

Лабораторна робота № 2.4.

Проектування цифрових пристроїв на основі ПЛІС за допомогою програмного описання мовою *AHDL* в середовищі *Quartus II*.

Мета роботи: освоєння технологій проектування логічних елементів на основі ПЛІС шляхом створення програмного описання мовою *AHDL* в середовищі *Quartus II*.

1. Короткі теоретичні відомості

AHDL (мова опису апаратури фірми *Altera*) є високорівневою, модульною мовою, яка повністю інтегрована в систему *Quartus II*. Вона особливо добре підходить для проектування складної комбінаційної логіки, шин, кінцевих автоматів, таблиць істинності та параметричної логіки.

Оператори і елементи *AHDL* є потужним, багатогранним і легким у використанні засобом.

Хоча Ви можете використати текстовий редактор для створення TDF файлів, але тільки текстовий редактор системи *Quartus* надає Вам можливість скористатися його перевагами, коли Ви вводите, компілюєте і налагоджуєте Ваш *AHDL* проект.

AHDL проект легко вставити в ієрархію проекту. У текстовому редакторі Ви можете автоматично створювати символ, який представляє TDF файл і вводити його в графічний файл проекту (*Graphic Design File (.gdf)*). Ви можете також перевірити синтаксис і виконати повну компіляцію для налагодження та прогону Вашого проекту. Будь-які помилки автоматично локалізуються процесором повідомлень (*Message Processor*) і виділяються у вікні текстового редактора.

Зарезервовані ключові слова мови *AHDL*

Зарезервовані ключові слова використовуються для управління операторами *AHDL*, а також для констант *GND* і *VCC*.

Зарезервовані ключові слова відрізняються від зарезервованих ідентифікаторів тим, що ключові слова можна використовувати як символічні імена при укладанні їх в одиночні лапки ('), в той час як зарезервовані ідентифікатори не можна.

Altera рекомендує вводити всі ключові слова з великих літер для зручності читання.

Зарезервовані ключові слова мови *AHDL*

Таблиця 1

AND	FUNCTION	OUTPUT
ASSERT	GENERATE	PARAMETERS
BEGIN	GND	REPORT
BIDIR	HELP_ID	RETURNS
BITS	IF	SEGMENTS
BURIED	INCLUDE	SEVERITY

CASE	INPUT	STATES
CLIQUE	IS	SUBDESIGN
CONNECTED_PINS	LOG2	TABLE
CONSTANT	MACHINE	THEN
DEFAULTS	MOD	TITLE
DEFINE	NAND	TO
DESIGN	NODE	TRI_STATE_NODE
DEVICE	NOR	VARIABLE
DIV	NOT	VCC
ELSE	OF	WHEN
ELSIF	OPTIONS	WITH
END	OR	XNOR
FOR	OTHERS	XOR

2. Порядок виконання роботи

2.1. Вимоги до устаткування і програмного забезпечення

Лабораторна робота виконується на ПК з використанням програми *Quartus II*.

Системні вимоги

Платформа: *Windows XP(Professional or Home) or Windows 2000 Professional*

- 2 ГГц Pentium 4 процесор або еквівалентний
- 1 ГБ ОЗУ
- 2 ГБ простору жорсткого диска

Монітор роздільною здатністю 1280×1024, 32-бит кольору, 64 МБ ОЗУ відео карту

2.2. Створення нового проекту

В даній лабораторній роботі можна використовувати проект, створений у попередній роботі, або створювати новий з іншим ім'ям, за описом п. 2.2. (Лаб. роб. №2.3). Обрати той самий тип ПЛІС з найменшою кількістю вентилів та виходів і завершити її налаштування, натиснувши *Finish* (Закінчити).

2.3. Проектування логічного елементу

2.3.1. Створення *AHDL* файлу

Після створення проекту необхідно створити *AHDL* файл, для цього виконати команду *File > New* (Файл > Новий) і у діалоговому вікні *New* (рис.1) на вкладці *Design Files* (Проектування файлу) вибрати тип файлу *AHDL File* (*AHDL* файл) і натиснути *OK*.

