ECE 545 Project
Option 2
To be done individually
The same for all students choosing this option

Project 3
due Wednesday, December 14, 2022, 11:59 PM

Develop synthesizable VHDL code for the Controller and Top-Level Unit of the XTEA cipher described below using its

A. **pseudocode**
B. **interface diagram and table of input/output ports**
C. **protocol**
D. **block diagram**
E. **interface with the division into the Datapath and Controller**
F. **ASM chart.**

Verify the functionality of your Top-Level Unit using a testbench based on test vectors generated in Project 2.

A. Pseudocode

The XTEA cipher is defined below:

An input message block M has $2 \cdot w$ bits (where w is a parameter of the cipher).
The corresponding ciphertext block (i.e., encrypted message block) also has $2 \cdot w$ bits.

To encrypt a message block M, the algorithm performs the following operations:

```
Split M into two equal parts V0, V1 each of the size of w bits

SUM = 0

for j= 1 to r do
  {
      W00 = ((V1 << 4) ⊕ (V1 >> 5)) + V1
      W01 = SUM + KEY[SUM mod 4]
      T0 = W00 ⊕ W01
      V0′ = V0 + T0

      SUM′ = SUM + DELTA

      W10 = ((V0′ << 4) ⊕ (V0′ >> 5)) + V0′
      W11 = SUM′ + KEY[(SUM′>>11) mod 4]
      T1 = W10 ⊕ W11
      V1′ = V1 + T1

      SUM = SUM′
      V0 = V0′
      V1 = V1′
 }
C = V0 || V1
```

Notation:

V0, V1, V0', V1', W00, W01, W10, W11, T0, T1, SUM, SUM' = w-bit variables
DELTA = a w-bit constant
K[0], K[1], K[2], K[3] = a set of 4 round keys; each round key is a w-bit variable
$\oplus$ = an XOR of two w-bit words
$+$ = unsigned addition mod 2^w
A << k = logic shift left by k positions
A >> k = logic shift right by k positions
A || B = concatenation of A and B.

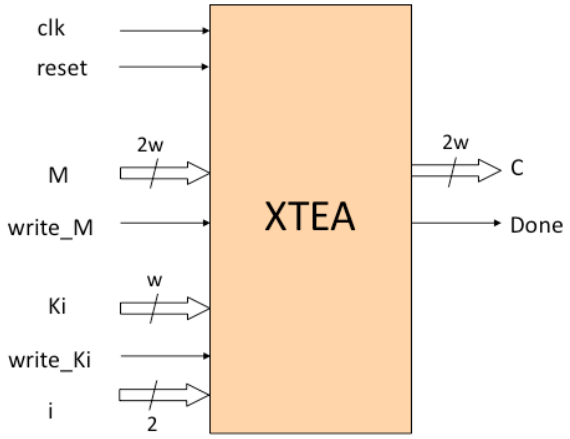
Please note that the algorithm has two parameters:

- r = number of rounds (e.g., 64)
- w = word size (always a power of 2, e.g., $w = 2^5 = 32$)

These parameters should be treated as constants.

B. Interface diagram and table of inputs/outputs

Assume the following interface to your circuit:



Port	Width	Meaning
clk	1	System clock.
reset	1	System reset – clears internal registers.
M	$2w$	Message block.
write_M	1	Synchronous write control signal for the message block M. After the block M is written to the XTEA unit, the encryption of M starts automatically.
Ki	w	Round key K[i] loaded to the internal storage.
write_Ki	1	Synchronous write control signal for the round key K[i].
i	2	Index of the round key K[i] loaded using input Ki.
C	$2w$	Ciphertext block = Encrypted block M.
Done	1	Asserted when ciphertext is ready and available at the output.

C. Protocol

An external circuit first loads all round keys
K[0], K[1], K[2], K[3]
to the internal storage of the XTEA unit.

Loading of round keys is performed using inputs: Ki, i, write_Ki, clk.

