**Introduction & tasks:**

The key scheduling unit of the RC5 cipher is a circuit that takes a 128-bit key K, and converts it to $2r+2$ round keys $S[i]$, $i=0..2r+1$, stored in the internal memory $S[i]$.

This unit is specified below using its

- a. pseudocode
- b. interface
- c. table of input/output ports.

Perform the following six phases of the design process for the RC5 key scheduling unit implementation:

1. **Block diagram of the Datapath**
2. **Interface with the division into the Datapath and the Controller**
3. **ASM chart of the Controller**
4. **VHDL code of the Controller**
5. **Timing analysis**
6. **Expected-output analysis**

**Pseudocode:**

The RC5 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 RC5, 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+1 do
    S[i] = S[i-1] + Q32
end for
```
\begin{verbatim}
\begin{align*}
  i = j = 0 \\
  A = B = 0 \\
  \text{for } k=1 \text{ to } 3 \cdot (2r+2) \text{ do} \\
  \quad A' = S[i] = (S[i] + A + B) \ll 3 \\
  \quad B' = L[j] = (L[j] + A' + B) \ll (A' + B) \\
  \quad A = A' \\
  \quad B = B' \\
  \quad i = (i+1) \mod (2r+2) \\
  \quad j = (j+1) \mod 4 \\
\end{align*}
\end{verbatim}

\textbf{Notation:}

\begin{itemize}
  \item \( A, B, A', B' \) = 32-bit variables
  \item + = unsigned addition \( \mod 2^{32} \)
  \item \( X \ll Y \) = rotation of the variable \( X \) by a number of positions given by the current value of the variable \( Y \)
\end{itemize}

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.

\textbf{Interface:}

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.</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>----</td>
<td>----</td>
<td>--------------------------------</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 $2^m \geq 2r+2$.

**Tasks & Assumptions:**

**Task 1:** Draw a block diagram of the Datapath

**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 the Controller

**Task 3:** Draw an ASM chart of a controller capable of performing computations described in the pseudocode.

**Task 4:** Write synthesizable VHDL code of the controller defined in Task 3.

**Assume that:**

- values of the parameters $r$ and $m$ are specified as constants or generics.

**Task 5:** Write formulas for the

- latency of your circuit as a function of the minimum clock period, and
- minimum clock period in terms of delays of basic logic components shown in your block diagram

Mark the critical path of your circuit on its block diagram.

**Task 6:**

Draw timing waveforms showing all inputs and outputs from the controller, as well as its internal state during

- first 6 clock cycles after reset becomes inactive (assume that the write_key signal becomes active immediately after reset goes low)
- first 6 clock cycles of the main (third) for-loop of the pseudocode.