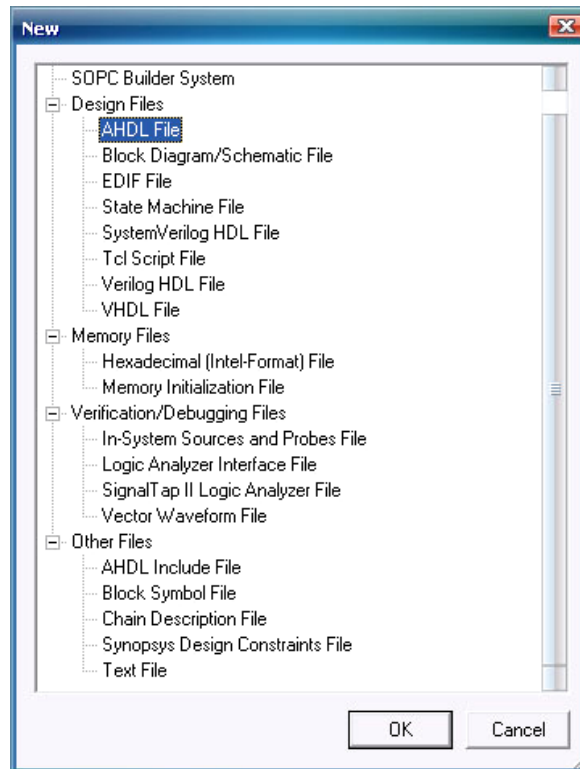


Рис.1. Створення *AHDL* файлу

Після створення даного файлу необхідно відразу його зберегти, надавши йому певне ім'я командою *File > Save As...* (Файл > Зберегти як...). Наприклад *Element.tdf*.

Для прикладу розглянемо наступне рівняння:

$$F = \overline{(IN1 \wedge IN2)} \vee (IN3 \oplus IN2) \vee (\overline{(IN1 \oplus IN2)} \wedge (IN1 \vee IN2)) \vee (\overline{(IN1 \oplus IN3)} \vee IN1)$$

На основі даного рівняння сформуємо файл опису *AHDL*:

```
subdesign Element
(
  IN1 : input;
  IN2 : input;
  IN3 : input;

  OUT1: output;
  OUT2: output;
  OUT3: output;
)
begin
  OUT1 = (not(IN1 and IN2)) or (IN3 xor IN2);
  OUT2 = (not(IN1 xor IN3)) and (IN1 or IN2);

  OUT3=(OUT1 or OUT2) or (OUT2 or IN1);
end;
```

Для визначення файлу *Element.tdf* найвищим в ієрархії у вікні *Entity* (Об'єкт) панелі *Project Navigator* клацнути ПКМ на назві файлу *Element*, у

випадаючому меню обрати *Setting...* (Налаштування) та в категорії *General* (Головне) визначити файл у вікні *Top-level entity* (Об'єкт вищого рівня) (рис.2).