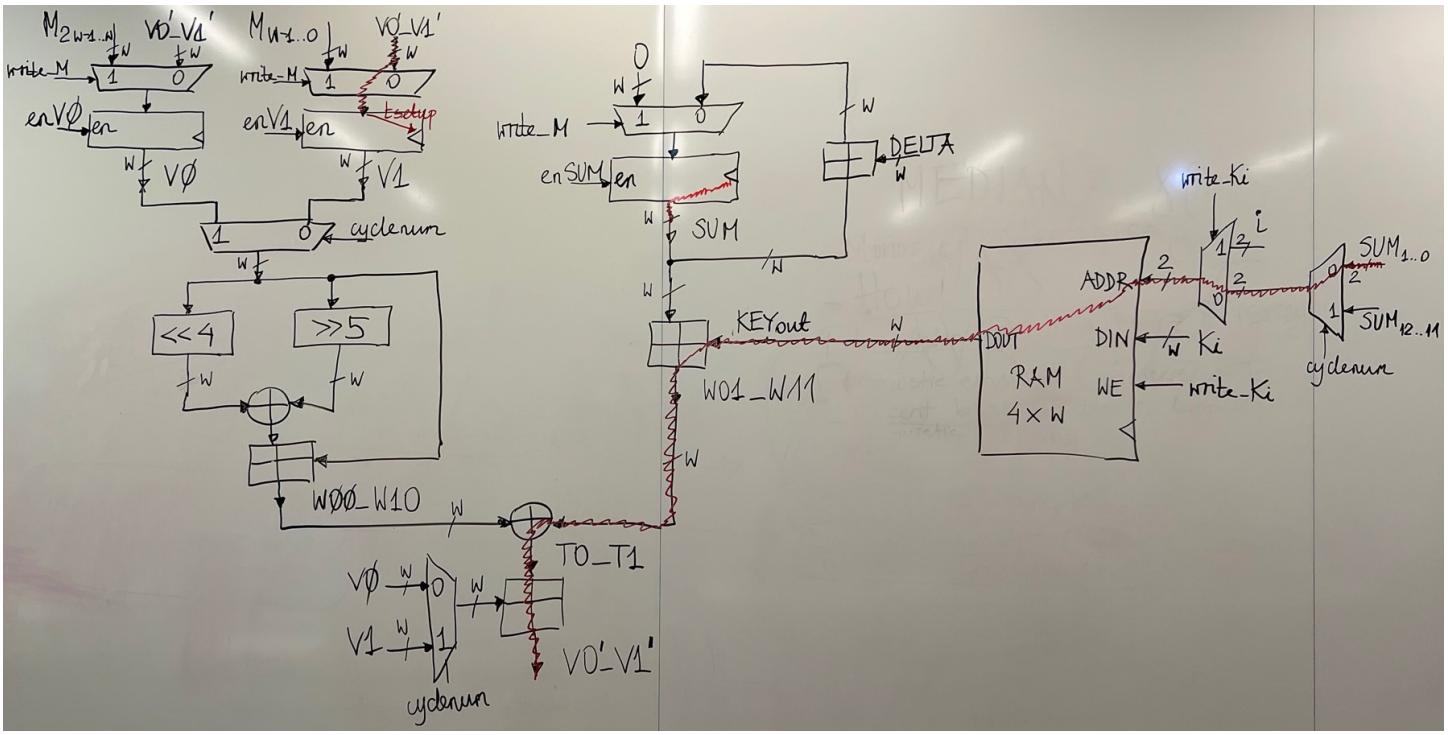
Then, the external circuit loads a message block M to the XTEA unit, using inputs: M, write_M, clk.

After a message block M is loaded to the XTEA unit, the encryption starts automatically.
When the encryption of each block is completed, signal Done becomes active for one clock cycle, and the output C changes to the new value of the ciphertext. The next message block, M, can be loaded to the circuit one clock cycle later.

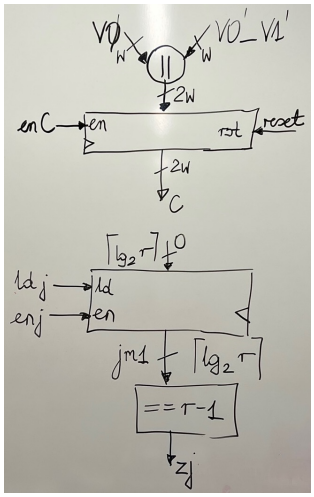
The output C keeps the last value of the ciphertext at the output until the next encryption is completed. Before the first encryption is completed, this output should be equal to zero.

D. Block diagram

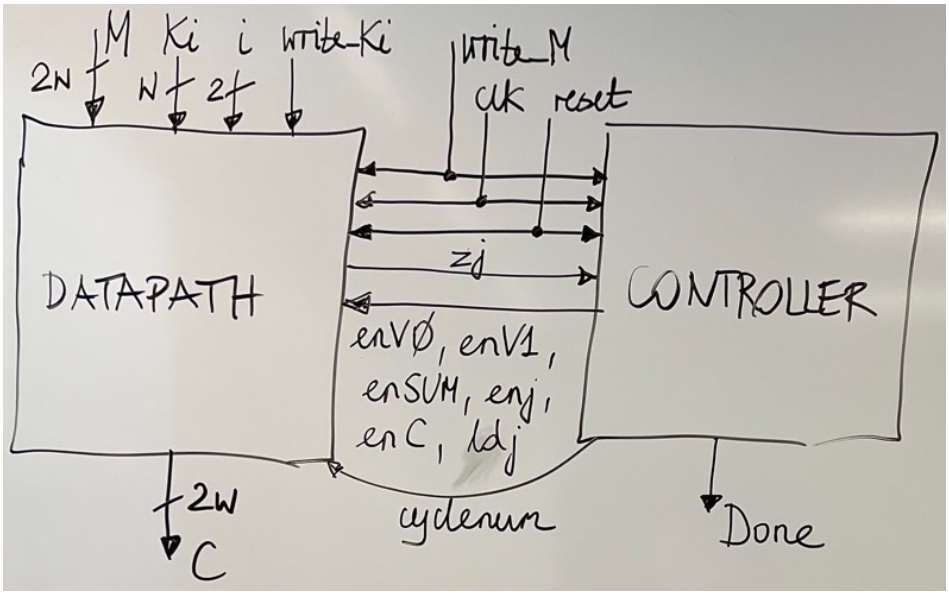
Main loop:



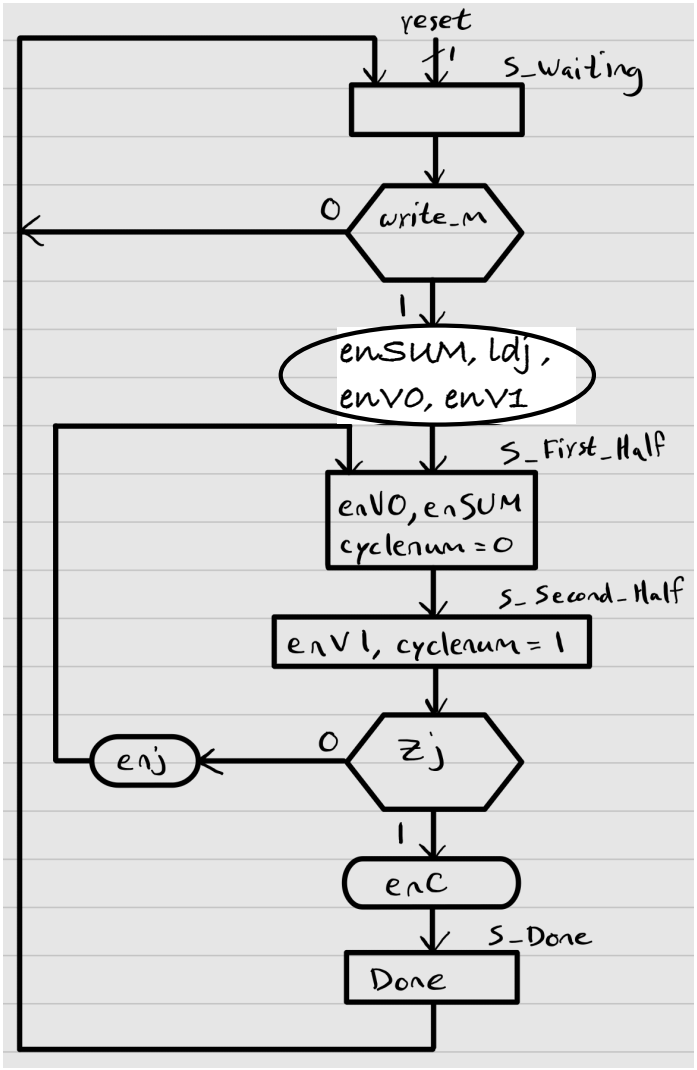
Output logic and j-counter:



E. Interface with the division into the Datapath and Controller



F. ASM Chart



Tasks & Assumptions:

Task 1:

Develop the code of the Controller closely matching the ASM Chart provided above.
Develop the code of the Top-Level Unit closely matching the Interface with the division into the Datapath and Controller provided above.

Make sure that your code is fully synthesizable by performing synthesis targeting the following FPGA device:

Family: Artix-7
Device: xc7a12t
Package: csg325
Speed: -3
Full name: xc7a12tcsg325-3

Eliminate all synthesis errors and minimize the number of synthesis warnings.

Task 2:

Write a simple testbench and debug your entire code so that the functional simulation generates the same intermediate and final results as those generated in Tasks 3 and 4 of Project 2.

Task 3:

Perform the synthesis and implementation of your fully debugged code targeting the same FPGA device as in Task 1.
If needed, address all synthesis and implementation errors. Do your best to minimize the number of warnings.

Please use a binary search to determine the approximate maximum clock frequency of your implementation. Start from any two target clock frequencies for which the WNS (Worst Negative Slack) values have the opposite signs and end with two frequencies with the same feature that are no more than 25 MHz apart.

Deliverables:

1. Synthesizable source code developed in Tasks 1 and 3.
2. Testbench developed in Task 2.
3. Report containing at least the following information:
    A. Results of verification obtained in Task 2.
    B. For each of your synthesis & implementation runs, please determine and include in the report the following values obtained in Task 3:
        a. Resource Utilization (LUTs, FFs, BRAMs, DSPs)
        b. Target Clock Period [ns]
        c. Target Clock Frequency [MHz]
        d. WNS (Worst Negative Slack) [ns]
        e. TNS (Total Negative Slack) [ns].