

Лабораторна робота №1

Знайомство з Verilog та симулятором Modelsim

Мета роботи: отримання базових знань про структуру програми на Verilog, налагодження програми за допомогою симулятора.

Зміст:

Короткі теоретичні відомості	1
Найпростіша програма на Verilog	1
Основні елементи інтерфейсу ModelSim	2
Створення нового проекту	2
Додавання файлів до проекту	3
Відкриття існуючих проектів	4
Компіляція програми	5
Моделювання роботи програми	7
Завдання для виконання	12
Контрольні питання	13
Вимоги до оформлення звіту	14
Література	14

Короткі теоретичні відомості

ModelSim – це система HDL-моделювання цифрових пристроїв. Одна з характерних тенденцій сучасного етапу розвитку технології проектування цифрових систем - застосування мов опису апаратури *HDL* (*Hardware Description Language*) високого рівня, серед яких найбільшого поширення набули *VHDL* та *Verilog* .

При цьому не тільки зростає роль засобів моделювання в процесі розробки пристроїв, а й змінюються методи та вимоги до процедур верифікації. Для своєчасного виявлення можливих помилок засоби моделювання повинні забезпечувати можливість контролю результатів кожного етапу процесу проектування: створення вихідних *HDL*-описів, синтезу, розміщення та трасування в кристал. Такий підхід забезпечує мінімальний час розробки пристрою та скорочує вартість цього процесу, оскільки ціна помилки зростає з кожним наступним кроком проектування.

Пакет програмних засобів *ModelSim*TM корпорації *Model Technology* (одного з підрозділів компанії *Mentor Graphics*) є однією з найпоширеніших систем *HDL*-моделювання.

Найпростіша програма на Verilog

Наведена далі найпростіша програма складається з ключових слів `module` та `endmodule`, гкі означають початок та кінець блоку *Verilog*. `top` – ім'я модуля, що розглядається.

За допомогою процесу ініціалізації, який визначає ключове слово `initial`, задаються ті інструкції, які будуть виконані один раз перед початком моделювання роботи програми.

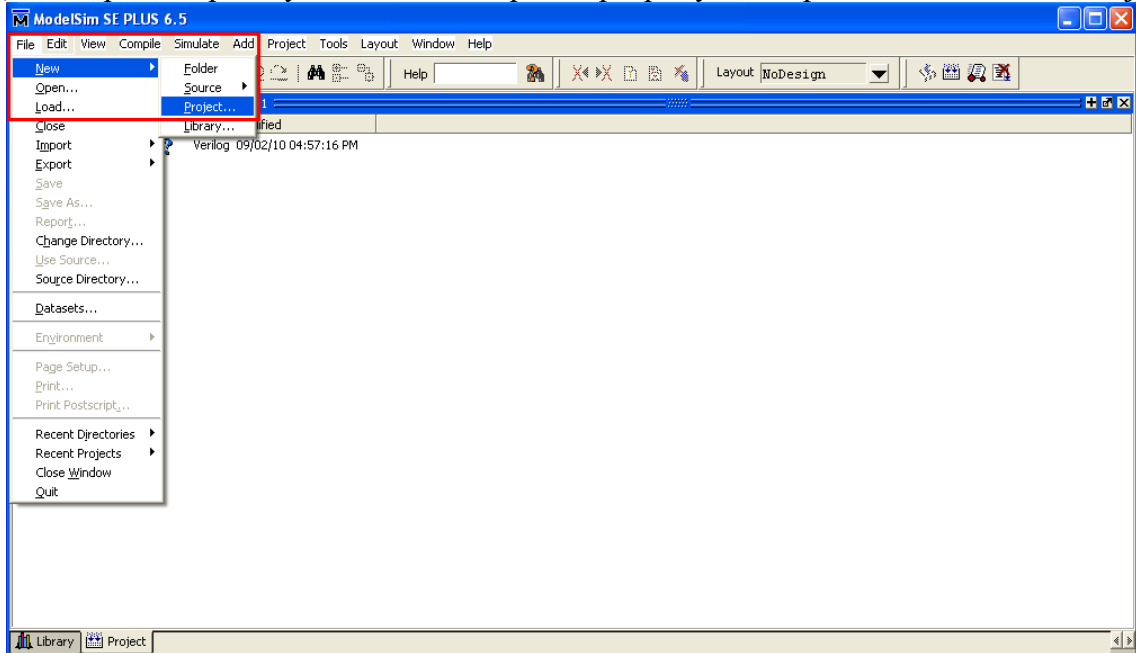
Системна задача $\$display$ дозволяє вивести на екран повідомлення «Hello world!».

```
module top ;
  initial $display( "Hello world!" );
endmodule
```

Основні елементи інтерфейсу ModelSim

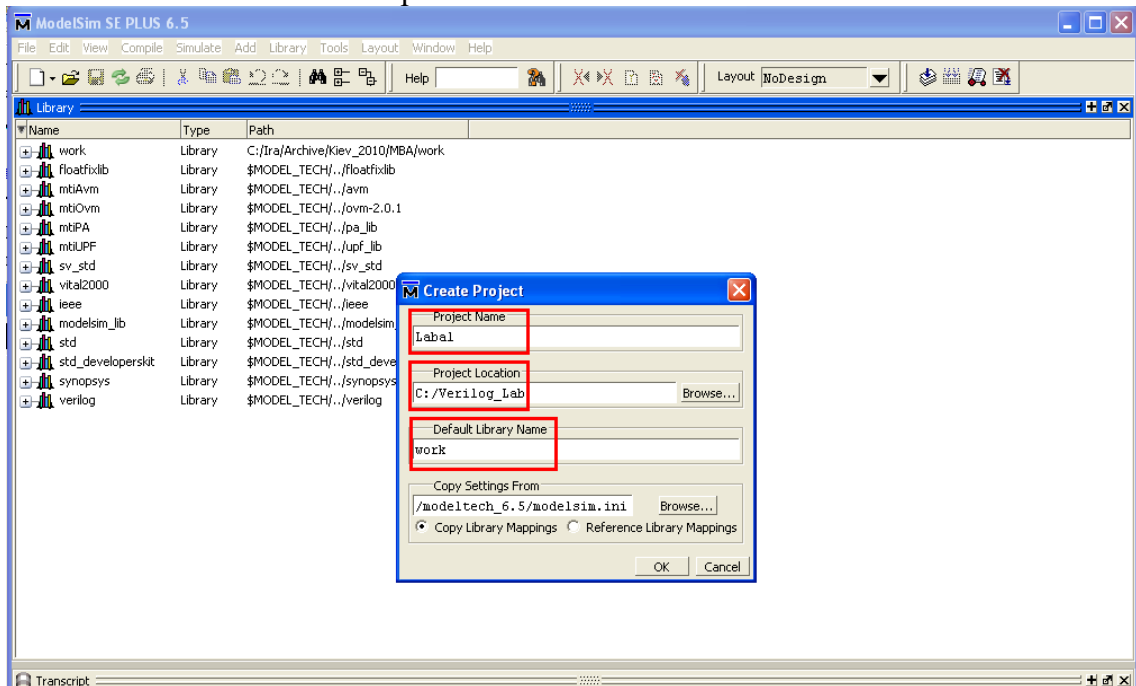
Створення нового проекту

Для створення проекту необхідно відкрити програму та вибрати **File** → **New** → **Project**.