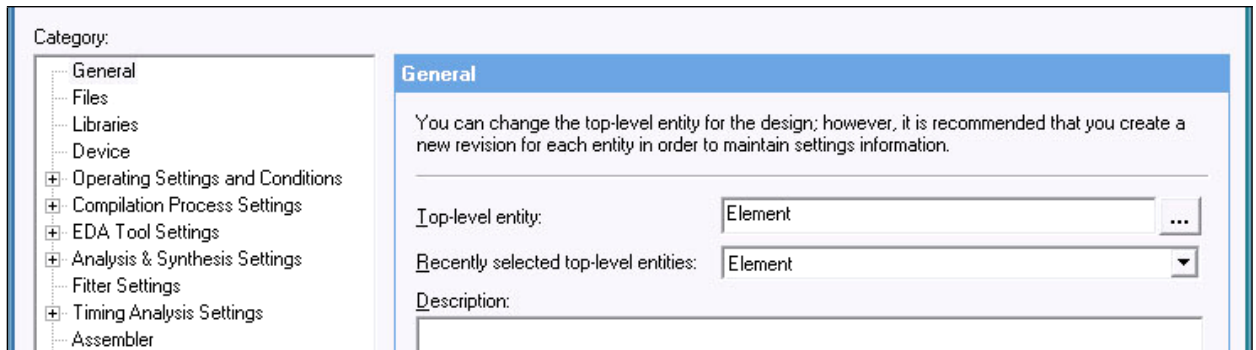



Рис.2. Визначення ієрархії файлів

Далі необхідно провести компіляцію файлу. Для цього натиснути на іконку  – *Start Compilation* (Початок компіляції) в верхній панелі елементів, або обрати команду *Processing > Start Compilation* (Обробка > Початок компіляції). В результаті отримаємо звіт, про кількість затрачених логічних вентилів, та пінів ПЛІС (рис.3).

2.3.2. Підготовка процесу симуляції

Для перевірки правильності функціонування логічної схеми потрібно виконати функціональне моделювання. Тому треба створити файл моделювання для відображення форм сигналів обравши пункт меню *File > New* (Файл > Новий) і після цього в діалозі *Verification / Debugging Files* (Перевірка / Інструментальний файл) > *Vector Waveform File* (Файл векторної форми представлення) (рис.4).

Flow Summary	
Flow Status	Successful - Mon Apr 17 15:08:13 2017
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Web Edition
Revision Name	4
Top-level Entity Name	Element
Family	Cyclone II
Met timing requirements	Yes
Total logic elements	2 / 4,608 (< 1 %)
Total combinational functions	2 / 4,608 (< 1 %)
Dedicated logic registers	0 / 4,608 (0 %)
Total registers	0
Total pins	6 / 158 (4 %)
Total virtual pins	0
Total memory bits	0 / 119,808 (0 %)
Embedded Multiplier 9-bit elements	0 / 26 (0 %)
Total PLLs	0 / 2 (0 %)
Device	EP2C5F256C6
Timing Models	Final

Рис.3. Звіт компіляції файлу

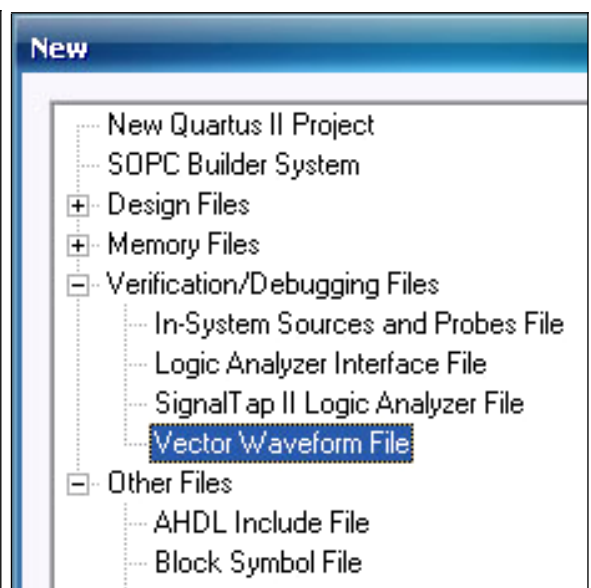


Рис.4. Створення файлу моделювання

Далі необхідно додати вхідні та вихідні сигнали комбінаційної логічної схеми, для цього клацнути правою клавiшею миші на лiвiй панелi *Name* (Им'я) i в випадiючому меню обрати пункт *Insert / Insert Node or Bus* (Установка/Установити вузол або шину) (рис.5).

