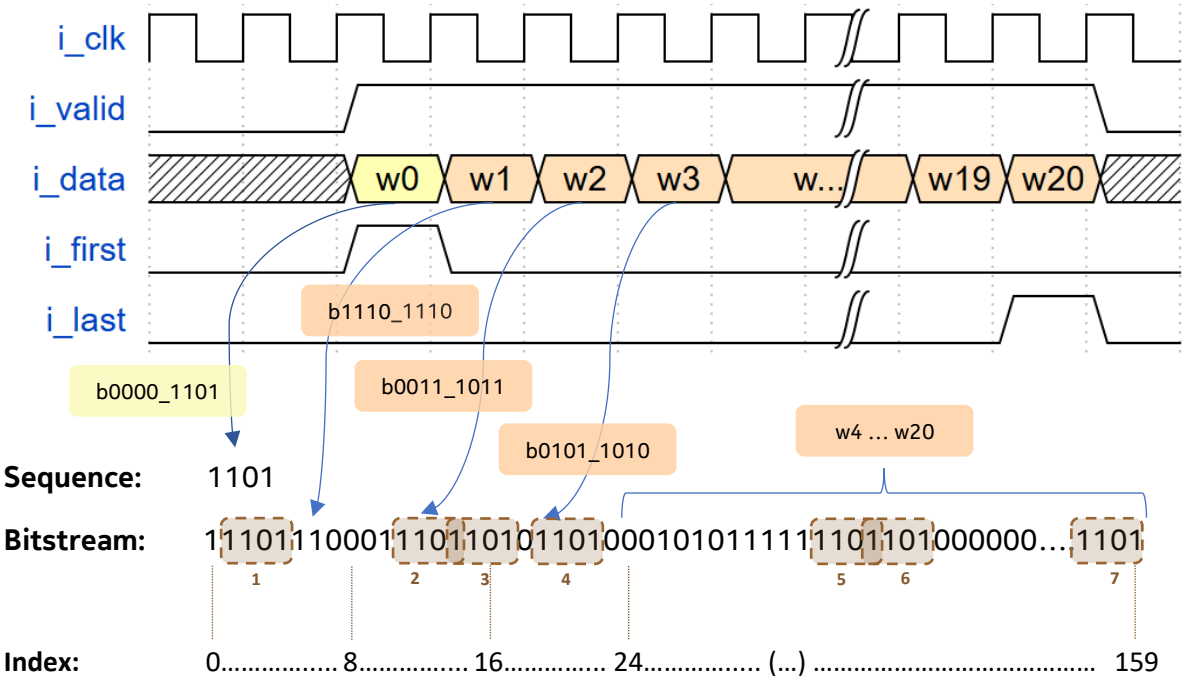
Sequence Counter

Theoretical background

Binary representation of numbers is one of the pillars of the digital world. Understanding and manipulating these binary sequences is crucial for various applications. Recognition of certain patterns occurring in bitstreams may be applied in different industry fields: data communication (synchronization, protocol handling, error detection), control systems (triggering events or state changes under specified circumstances), signal processing (e.g. pattern recognition) or cryptography.

Goal

Create an RTL module that counts occurrences of the bit sequence in the bitstream received on the task input and stores the indices of the matches. The sequence is 4-bit wide and it is sent as the **first** word sent over the *i_data* bus. The module receives maximum 33 input data words (including the sequence word). Return the number of sequence occurrences and the indices of the matches. Beware of the possibility of overlapping sequences in the bitstream (see: the example below).



In the shown example, the expected output should be:

Table 1 - Expected output data for given example

Consecutive words send over *o_data* bus	**Value** (shown here in decimal format)	**Comment**
o_data[0]	7	Number of sequence occurrences
o_data[1]	1	Index of 1st sequence match
o_data[2]	11	Index of 2nd sequence match
o_data[3]	14	Index of 3rd sequence match
o_data[4]	19	Index of 4th sequence match
o_data[5]	35	Index of 5th sequence match
o_data[6]	38	Index of 6th sequence match
(assumption: no sequence matches detected between 35 – 156 indices)		
o_data[7]	156	Index of 7th sequence match

Provided module

You can edit only the *task_3.sv* file. You may add other files for supportive submodules if needed.

