 1 = processing.</td>
</tr>
<tr>
<td>rd</td>
<td>1</td>
<td>Read enable. 0 = high impedance on the output bus dout, 1 = valid output dout</td>
</tr>
<tr>
<td>dout</td>
<td>10</td>
<td>One of the three results calculated by the circuit.</td>
</tr>
<tr>
<td>sel_out</td>
<td>2</td>
<td>Selection between the three calculated results: 0 = avr, 1 = max, 2 = min</td>
</tr>
<tr>
<td>mem_addr</td>
<td>11</td>
<td>Address in memory location MEM_D.</td>
</tr>
<tr>
<td>mem_dout</td>
<td>10</td>
<td>High impedance (if s=1) or value of memory location MEM_D[mem_addr] (if s=0).</td>
</tr>
<tr>
<td>done</td>
<td>1</td>
<td>Asserted when all results are ready, zero otherwise</td>
</tr>
</tbody>
</table>

**Problem 1B**

Draw an interface with the division into the Datapath and Controller for the circuit from Problem 1A.

**Problem 2**

The key scheduling unit of the RC6 cipher is a circuit that takes a 128-bit key $K$, and converts it to $2r+4$ round keys $S[i]$, $i=0..2r+3$, stored in the internal memory $S[i]$. This unit is specified below using its
- pseudocode
- interface
- table of input/output ports.

Perform the following two phases of the design process for the RC6 key scheduling unit:
1. Block diagram of the Datapath
2. Interface with the division into the Datapath and Controller

**Pseudocode:**
The RC6 key scheduling unit is defined using the given below pseudocode. The input key $K$ consists of four words $K[0]$, $K[1]$, $K[2]$, and $K[3]$, each of the size of 32 bits. These words are first written to the internal memory $L[j]$, $j=0..3$, using the control signal write_key (active high) and the key input $Kin$. This process is described using the first for-loop of the pseudocode. Afterwards, the remaining part of the pseudocode is executed by the controller.

$P32$ and $Q32$ are 32-bit constants. $r$ is a parameter of RC6, with the default value of 12.
for j = 0 to 3 do
    while (not write_key)
        do nothing
    end while
    L[j] = Kin
end for

S[0] = P32
for i=1 to 2r+3 do
    S[i] = S[i-1] + Q32
end for

i = j = 0
A = B = 0
for k=1 to 3 \cdot (2r+4) do
    A = S[i] = (S[i] + A + B) \ll 3
    B = L[j] = (L[j] + A + B) \ll (A+B)
    i = (i+1) \mod (2r+4)
    j = (j+1) \mod 4
end for

Notation:
A, B = 32-bit variables
+ = unsigned addition mod 2^{32}
X \ll Y = rotation of the variable X by a number of positions given by the current value of the variable Y

Assume that you can use memories with the following inputs:
ADDR, DIN, CLK, WE, and the output DOUT,
which output data in the same clock cycle in which a new address is applied.

Writing to memory takes effect at the rising edge of the clock when WE = 1.

Interface:

Assume the following interface to your circuit:
<table>
<thead>
<tr>
<th><strong>Port</strong></th>
<th><strong>Width</strong></th>
<th><strong>Meaning</strong></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.</td>
</tr>
<tr>
<td>write_key</td>
<td>1</td>
<td>Synchronous write control signal for the key input, Kin. Active high.</td>
</tr>
<tr>
<td>Kin</td>
<td>32</td>
<td>The key input, through which subsequent words of the key K[0]..K[3] are loaded to the circuit.</td>
</tr>
<tr>
<td>n</td>
<td>m</td>
<td>Index n of the round key S[n].</td>
</tr>
<tr>
<td>Sn</td>
<td>w</td>
<td>Value of the round key S[n] corresponding to the index provided through the input n.</td>
</tr>
<tr>
<td>Done</td>
<td>1</td>
<td>Asserted when the computations are completed.</td>
</tr>
</tbody>
</table>

$m$ is a size of index $n$. It is a minimum integer, such that $2^m \geq 2r+4$.

**Tasks & Assumptions:**

Assume that
- one round of the main for loop of the pseudocode executes in one clock cycle
- you can access only one position of each internal memory per clock cycle
- after the circuit is done with computations, applying input $n$ should generate value of $S[n]$ at the output $Sn$ in the same clock cycle.

Task 2: Draw an Interface of the circuit with the division into the Datapath and Controller.