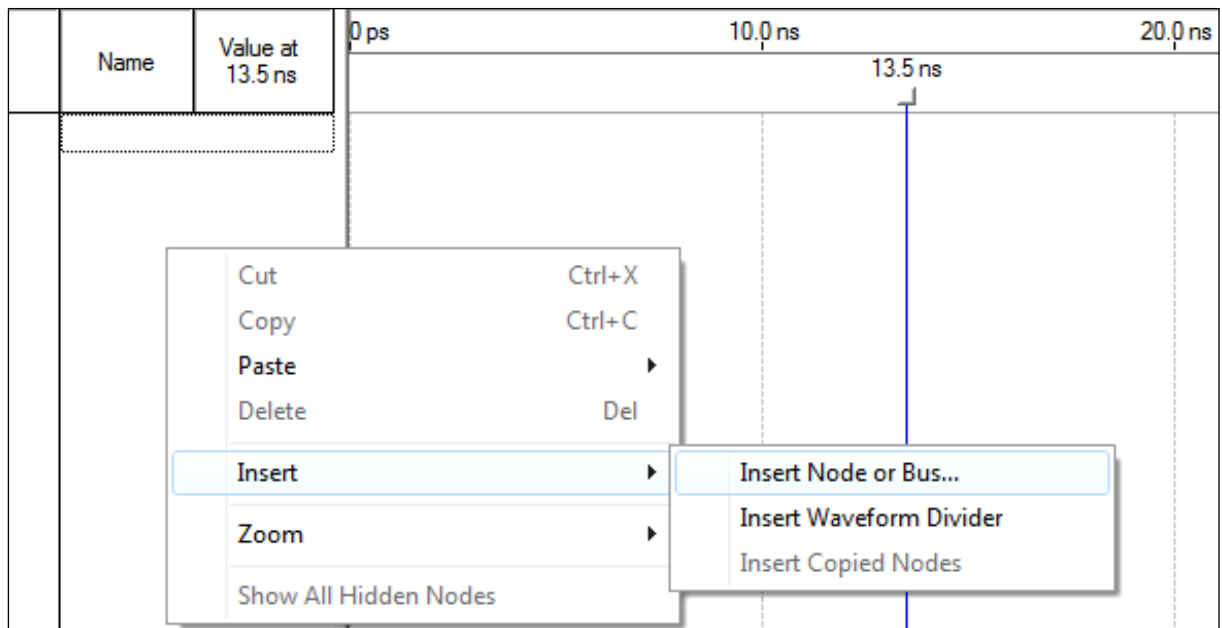


Рис.5. Вибір необхідних сигналів комбінаційної схеми

У діалоговому вікні, що з'явилося (рис. 6) натиснути кнопку *Node Finder* (Пошук сигналу), з'являється діалогове вікно для пошуку сигналів у проєкті, з натисканням кнопки *List* (Список) з'явиться список входів і виходів у вікні *Node Finder* (Пошук сигналу). Транспортувати за допомогою кнопки >> усі сигнали входів та виходів для початку моделювання у вікно *Selected Nodes* (Обрані сигнали) (рис. 7). Натиснути кнопку *OK*.

