

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
КАФЕДРА КОНСТРУЮВАННЯ ЕЛЕКТРОННО-ОБЧИСЛЮВАЛЬНОЇ АПАРАТУРИ

Лабораторна робота №3
З курсу: «Пристрої зі змінною архітектурою»
Тема: «Моделювання проекту в середовищі Quartus II»

*Виконав студент IV курсу гр.
ДК-91 Тисяк Є. В.*

*Перевірів
Антонюк О.І.*

Більшість перевірки я виконував не за допомогою Quartus II, а з використанням ModelSim. Я так зробив з наступних причин:

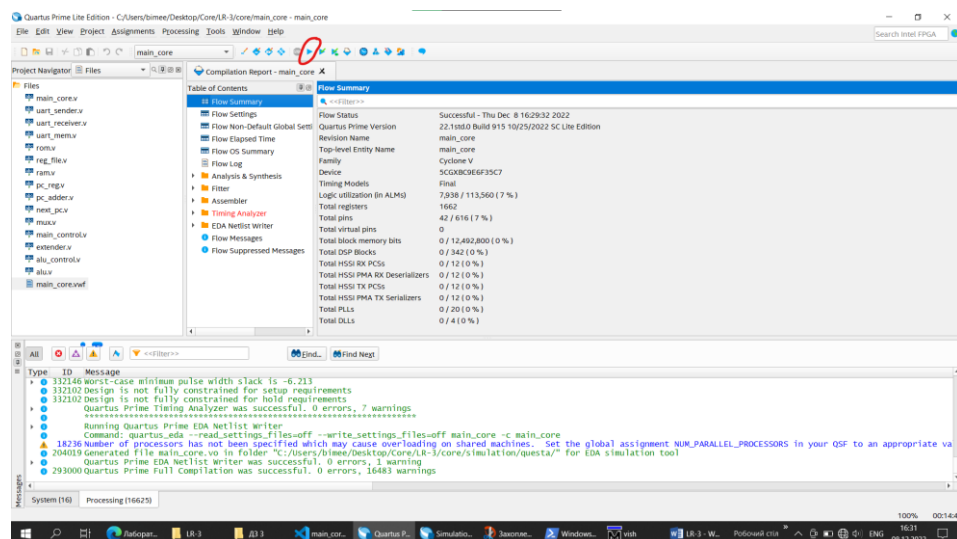
- Я звик працювати в ModelSim
- Я не зміг розібратись з тим, як переглядати в Quartus II значення регістрів й інших структур, що не є вхідними/вихідними сигналами
- ModelSim дозволяє використовувати debug режим

Результат виконання ЛР, а саме файли мого проекту я розмістив за наступним посиланням: <https://github.com/LAMPA23/ModelSim/tree/semestr-7/Lab-3/>

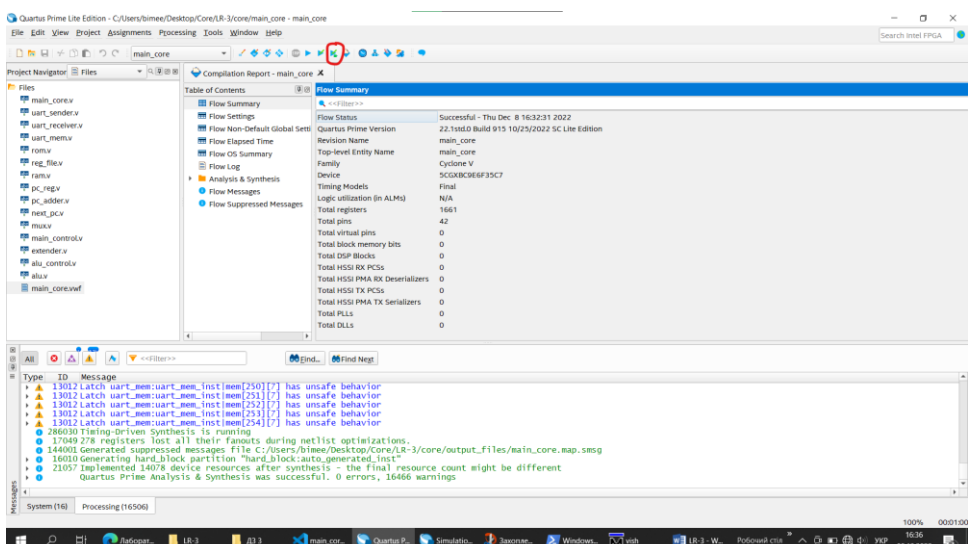
Значних змін зазнали файли *rom.v*, *rom_init.dat*, *uart_sender.v*, *uart_receiver.v* *uart_mem.v*.