Відкриється вікно діалогу створення проекту. Заповнюємо необхідні поля:

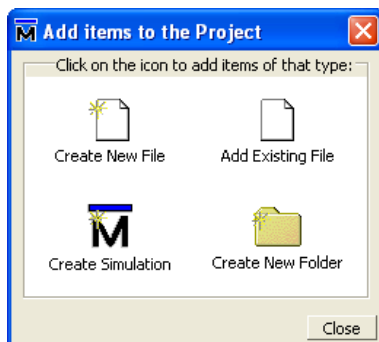
- ✓ Ім'я проекту (*Project Name*),
- ✓ Шлях до проекту (*Project Location*),
- ✓ Назва бібліотеки, в яку компілюватимуться об'єкти проекту (*Default Library Name*). Для всіх завдань з курсу лабораторних робіт достатньо користуватися бібліотекою за промовчанням – *work*.



Після заповнення цих полів натискаємо кнопку **OK**.

Додавання файлів до проекту

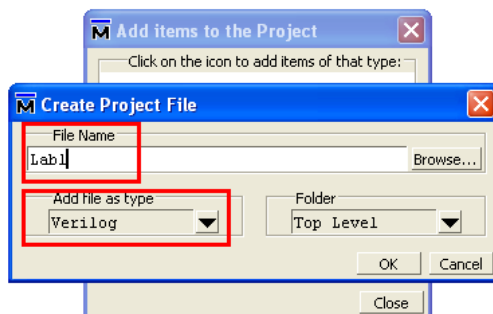
Після створення нового проекту з'явиться діалогове вікно, яке дає змогу додати файли до проекту.



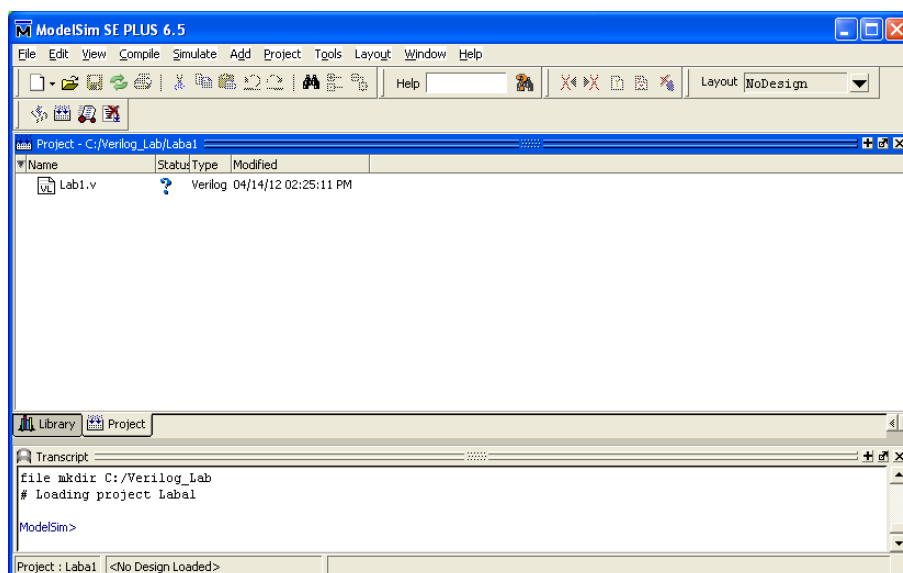
Можливі наступні варіанти файлів, що додаються:

- ✓ Створити новий файл і додати його до поточного проекту (*Create New File*),
- ✓ Додати існуючий файл у проєкт (*Add Existing File*),
- ✓ Виконати моделювання бібліотечного елемента (*Create Simulation*).

Якщо потрібно додати новий файл до проекту, то в діалоговому вікні, що з'явилося, ввести ім'я файла (*File Name*) та вибрати тип цього файлу (*Add file as type*).

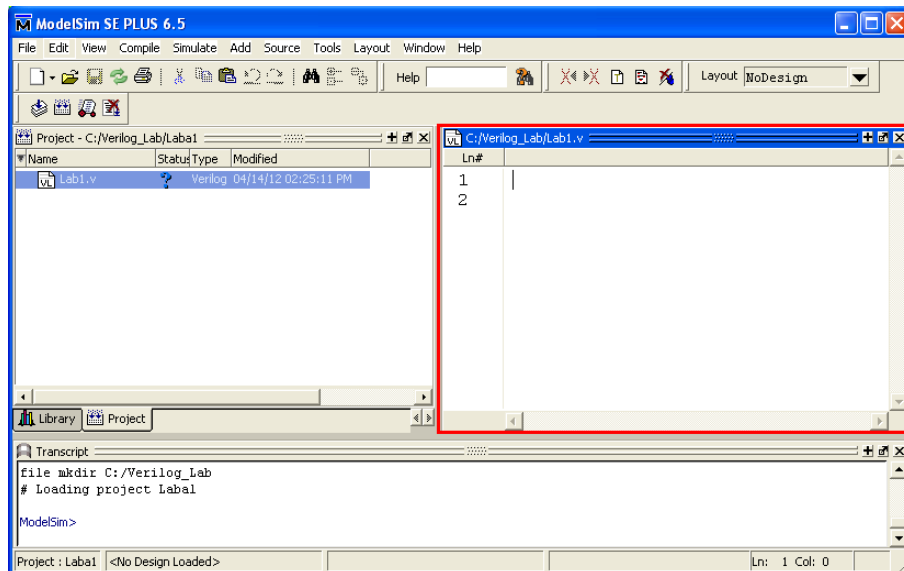


У результаті всіх маніпуляцій ми маємо побачити вікно під назвою **Project** з нашим файлом усередині:



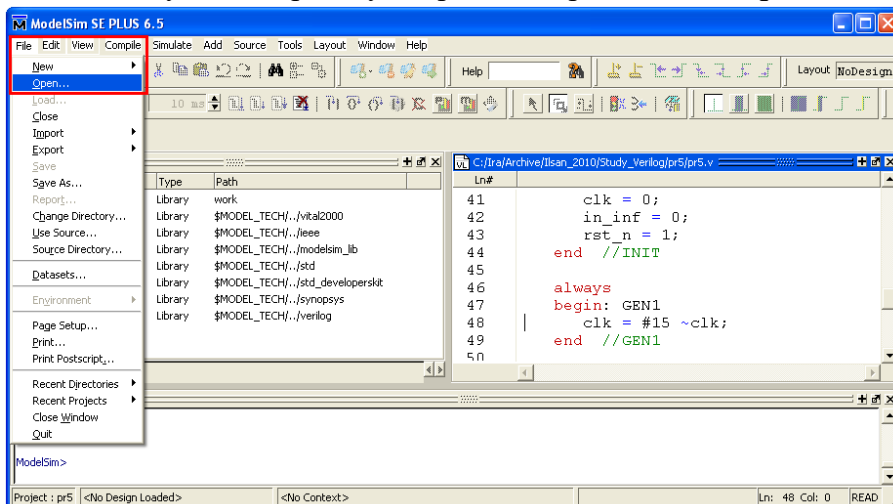
Це вікно є аналогом менеджера проекту, з нього можна керувати файлами проекту та їх компіляцією. У ньому видно тип файлу та його статус, який вказує на те, що є помилки при компіляції, або їх відсутність або те, що файл змінено та не скомпільовано.