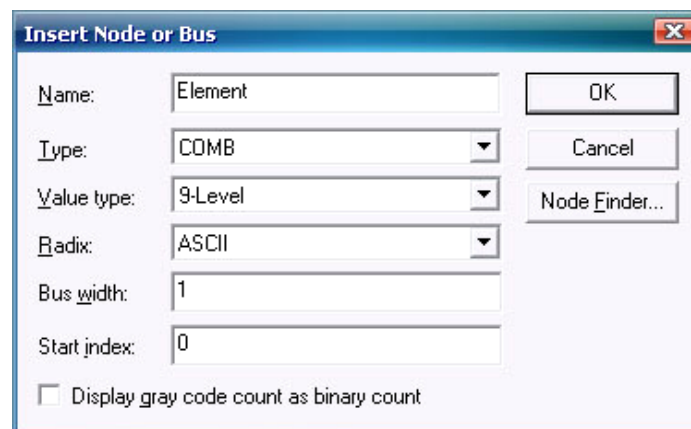


Рис. 6

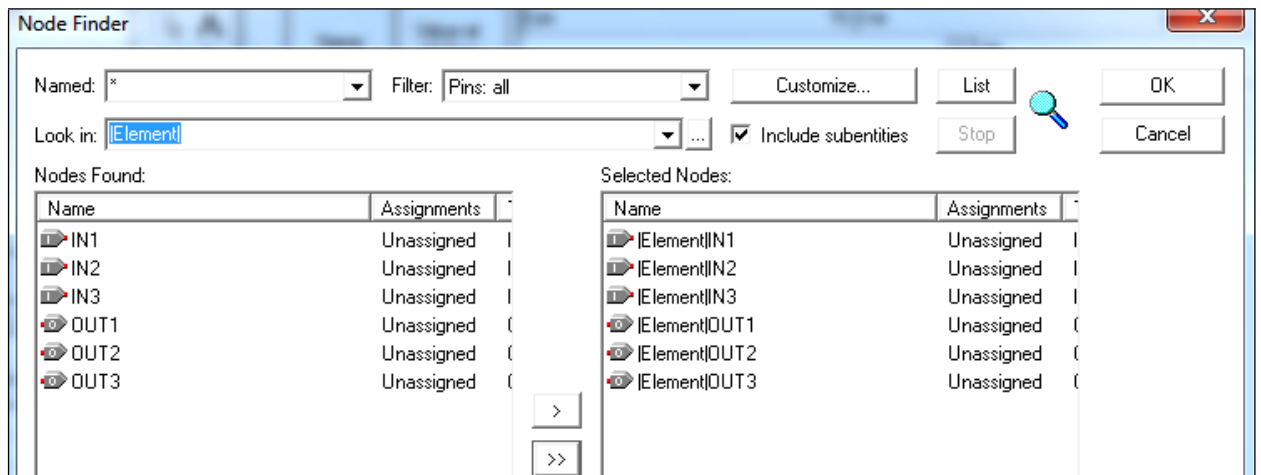


Рис.7. Експорт сигналів комбінаційної схеми

Визначити час симуляції, для цього відкрити меню *Edit > End Time* (Правка > Час закінчення) і встановити час рівним 1мкс (рис.8).

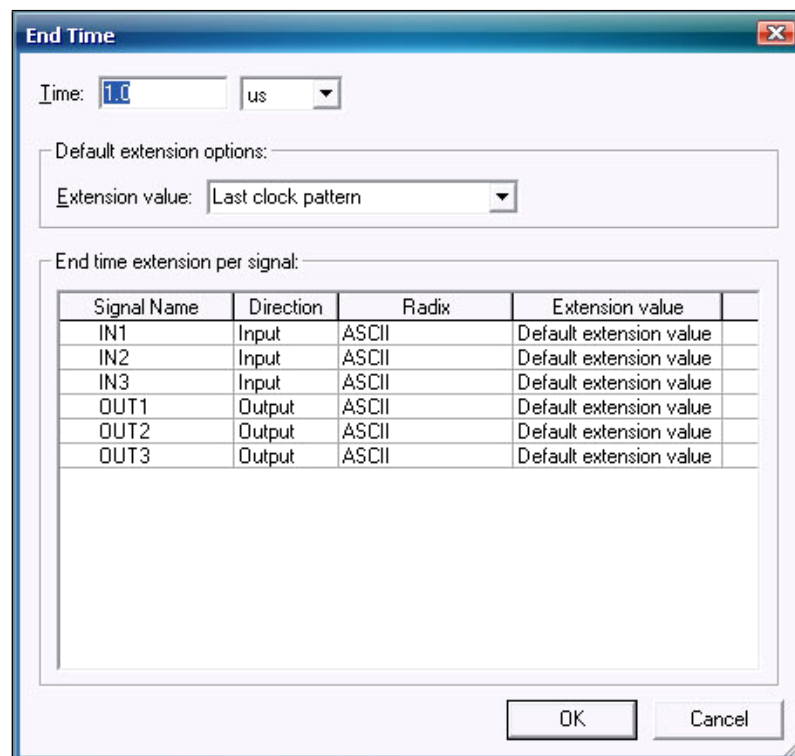


Рис. 8. Налаштування часу симуляції

Задати часові інтервали вхідних сигналів, для цього вибрати сигнал *IN1* і натиснути на панелі інструментів кнопку *Overwrite Clock* (Встановлення годинника) і задати період тактової частоти 10ns (рис.9). Аналогічним чином встановити для *IN2* – 20 ns, а для *IN3* – 40 ns.

