Date: 21.05.2021
Author: Sławomir Batruch

I declare that this piece of work that is a basis for the grading of edc1 laboratory 5 task was performed on my own without any unallowed help (indirect or direct) from other students.

Sławomir Batruch; ID 303827

EDC1 Laboratory 5 – Microprogramming (1)

For this laboratory, the task was to design a Datapath unit that would compute a mathematical average of some given k=9 bit numbers. Here is the mathematical formula from the laboratory .pdf, as well as the requirements for numbers and other details:

Design a digital system computing:

$$Y = Int\left[\sum_{i=1}^{n} x_i / n\right]$$

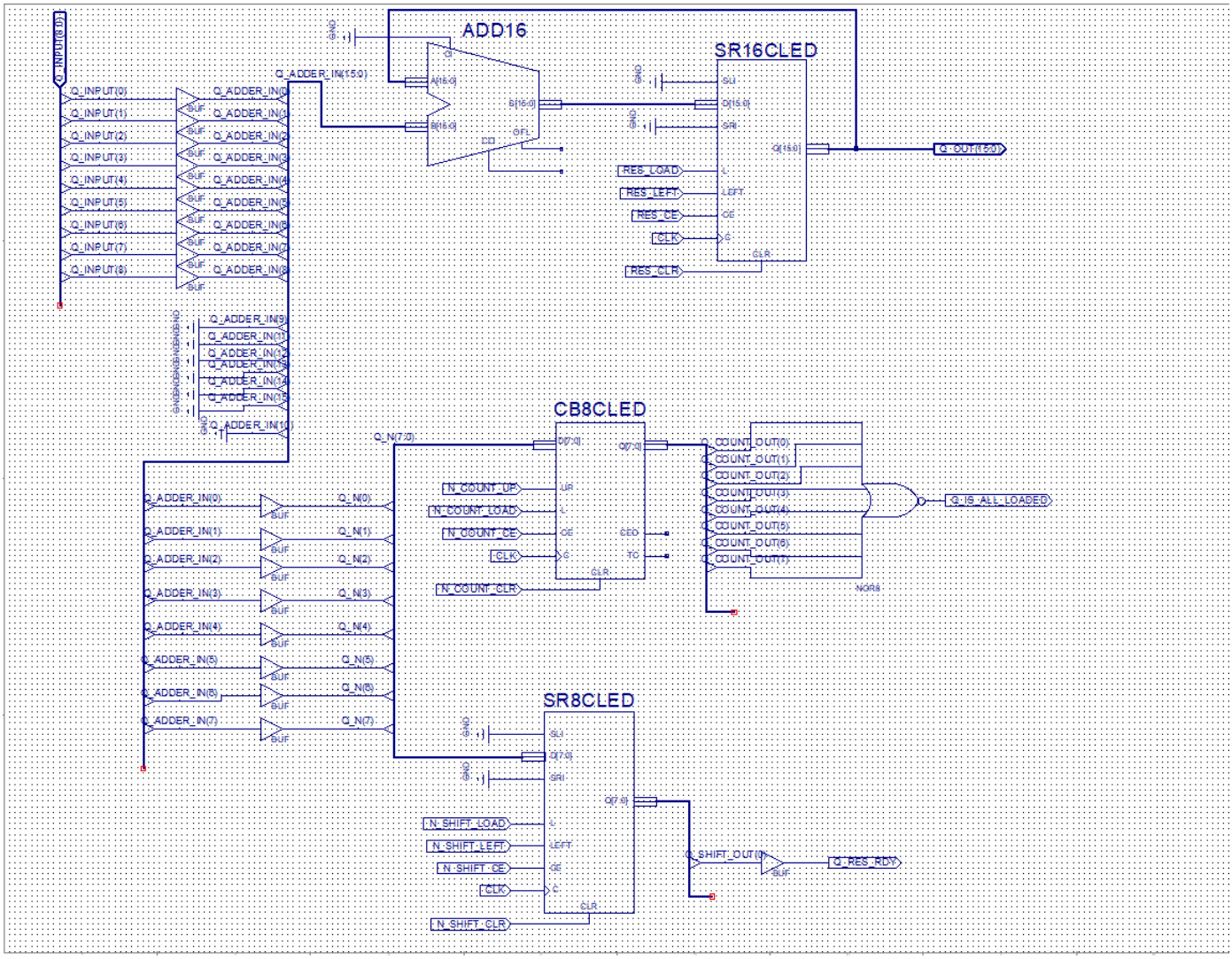
where:

- n and x_i for i = 1, ..., n are k-bit fractions of the 16-bit natural binary coded numbers;
- n > 0, n < 2^8, n is a power of 2 ;
- the result Y is 16-bit binary coded number;
- numbers n and x_i are supplied in parallel.
- All busess, including input and output must be named beginning with the character assigned individually by the lecturer

The letter given to me was Q (regarding last point)

Datapath schematic and principle of operation

Here is the datapath schematic I designed in Xilinix:



Starting from the top of the schematic, SR16CLED and ADD16 are devices that perform the addition of supplied numbers from the input. SR16CLED stores the number, and ADD16 adds the stored number to a supplied number.