РЕЗУЛЬТАТ МОДЕЛЮВАННЯ В QUARTUS II:

Створений мною проект коректно компілюється:



Створений мною проект коректно проходить етап аналізу і синтезу:



Для поверхневої перевірки я добавив 2 сигнали.

Перший:

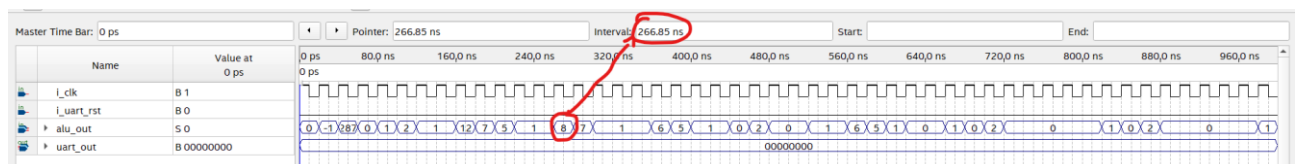
```
main_core.v
1  module main_core(
2      i_clk,
3      i_uart_rst,
4      uart_out,
5      alu_out
6  );
7
8  input i_clk, i_uart_rst;
9  output [7:0] uart_out;
10 output [31:0] alu_out;
11
```

```
main_core.v
49
50 assign alu_out = wire_32_3;
51
```

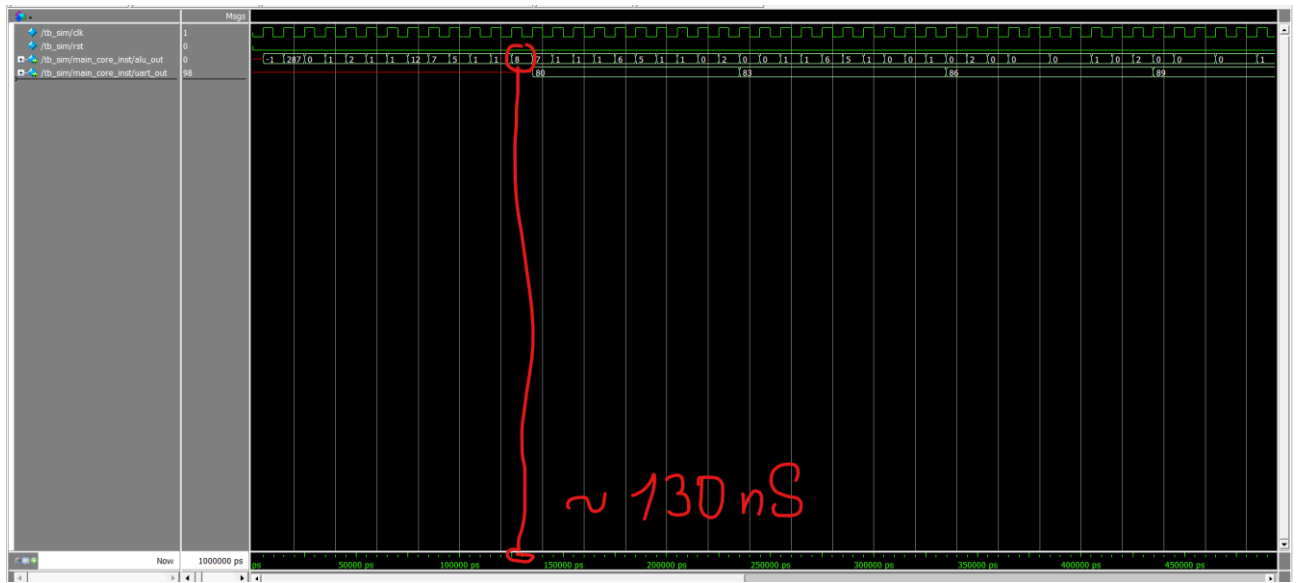
```
main_core.v
107
108 alu alu_inst(
109     .i_op1(wire_32_7),
110     .i_op2(wire_32_0),
111     .i_control(wire_6_0),
112     .o_result(wire_32_3),
113     .o_zf(zero)
114 );
115
```

В результаті даний сигнал повністю відповідає очікуванням. Сигнал по послідовності значень збігається з моделлю в ModelSim, але має зовсім інші значення в часі.