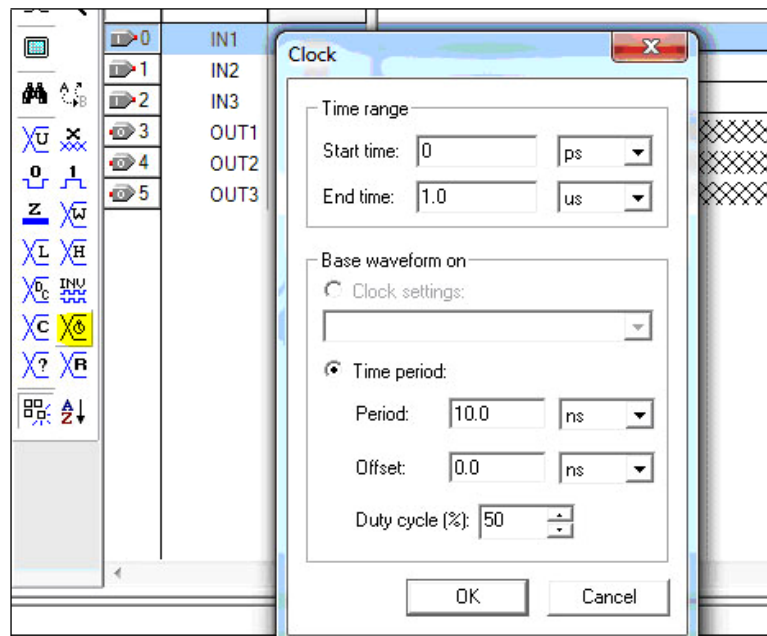


Рис.9. Налаштування інтервалів вхідних сигналів

Задати тип симуляції в пункті меню *Assignment > Settings...* (Призначення > Налаштування). У діалоговому вікні *Settings – lab1*, що відкрилося, у розділі *Simulator Settings* (Налаштування симуляції) потрібно задати тип симуляції – *Functional* (Функціональна) (рис.10).

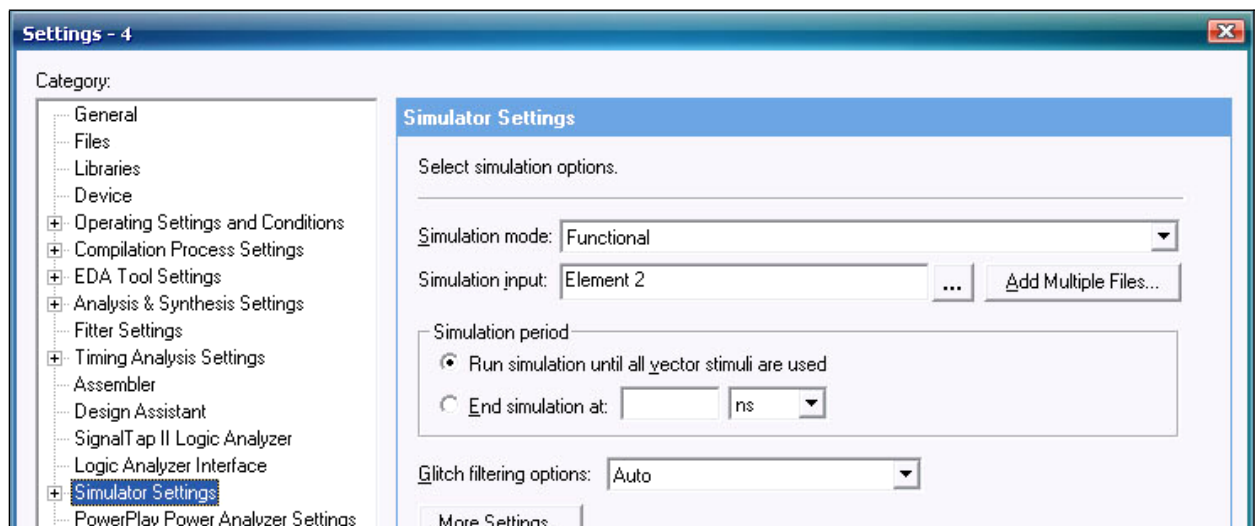


Рис.10. Налаштування режиму симуляції

2.3.3. Проведення симуляції

Перш за все необхідно створити сценарій симуляції (рис. 11).