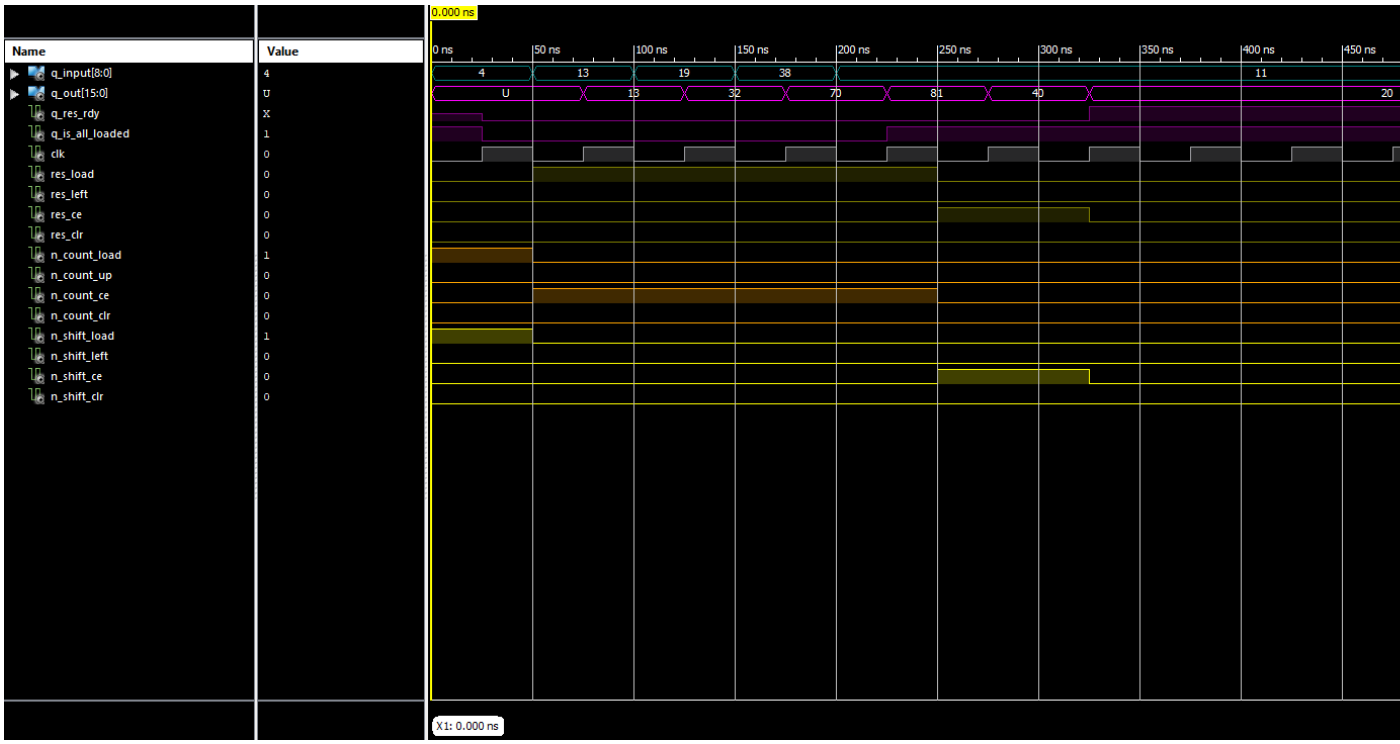
At the bottom we got a shift register CB8CLED and counter SR8CLED that store the value of n, which is supplied to the system at the start of its operation. Counter is used to track the amount of supplied numbers into the circuit, and the shift register is used to track the moment at which we should stop shifting the register storing our result.

Here is a simplified principle of operation, step by step:

1. *Enable load on the counter and shift register from the bottom of the schematic*
2. *Load n into the input*
3. *Disable load on abovementioned counter and shift register*
4. *Load a number*
5. *Enable load on the register storing our result*
6. *Load numbers until the total of loaded numbers is N*
    a. *This is tracked by the counter decrementing by 1 every number loaded*
7. *When all numbers have been loaded, a control output will change to 1 (this indicates counter has reached 0 – so all numbers were loaded)*
8. *Disable the counter*
9. *Start shifting to right in both registers*
10. *When the register with n has shifted and obtained 00…0001, stop shift operation in both register*
    a. *This is tracked by an output indicating 1 when the shifting should stop, i.e. it indicates when bit at position 0 is equal to 1. This is due to n being a power of two, so it's binary representation is one '1' and rest is '0'*
11. *Declare that the result is ready and disable the LOAD and CLOCK_ENABLE function of all registers*

Simulation

The simulation output was correct. This is how the waveforms look like after proper VHDL testbench design and some waveform display changes (order, radix, colors etc.):



(Note: Better quality in a sent .zip file after lab)

The correctness of the result for the given numbers. n=4, so we take 4 numbers shown below. <u>n is not a part of the sum.</u>

$$\mathbf{13 + 19 + 38 + 11 = 81},\ \boldsymbol{then}\ \mathbf{81 / 4 = 20}$$

(Result is rounded down due to how shifting right binary numbers works, i.e. when shifting 81 in a binary representation (or any other odd number), we remove '1' from the LSB)

Troubleshooting and issues encountered

Some issues I encountered were:

- Due to a missing bus tap for digit 10 (that should be connected to ground) from the bus Q_ADDER_IN, the result register didn't work at all, it was const. 0. Fixed by adding the missing digit. *Interesting how the simulator didn't show X (invalid) nor U(nknown) value for this bit. It just continued working, but not correctly*
- Negligible issues at the start of the simulation most likely related to the way Xilinx ISE initialises the devices in the circuit. U(nknown) or X (invalid) at the start for some values don't influence the results and the operating principle of the circuit