UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

SIMPLE SIGNAL PROCESSING SYSTEM THAT COMPUTES THE ROLLING AVERAGE OF 8-BIT DATA STREAM

-DSD FIRST YEAR PROJECT

STUDENTS:
-ORIAN OCTAVIAN
-ROSIANU SEBASTIAN

PROJECT SUPERVISOR:
-TIMAR MIHAI

TABLE OF CONTENTS:

1.Specifications

-The task was to develop a rolling average computer that takes at each clock cycle a random number generated in the data generator block and displays on the screen the resulted average.

-We used two LFSR on 4 bits and 8 bits respectively and the generated number travels through a filter and it's stored in a FIFO type memory from which we choose the last 2,4,8 or 16 numbers to compute the rolling average.

-The Data Flow (on 8 bits) is transmitted from the generator to the computer in serial manner (1 number/1 clock cycle).

-The numbers stored in the registers are then added up depending on the filter input and the sum is stored in another register and then shifted right to perform division (once for 2 numbers, twice for 4 numbers, three times for 8 numbers and four times for 16 numbers).

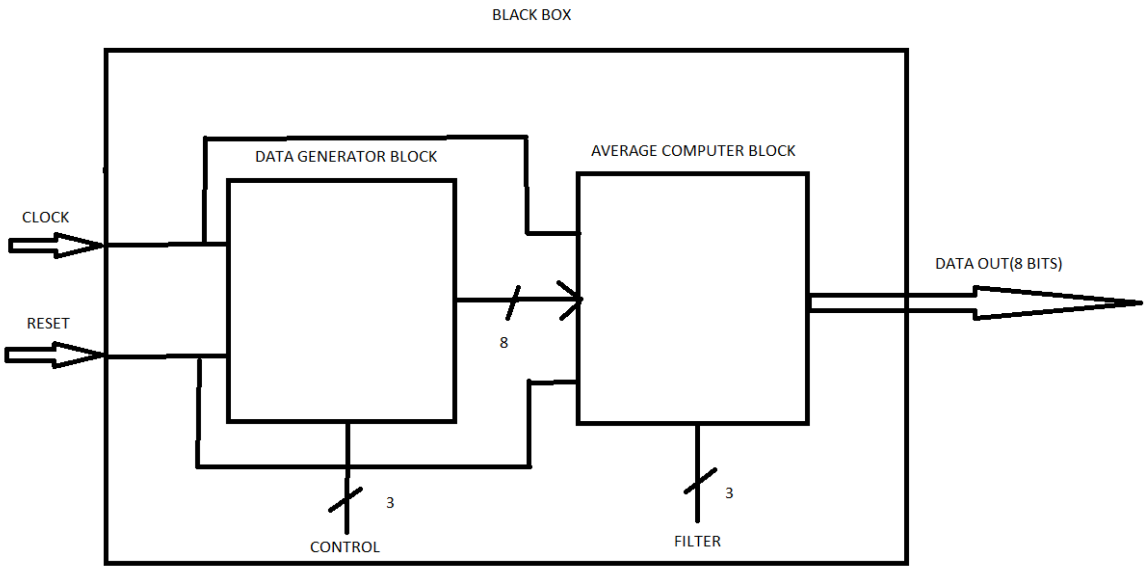
-The average is computed independently each time a number enters the memory so we used 8 8-bit adders, 4 9-bit adders, 2 10-bit adders and one 11-bit adder which are connected to each other.

-The entire project was described in a structural way, describing each component and linking them together.

-The main drawback of this is the fact that we can’t reuse the average for the first 4 numbers to compute the average of the first 8 numbers for example and we compute the sum each time, but despite this the system works very precisely (but with no decimals) and computes fast the desired average.

2.DESIGN

-Here we have the black box and the two major components inside it: data generator block and average computer block



3.Data generator block

-The data generator block is the unit where values are generated and are going to the next component which is Average computer block.

-Depending on the choice of the user, the generated numbers can be on 4 bits or on 8 bits.

-However, when the user wants to generate 4-bits numbers, to reduce the usage of additional components we concatenated “0000” in front of the other 4 bits to use the 8-bit registers and adders for both kinds of numbers.