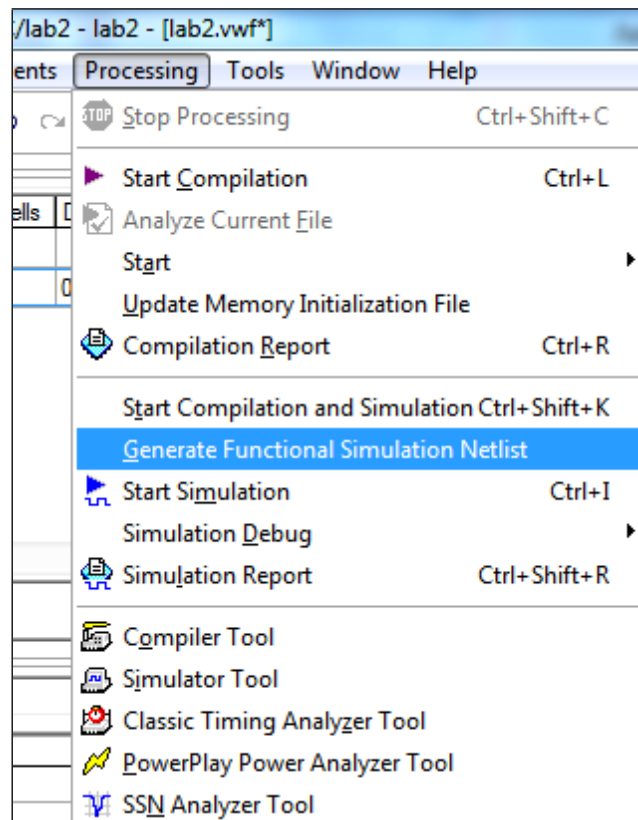


Рис. 11. Створення сценарію симуляції

Запустити симулятор з пункту меню *Processing* > *Start Simulation* (Обробка > Початок симуляції) (рис.12). Після успішної симуляції можна розглянути сигнали, які цікавлять.

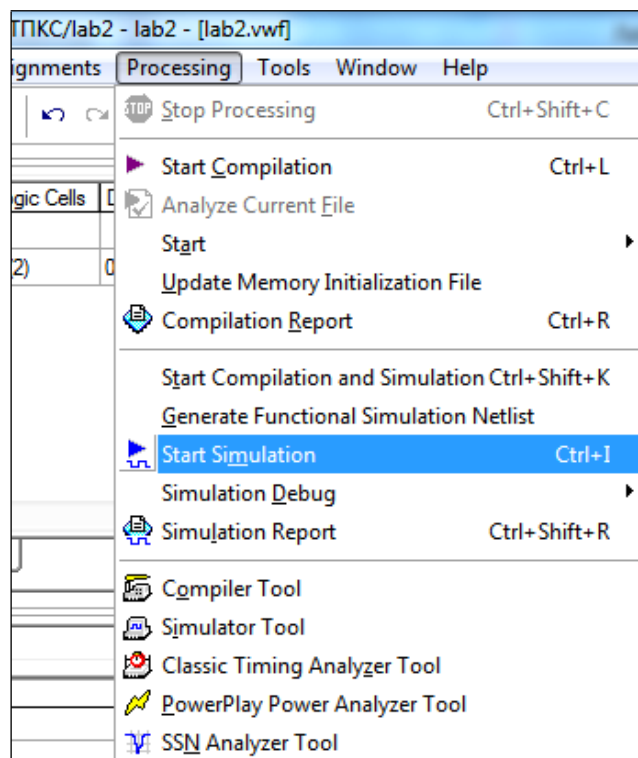


Рис.12. Запуск процесу симуляції

Результат виконання процесу симуляції надано на рис.13.