Для редагування файлів проекту треба двічі клацнути на імені файлу у вікні **Project**, після чого відкривається вікно **Source**, у якому можна редагувати файл.

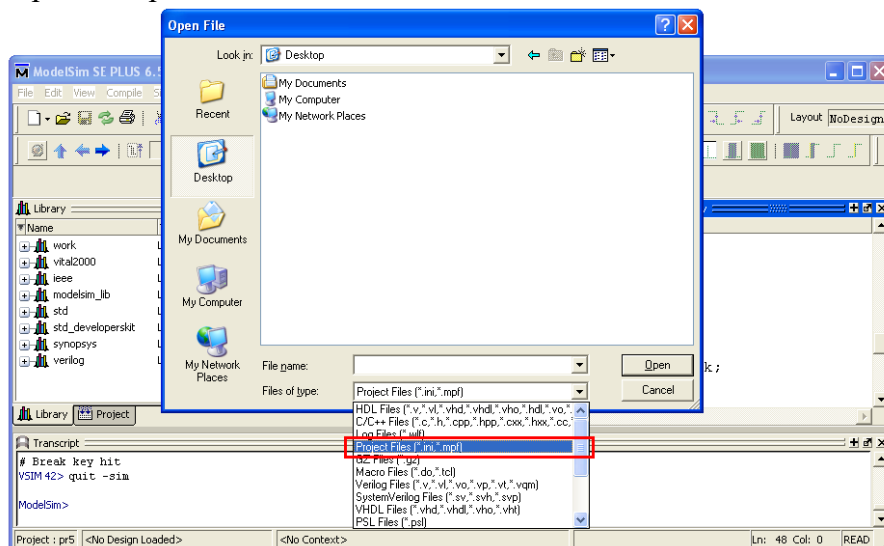


Відкриття існуючих проектів

Для відкриття вже існуючого проекту потрібно вибрати **File** → **Open**



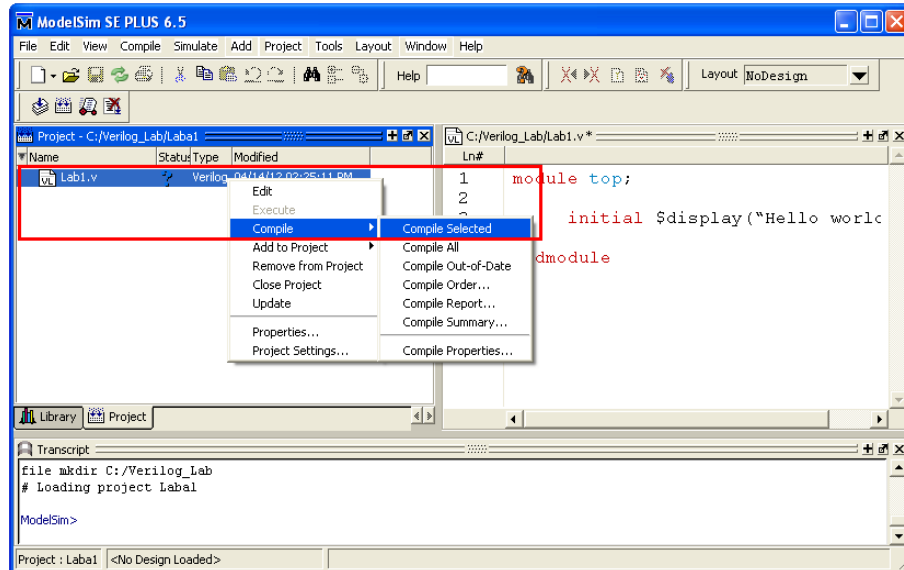
У діалоговому вікні, що відкрилося, в полі **Files of type** вибрати **Project files** та зі списку наявних файлів проекту вибрати потрібний.



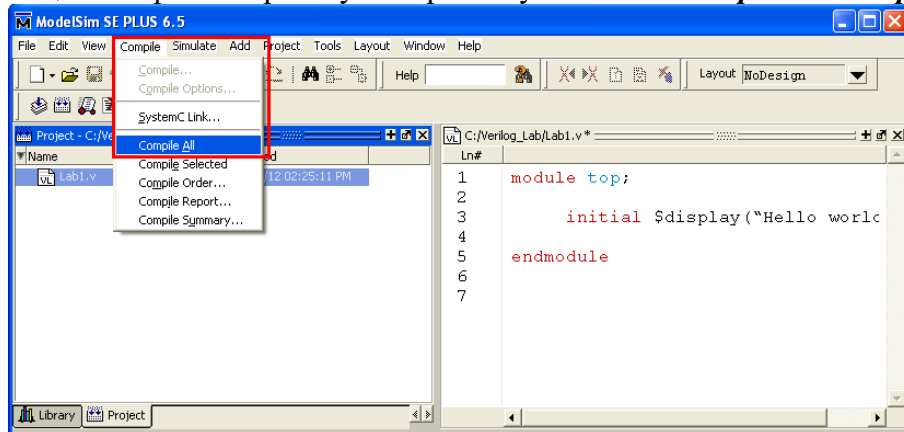
Компіляція програми

Перед компіляцією слід зберегти файл (*File* → *Save* або просто *Ctrl + S*).

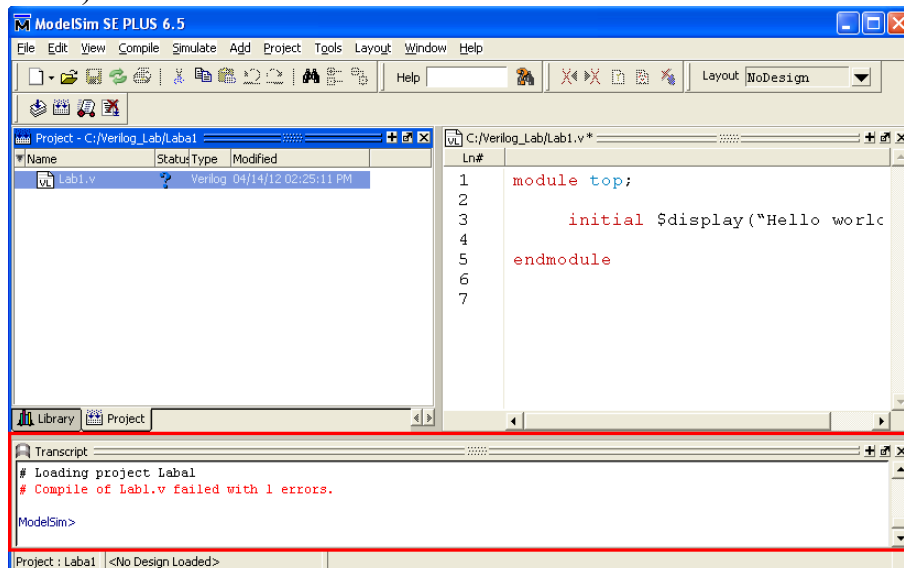
Для компіляції якогось конкретного файлу проекту натискаємо правою кнопкою імені файла, після чого в контекстному меню вибираємо *Compile* → *Compile selected* .