You only get the outline of the ***task_3*** module. Inputs and outputs are declared as shown below in the Table 1.

Table 1 List of inputs and outputs of the task_3 module.

Signal name	**Signal type**	**Bit length**
i_clk	Input	1 bit
i_rst	Input	1 bit
i_valid	Input	1 bit
i_data	Input	8-bit
i_first	Input	1 bit
i_last	Input	1 bit
o_data	Output	8-bit
o_valid	Output	1 bit
o_last	Output	1 bit

Input and output interfaces

In one packet task receives **the bitstream as consecutive samples** on the *i_data* interface. The first sample received contains the sequence on the 4 least significant bits (four most significant bits of the first sample may be neglected). The second sample should be treated as the bitstream's beginning and the last sample received (indicated by signal *i_last)* as the bitstream's end. **Do not change the bits' order.**

Task receives data in the way shown in the figure below:

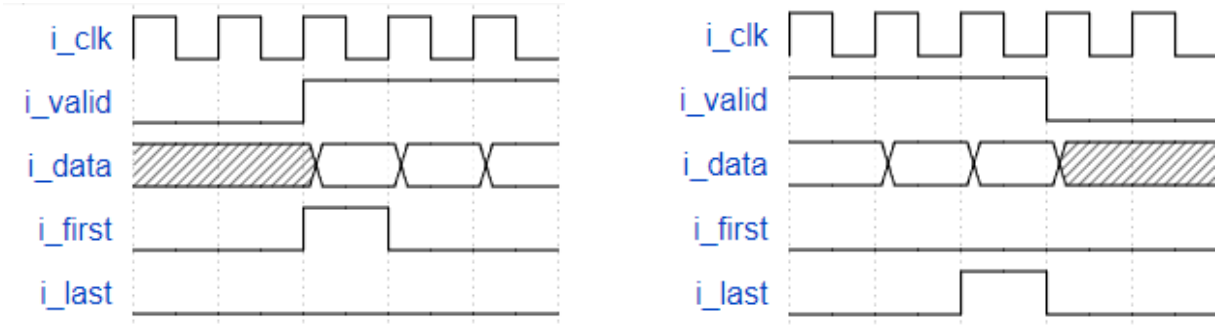


Figure 1 Input waveforms – the start of the packet (left) and the end of the packet (right).

The task should send number of sequence's occurrences and their indices using *o_data* interface and properly synchronized control signals: *o_valid* and *o_last.*

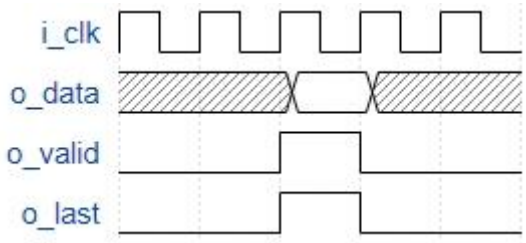


Figure 2 Output waveforms.

Evaluating the task

Evaluation is done by comparing the number of sequences and matches' indices returned by the task with the values calculated by the reference model in the judge software. If the values are equal, then the task is marked as correctly resolved. Ascending order of the indices is expected.

In Development mode, the task will be tested using a single set of data that is randomized on each run. In Evaluation mode, the task will be tested using 3 fixed sets of data, which remain the same across all runs and are identical for every team (for more information, check chapter 2.3.1 **Testing modes** in the **FPGA_Hackathon_2025_Project_Guide** document).

You earn 3 points for correctly processing each test vector. In Development mode, this value simply indicates that the task passed the test. Evaluation mode, your total base score is calculated by multiplying 3 points by the number of test vectors, so you can earn up to 9 points.

Additional points may be awarded for **resource utilization** (for more information, check chapter 2.3.2 **Bonus Points** in the **FPGA_Hackathon_2025_Project_Guide** document).