Симуляція в Quartus II:



Симуляція в ModelSim:



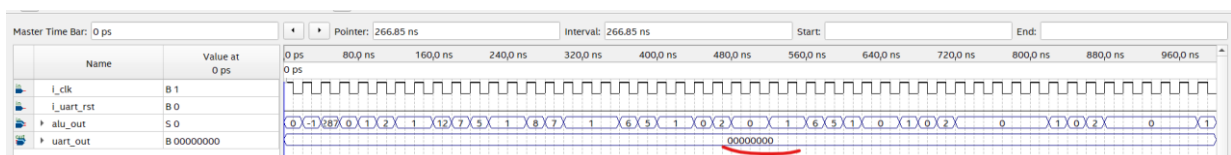
Другий:

```
main_core.v
1  module main_core(
2      i_clk,
3      i_uart_rst,
4      uart_out,
5      alu_out
6  );
7
8  input i_clk, i_uart_rst;
9  output [7:0] uart_out;
10 output [31:0] alu_out;
11
```

```
main_core.v
190
191  uart_receiver uart_receiver_inst(
192      .i_clk(i_clk),
193      .i_rst(i_uart_rst),
194      .i_data(uart_data),
195      .o_data(uart_out)
196  );
197
198  endmodule
```

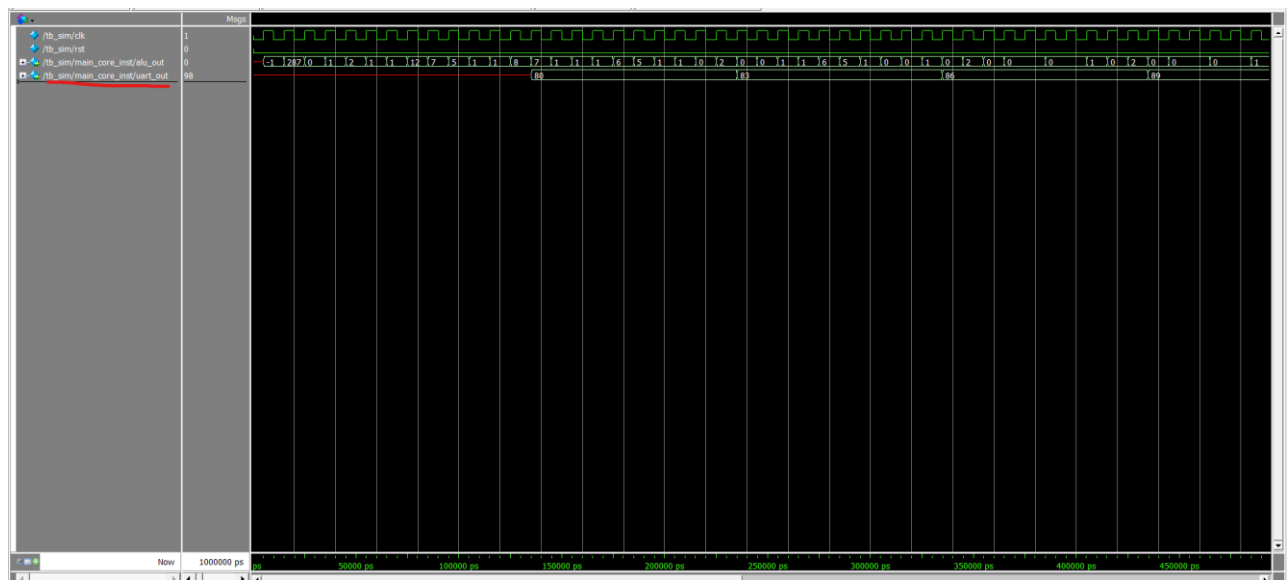
В результаті даний сигнал повністю НЕ відповідає очікуванням. Натомість в ModelSim він є коректним і все працює так як задумано.