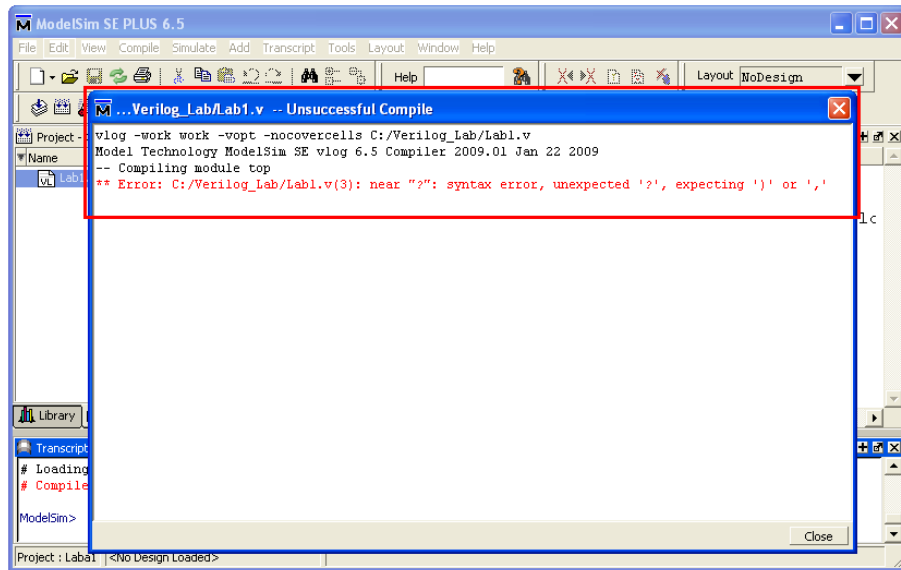
Для компіляції всіх файлів проекту використовуємо меню *Compile* → *Compile All* .



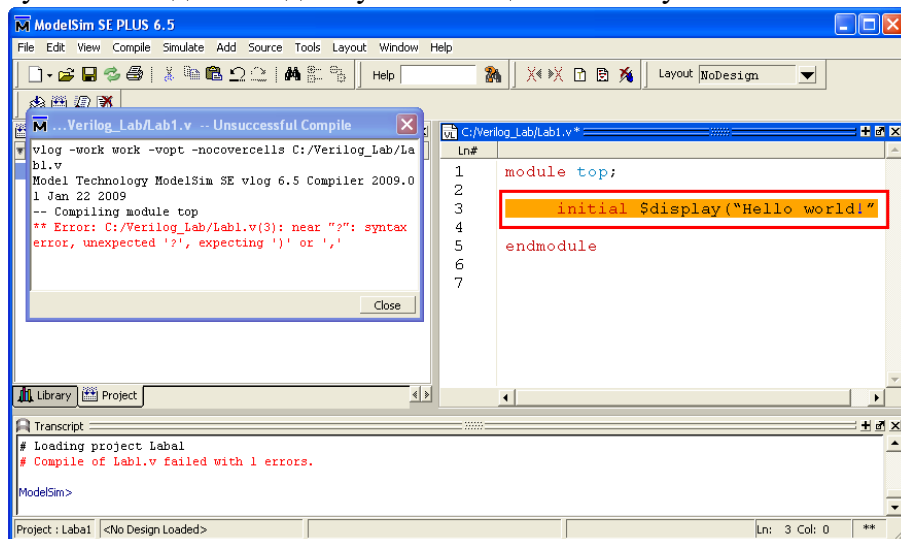
Відразу зміниться статус обраного файлу та у вікні *Transcript* з'явиться повідомлення про помилки (якщо вони є).



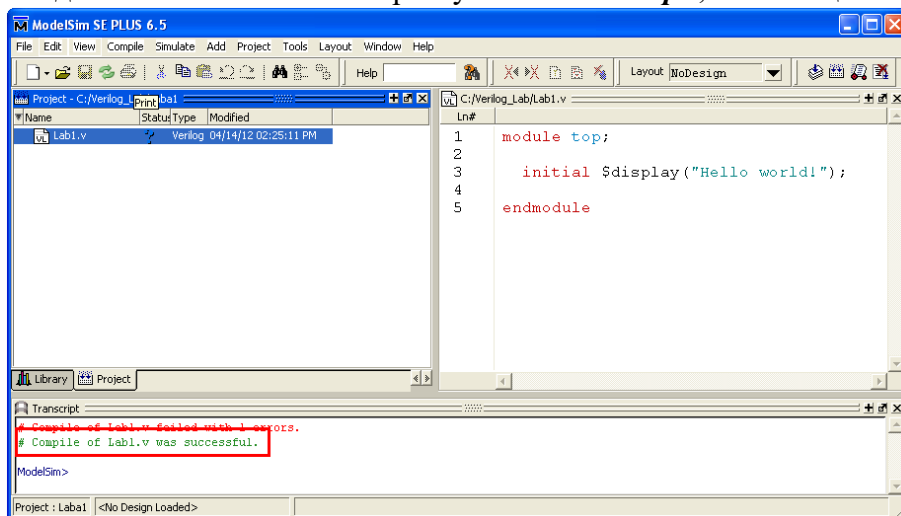
Для того, щоб знайти і виправити помилку, клацаємо двічі на повідомленні і про помилку. Побачимо вікно з описом та місцем помилки.



Клацаємо по опису помилки двічі. Підсвічується місце помилки у вікні *Source*.

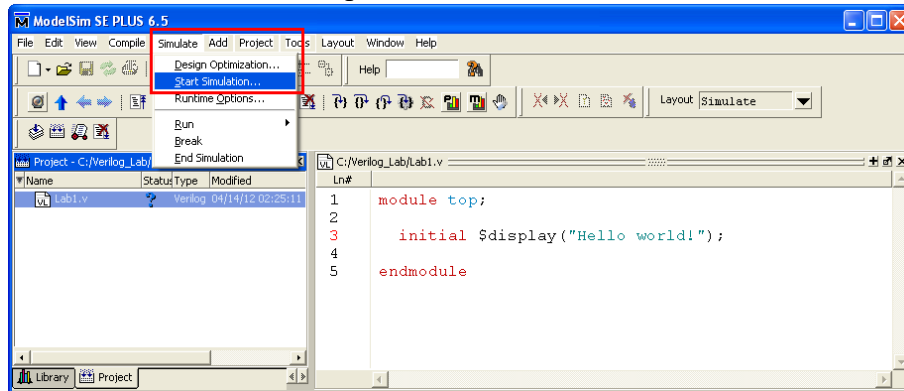


Судячи з опису помилки, є синтаксична помилка. Виправляємо, зберігаємо файл та компілюємо ще раз. Отримуємо повідомлення зеленим кольором у вікні *Transcript*, компіляція виконана успішно.