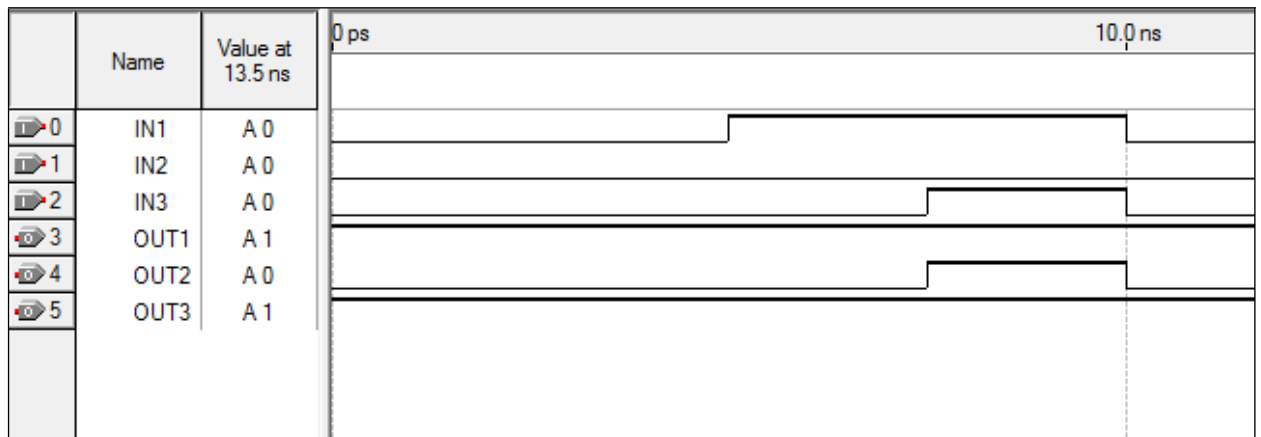


Рис.13.

2.4. Виведення результату проектування на друк

2.5. Вимоги до звіту з лабораторної роботи

Звіт про роботу виконується у вигляді альбому технічної документації згідно з вимогами ГОСТу.

Звіт складається з:

- титульної сторінки з позначенням прізвища, групи, номера залікової книжки та варіанта;
- цілі роботи;
- опису основних етапів виконання роботи, результатів, одержаних в процесі виконання роботи та необхідних пояснень;
- висновків по роботі;
- графічної частини.

Для захисту звіту має бути представлено іменний файл проекту розробки у програмі *Quartus II*.

2.6. Завдання до лабораторної роботи

Описати комбінаційну схему за допомогою мови *AHDL* за варіантом логічного рівняння.

Варіант	Завдання
1	$y = ((x1 \wedge x2 \wedge x3) \vee (x4 \wedge x1)) \wedge x1$
2	$y = (x1 \wedge x2) \vee x3 \vee x4 \wedge x2$
3	$y = x1 \vee x2 \oplus (x3 \vee x4) \vee x1$
4	$y = (x1 \oplus x2) \vee (x3 \wedge x4)$
5	$y = (x1 \vee x4) \wedge ((x2 \vee x3) \oplus x1)$
6	$y = (x1 \wedge x2 \wedge x3) \oplus (x1 \vee x2) \oplus (x1 \wedge x4)$
7	$y = (x1 \wedge x2) \vee (x3 \oplus x2) \vee (x1 \wedge x3)$
8	$y = (x2 \wedge x1 \vee x3) \vee (x2 \vee x1) \wedge (x1 \oplus x3)$

9	$y = \overline{(x1 \wedge x2 \wedge x3)} \wedge (x1 \wedge x3) \oplus (x1 \vee x3)$
10	$y = (x2 \wedge x3) \oplus (x1 \oplus x2) \wedge \overline{(x2 \wedge x1)} \vee x1$

Список літератури

1. Язык описания цифровых устройств Altera *AHDL* (Практический курс). Издание 2-е, стереотипное. А.П. Антонов. М.: ИП Радио Софт 2002;
2. Цифровая схемотехника. Угрюмов Е.П. СПб.: БХВ-Петербург, 2002;
3. Системы автоматизированного проектирования фирмы Altera MAX+Plus II и *Quartus* II. Краткое описание и самоучитель. Комолов Д.А., Мьяльк Р.А., Зобенко А.А., Филлипов А.С. М.: ИП ИП Радио Софт 2002;