-The next table shows the controls for the Data generator:

c1	c0	MODE
off	off	Pseudo-random sequence range 0-255
off	on	Pseudo-random sequence range 0-15
on	off	Repeated 6 digits sequence of st.1
on	on	Repeated 6 digits sequence of st.2

-The main components of the Data generator block are:

1.LFSR on 4 bits

-it was developed using a 4 bit signal that works like a D Flip-Flop and one XOR gate
-tap points taken are from Flip-Flop number 1 and 3

2.LFSR on 8 bits

-it was developed using a 8 bit signal that works like D Flip-Flops and three XOR gates
-tap points taken from Flip-Flops number 1,5,6,7

3.MUX 4:1 for control instructions

-we took c1c2 as selections for the output of the data generator block, which can be either the output of LFSR on 8 bits,4 bits or the inputs of the students.

4.Average computer block

-the average computer takes the output of the data generator block and let you choose using an input filter whether you want to compute the average of the last 2,4,8 or 16 generated numbers

-the main components of this block are:

1.Memory

-it’s composed of 16 8-bits registers which act as a FIFO memory to store the generated values and prepare them to enter in the adders

2.Adders

-there we have 4 types of adders:

a)8-bit adders- there are 8 such adders which compute the sum of the values stored in registers 1-2, 3-4, 5-6 and so on

b)9-bit adders- there are 4 such adders which compute the resulted sums from previous adders

c)10-bit adders- there are 2 such adders

d)11-bit adder- the result will be the total sum of the content inside memory

3.Average computers

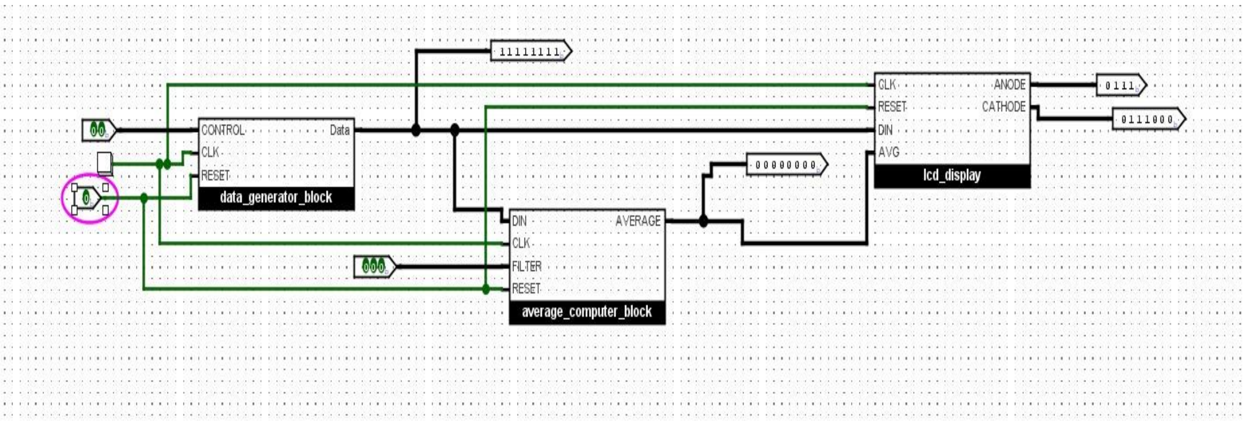
-there are 4 kinds of average computers:

a) for 2 numbers- takes the output of the 8-bit adder of the last 2 generated numbers and throws away the LSB in order to divide by 2
b) for 4 numbers- takes the output of the 9-bit adder of the last 4 generated numbers and throws away the 2 LSB in order to divide by 4
c) for 8 numbers- takes the output of the 10-bit adder of the last 8 generated numbers and throws away the 3 LSB in order to divide by 8
d) for 16 numbers- takes the output of the 11-bit adder of the last 16 generated numbers and throws away the 4 LSB in order to divide by 16

4. 4:1 MUX

-takes the output of each average computer and based on a filter outputs the selected average
-the next table shows how the filter works:

f1	f0	FILTER
off	off	Average of the last 2 numbers
off	on	Average of the last 4 numbers
on	off	Average of the last 8 numbers
on	on	Average of the last 16 numbers



5.Future development

-this system can be improved in many ways and can cover a larger variety of modes and functionalities

-it can read data from a RAM or ROM memory instead of generating numbers inside the data generator, or the numbers can be generated in other ways as the developer decides to make its architecture

-the average computer block can also be modified to calculate the sum and average of any other numbers inside its memory, also the memory can be enlarged if we want more numbers with bigger sizes