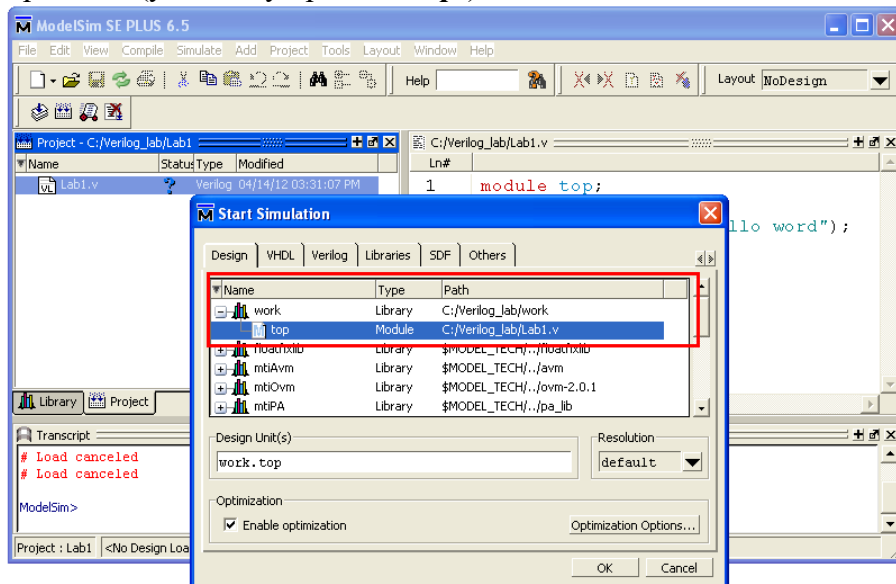


Моделювання роботи програми

Моделювання роботи програми можна запустити кількома шляхами.
Меню *Simulation* → *Start Simulation* відкриє вікно *Start Simulation*.

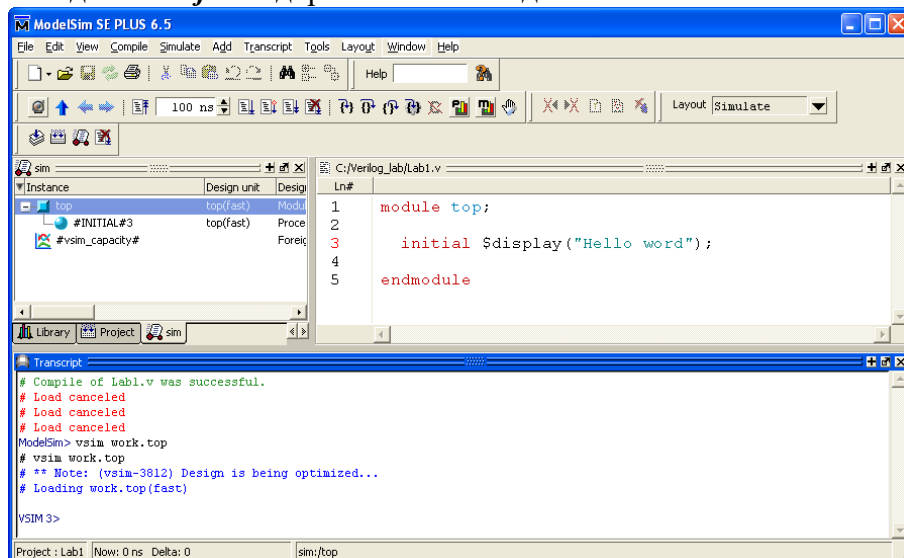


У вікні, в закладці *Design* треба вибрати ту назву бібліотеки, в яку скопіювалися об'єкти проекту (визначається при створенні проекту, зазвичай *work*) і в списку вибрати ім'я модуля, моделювання якого ми хочемо провести (у нашому прикладі *top*).

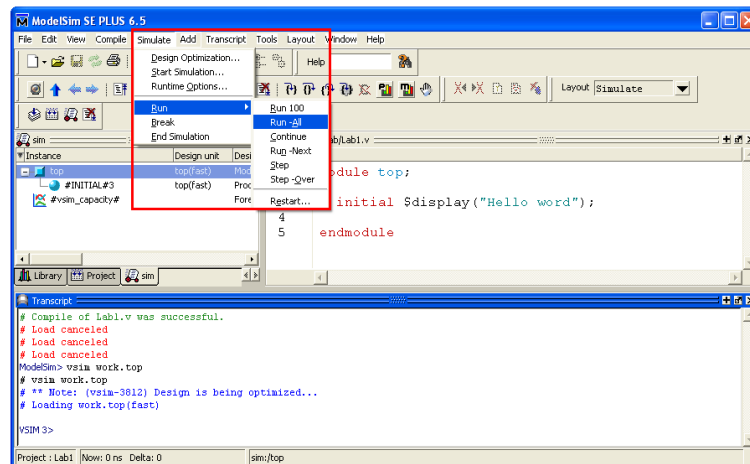


Натискаємо кнопку *OK*.

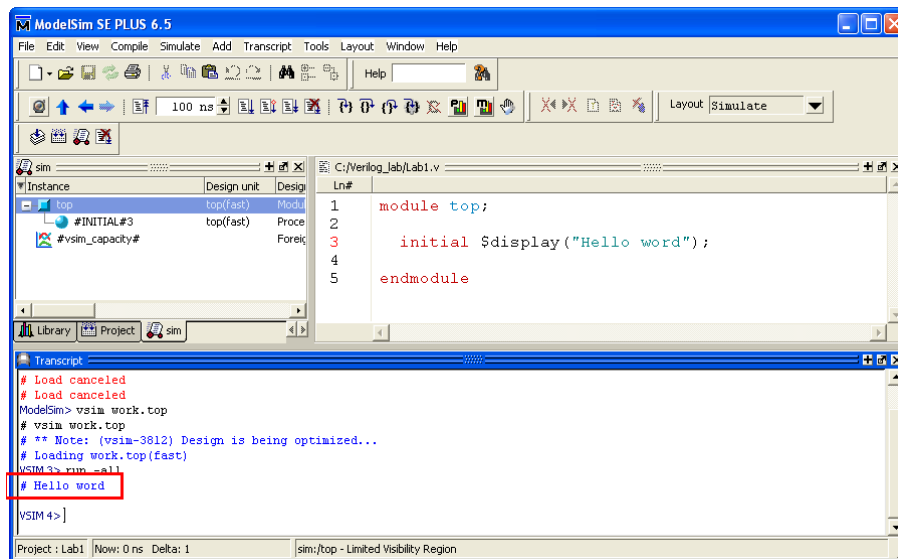
Поруч із закладкою *Project* відкривається закладка *sim*.



Натискаємо меню *Simulation* → *Run* → *Run - All* для того, щоб запустити моделювання на виконання.



Результат виконання у разі нашого прикладу ми бачимо у вікні *Transcript*. У нашому випадку це буде друк слів « *Hello Word!* ».



Загалом, залежно від налаштувань, можна побачити різні вікна. Їх розташування можна зберігати та скидати за допомогою меню *Layout*. Всі вікна можна увімкнути та вимкнути через меню *View*.