Симуляція в Quartus II:

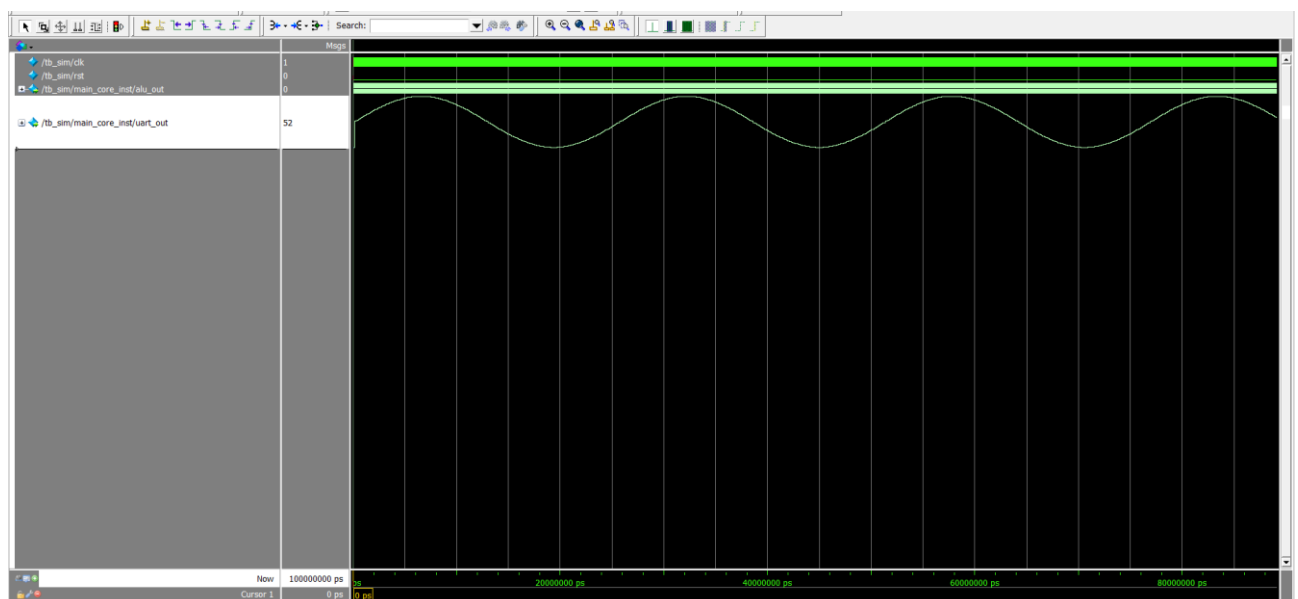


Симуляція в ModelSim:

```
uart_mem_init.dat
1  80838689
2  8c8f9295
3  989b9ea2
4  a5a7aaad
5  b0b3b6b9
6  bcbec1c4
7  c6c9cbce
8  d0d3d5d7
9  dadcdee0
10 e2e4e6e8
11 eaebedee
12 f0f1f3f4
13 f5f6f8f9
14 fafafbfc
15 fdfdfefe
16 feffffff
17 ffffffff
18 fefefefd
19 fdfcfbfa
20 faf0f8f6
```



А ось як буде передаватись вся синусоїда:



РЕЗУЛЬТАТ МОДЕЛЮВАННЯ В ModelSim:

Для перевірки одночасно з передачею по uart запустив сортування чисел в регістрах \$4, \$5, \$6. Після сортування число в \$6 змінюється одиничкою. Все працює коректно:

The screenshot displays the Waveform Viewer interface. The top menu bar includes File, Edit, View, Add, Format, Tools, Window, and Help. Below the menu is a toolbar with various icons for file operations, zooming, and waveform manipulation. The main area shows a list of signals on the left, including 'zb_sim/zb_main_core_inst/reg_file...', and a large waveform plot on the right. The plot shows a digital signal trace with logic values (0, 1) and a large block of 0s. The time scale ranges from 0 to 1200 ns. The signal 'zb_sim/zb_main_core_inst/reg_file...' is highlighted in the list, and its waveform is shown in the plot. The waveform shows a sequence of logic values (0, 1) and a large block of 0s. The time scale ranges from 0 to 1200 ns.

ВИСНОВОК:

Загалом етап моделювання можна вважати пройденим. Швидша за все, що і частину з uart, можна просимулювати в Quartus II, просто для цього слід відслідкувати де і коли, все починає йти «не за планом». Можливо ця проблема буде вирішена в інших ЛР.

Загалом етап моделювання можна вважати пройденим. Швидша за все, що і частину з uart, можна просимулювати в Quartus II, просто для цього слід відслідкувати де і коли, все починає йти «не за планом». Можливо ця проблема буде вирішена в інших ЛР.