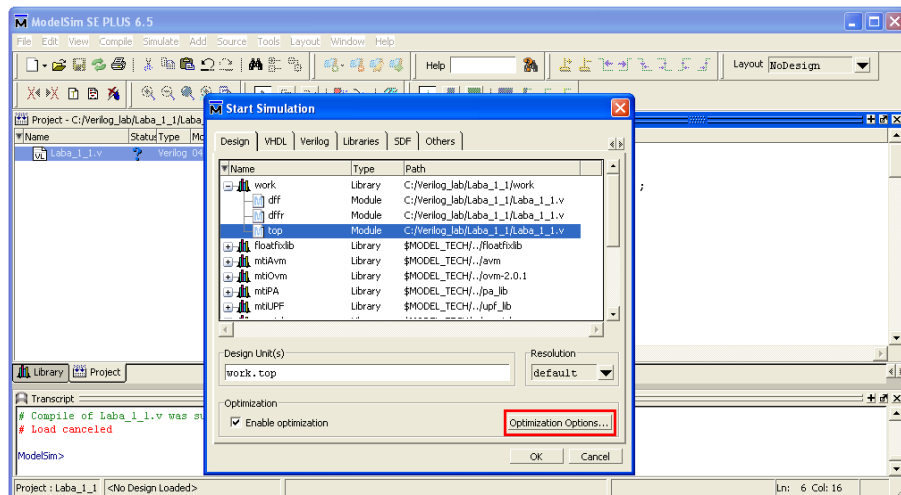
Найбільший інтерес становлять такі вікна:

- ✓ *Wave* – вікно побудови часових діаграм
- ✓ *Structure* – вікно структури проекту, показує ієрархію проекту, при виборі будь-якої структури в решті вікон відобразатимуться її дані
- ✓ *Objects* – вікно сигналів, тут відображаються сигнали вибраного об'єкта проекту
- ✓ *Processes* – показує існуючі процеси та їх статус
- ✓ *Dataflow* – структура процесів, що відображає схему протікання даних

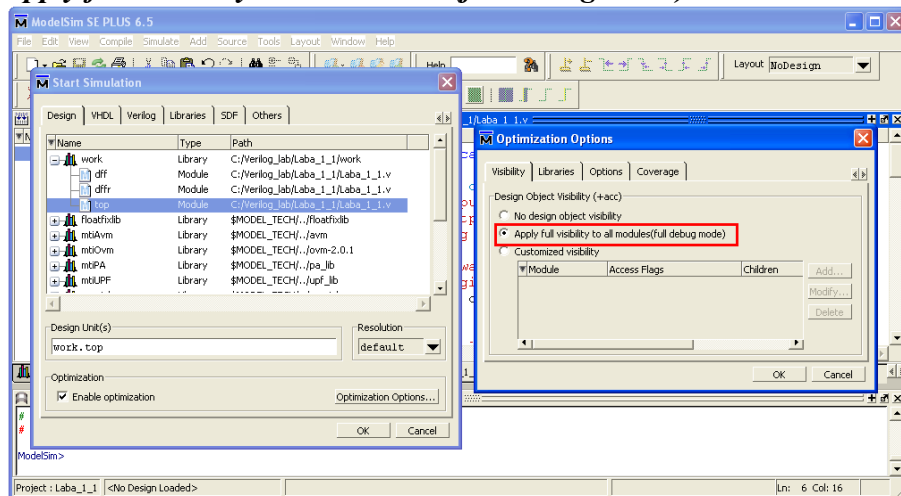
У *ModelSim* підтримується *Drag and Drop*, тому можна перетягувати елементи з одного вікна до іншого.

Розглянемо роботу з цими вікнами на прикладі складнішого проекту.

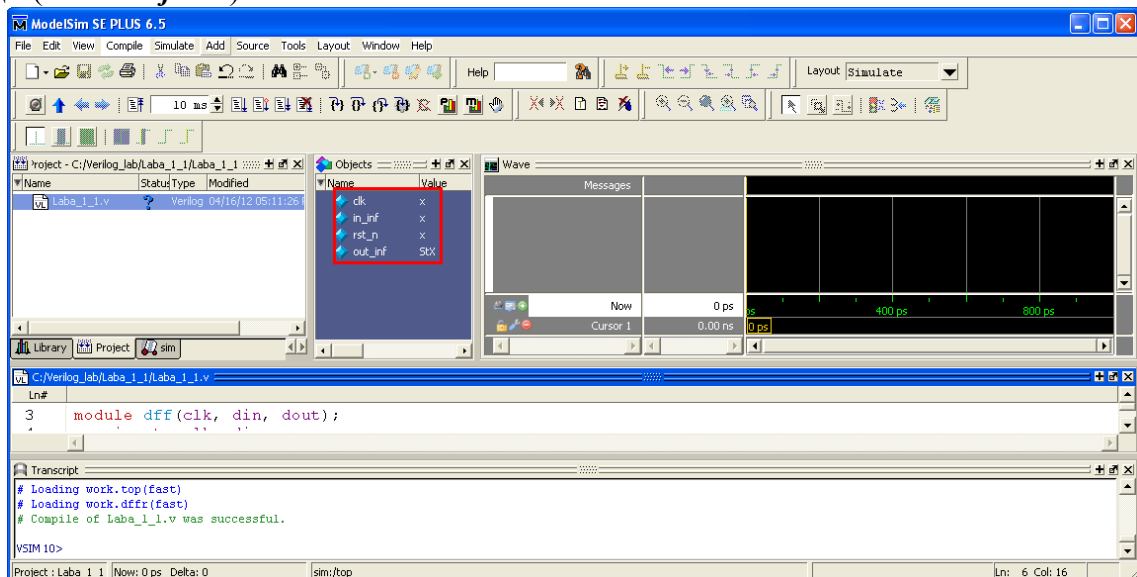
Почнемо моделювання так само, як і в попередньому прикладі. Для прискорення моделювання симулятор використовує оптимізацію, яка може змінювати внутрішню структуру модуля. Щоб побачити внутрішні сигнали модулів, відкриємо вікно *Optimization options ...*



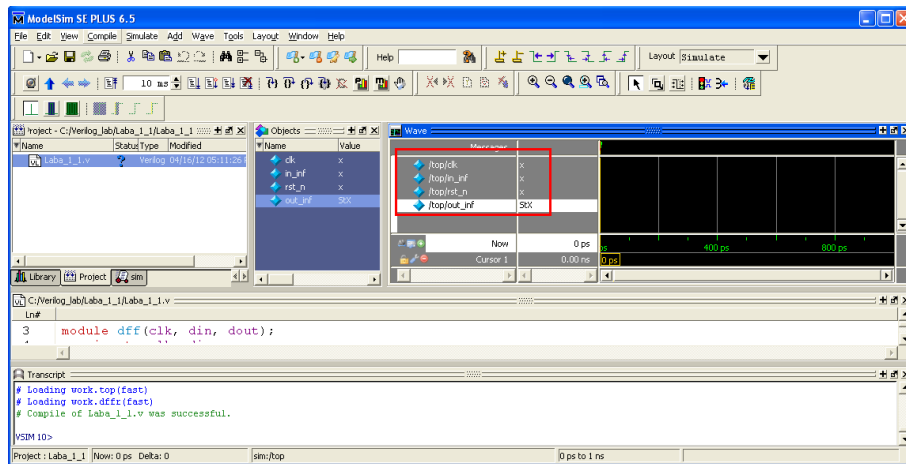
і оберемо опцію *Apply full visibility to all modules (full debug mode)* в закладці *Visibility*.



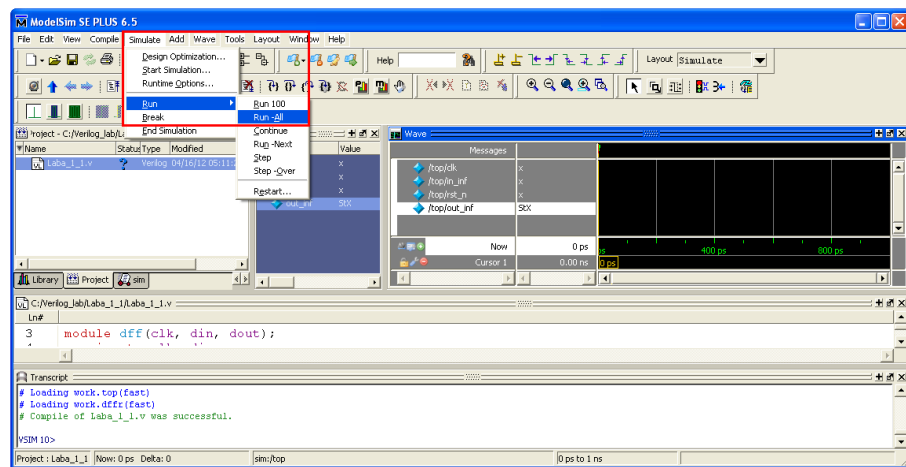
Після натискання ОК з'являться вікна, де можна буде побачити значення сигналів у процесі симуляції (вікно *Objects*).



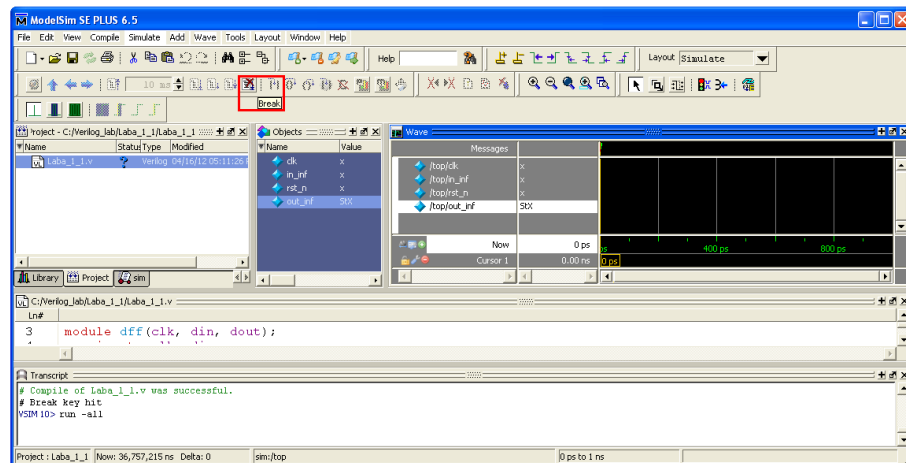
Часові діаграми зміни сигналів можна побачити у вікні *Wave*. Для цього треба «перетягнути» назву сигналів, які ви хочете побачити у вікно *Wave*.



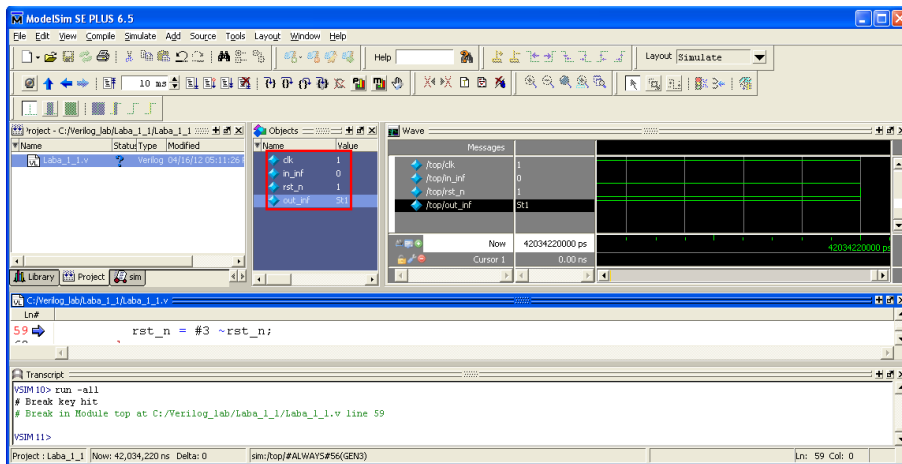
Безпосередньо моделювання, як ви знаєте, починається вибором опцій *Simulation* → *Run* → *Run - All*.



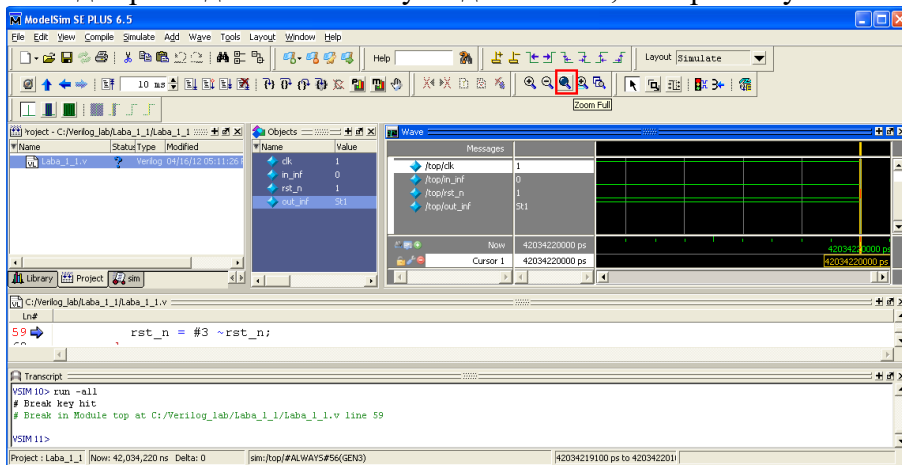
Зупинити моделювання можна або натиснувши кнопку *Break* або вибравши меню *Simulation* → *Break*.



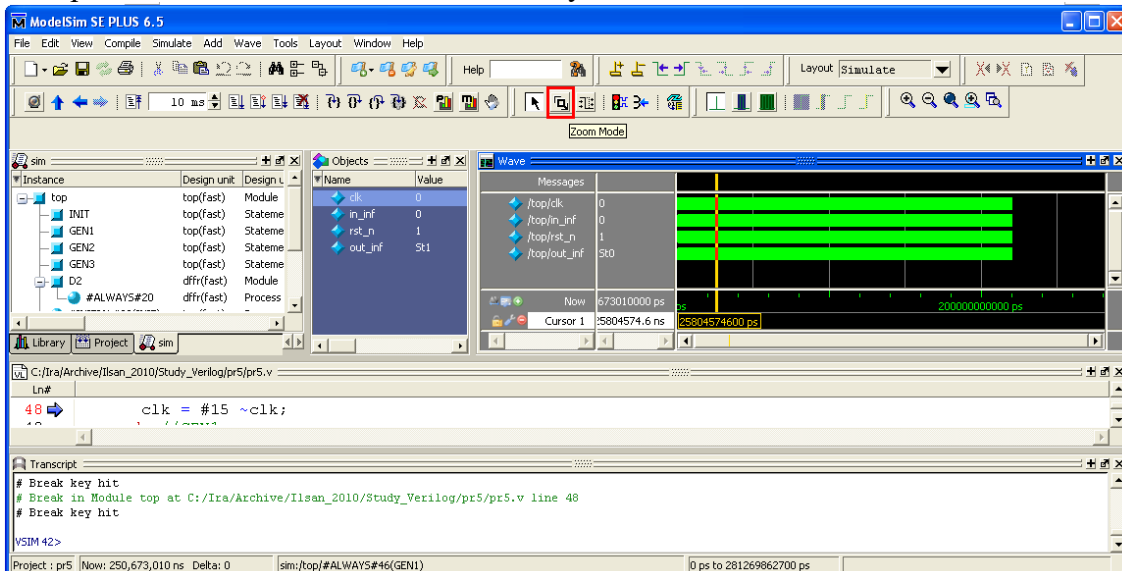
Поточне значення сигналів можна побачити у вікні *Objects*.



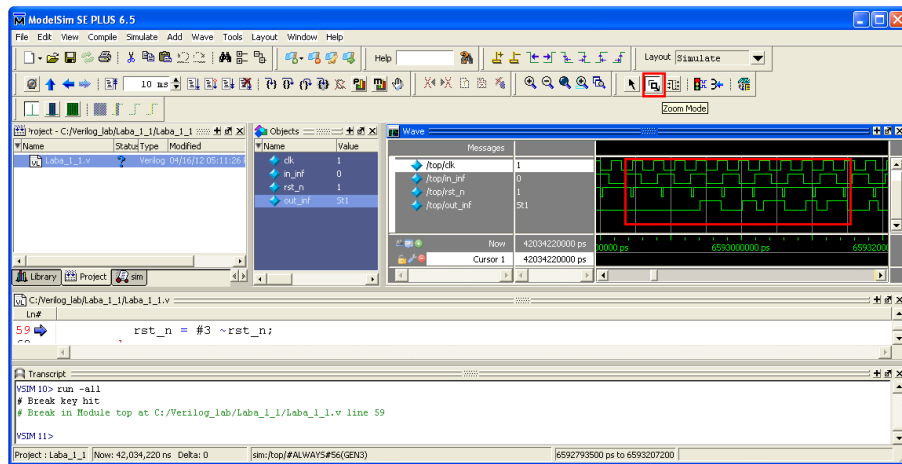
Щоб побачити часові діаграми для всього часу моделювання, використовується кнопка **Zoom Full**.



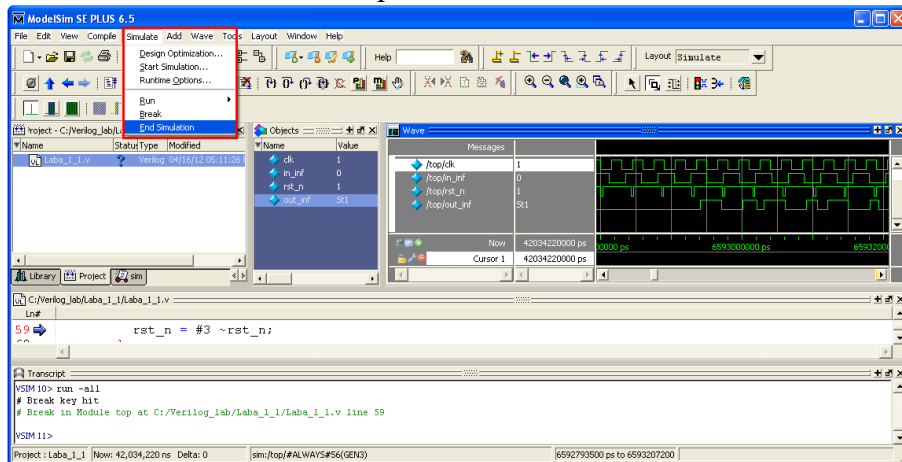
Для того, щоб побачити значення сигналів у певний проміжок часу використовують кнопку **Zoom Mode** та вибирають необхідний часовий діапазон у вікні **Wave** за допомогою миші.



Результат виконання такого масштабування видно нижче.



Завершити режим моделювання можна, вибравши меню *Simulation* → *End Simulation*.



Завдання для виконання

1. Створити проект у середовищі Modelsim.
2. Додати у створений проект файл з наступною програмою та випробувальний стенд до неї:

```

`timescale 1ns/100ps

module dff(clk, din, dout);
input clk, din;
output dout;
reg dout;

always @ (posedge clk)
begin
dout <= din;
end

endmodule

module dffr(clk, din, rst_n, dout);
input clk, din, rst_n;
output dout;
reg dout;

always @ (posedge clk, negedge rst_n)
begin
if (!rst_n)

```

```

dout <= #6 0;
else
dout <= #6 din;
end

endmodule

module top;
reg clk;
reg in_inf;
reg rst_n;
wire out_inf;

//dff D1 (clk, in_inf, out_inf);
dffr D2 (clk, in_inf, rst_n, out_inf);

initial
begin: INIT
clk = 0;
in_inf = 0;
rst_n = 1;
end //INIT

always
begin: GEN1
clk = # 15 ~ clk;
end // GEN1

always
begin: GEN2
in_inf = #23 ~in_inf;
end // GEN1

always
begin: GEN3
if (0 == rst_n)
rst_n = #3 ~rst_n;
else
rst_n = # 40 ~ rst_n;
end // GEN1

endmodule

```

3. Скомпілювати програму та випробувальний стенд до неї.
4. Запустити моделювання програми та випробувального стенду.
5. Продемонструвати часові діаграми роботи випробувального стенду.

Контрольні питання

1. У чому подібність між мовами програмування та мовами опису апаратури?
2. У чому полягає принципова відмінність між мовами програмування та мовами опису апаратури?
3. Які мови опису апаратури ви знаєте?
4. Які з них набули найбільшого поширення в сучасній інженерній практиці?
5. Які можливості щодо моделювання надає середовище Modelsim?

Вимоги до оформлення звіту

Ця лабораторна робота не потребує оформлення протоколу. Для отримання оцінки достатньо продемонструвати такі вміння:

1. Створити проект.
2. Скомпіювати наявну програму та випробувальний стенд до неї.
3. Вміти запустити моделювання наявної програми та випробувального стенду до неї.
4. Вміти подивитися поточний стан сигналів та їх часові діаграми.

Література

1. <http://model.com/>