

## Лабораторна робота № 2.3.

### Проектування цифрових пристроїв на основі програмованих логічних інтегральних схем (ПЛІС) в середовищі *Quartus II*

**Мета роботи:** Ознайомлення з можливостями середовища проектування ПЛІС *Quartus II*. Оволодіння технологіями проектування простих логічних елементів на основі ПЛІС в середовищі *Quartus II*.

#### 1. Короткі теоретичні відомості

Програмовані логічні інтегральні схеми (ПЛІС) є одними з найперспективніших елементів цифрової схемотехніки. ПЛІС випускається за двома основними технологіями:

- Програмовані логічні матриці;
- CPLD – набір універсальних цифрових блоків;
- FPGA – вентильні матриці;

ПЛІС являє собою кристал, що складається з двох великих частин. Одна частина велику кількість простих логічних елементів. Спочатку ці елементи не з'єднані між собою. З'єднання елементів (перетворення розрізнених елементів в електричну схему) здійснюється за допомогою електронних ключів, розташованих у цьому ж кристалі. Електронні ключі управляються спеціальною пам'яттю, в осередки якої заноситься код конфігурації цифрової схеми. Таким чином, записавши в пам'ять ПЛІС певні коди, можна зібрати цифровий пристрій будь-якого ступеня складності (це залежить від кількості елементів на кристалі і параметрів ПЛІС). На відміну від мікропроцесорів, в ПЛІС можна організувати алгоритми цифрової обробки на апаратному рівні. При цьому швидкодія цифрової обробки різко зростає.

Перевагами технології проектування пристроїв на основі ПЛІС є:

- мінімальний час розробки схеми (потрібно лише занести в пам'ять ПЛІС конфігураційний код);
- на відміну від звичайних елементів цифрової схемотехніки тут відпадає необхідність у розробці та виготовленні складних друкованих плат;
- швидке перетворення однієї конфігурації цифрової схеми в іншу (заміна коду конфігурації схеми в пам'яті);
- для створення пристроїв на основі ПЛІС не потрібно складне технологічне виробництво. ПЛІС конфігурується за допомогою персонального комп'ютера розробника.

Одним зі світових лідерів з виробництва ПЛІС є фірма *Altera*. Для створення цифрових пристроїв на основі своїх виробів *Altera* розробила спеціальну програмну середу *Quartus II*. Це середовище дозволяє:

- за допомогою графічного редактора ввести в пам'ять персонального комп'ютера електричну схему;
- перевірити і виправити помилки;
- визначити параметри і характеристики розробленого пристрою;

- сформувати файл конфігурації для конкретної ПЛІС;
- завантажити цей файл в пам'ять інтегральної схеми.

## 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 проекту в середовищі *Quartus II*

Робота в середовищі *Quartus II* починається із створенням проекту. Перш за все, при виконанні лабораторної роботи кожен студент має створити іменну теку для зберігання файлів проекту.

Запустити програму *Quartus II*. Вибрати з меню *File > New Project Wizard...* (Файл > Майстер створення нових проектів). У вікні, натиснути кнопку *Next* (Наступний) і потрапляємо у вікно для завдання поточної директорії проекту. Заповнити три рядки як, показано на рис.1.

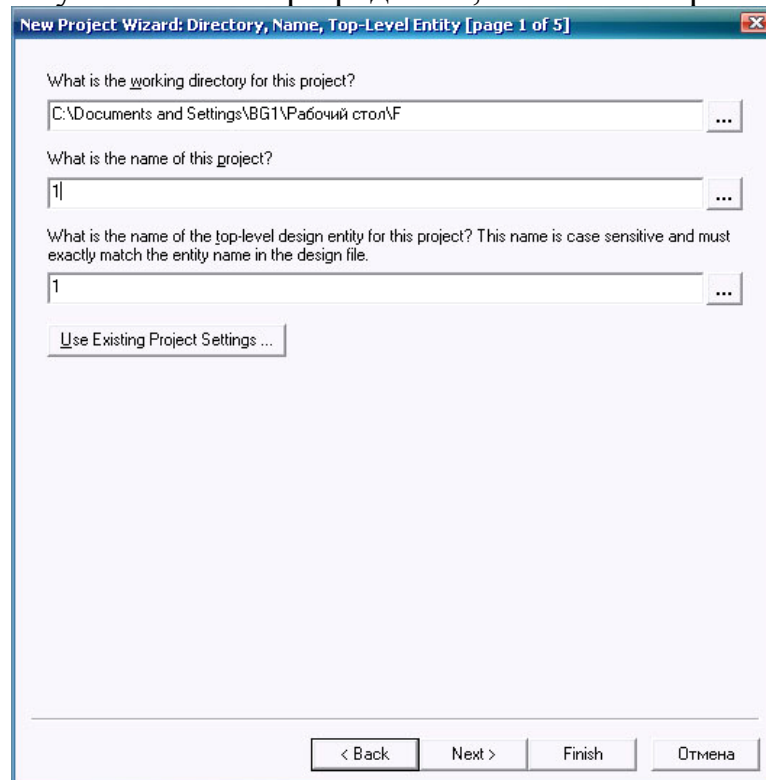


Рисунок 1. Створення нового проекту

У даному випадку поточний проект буде названий «1». Проект буде створений в теці «F» на робочому столі. Натиснути кнопку *Next* (Наступний). На наступній вкладці *New Project Wizard: Add File [page 2 of 5]* нічого обирати не треба оскільки тека нового проекту не містить файлів для використання, тут також натиснути кнопку *Next* (Наступний).

На наступній вкладці необхідно вибрати, тип ПЛІС. Оскільки нам не потрібно великої кількості логічних вентилів для реалізації лабораторної роботи, то логічно обрати ПЛІС з найменшою кількістю вентилів та виходів, див. рис.2. Після цього натиснути *Finish* (Закінчити), налаштування проекту завершено.

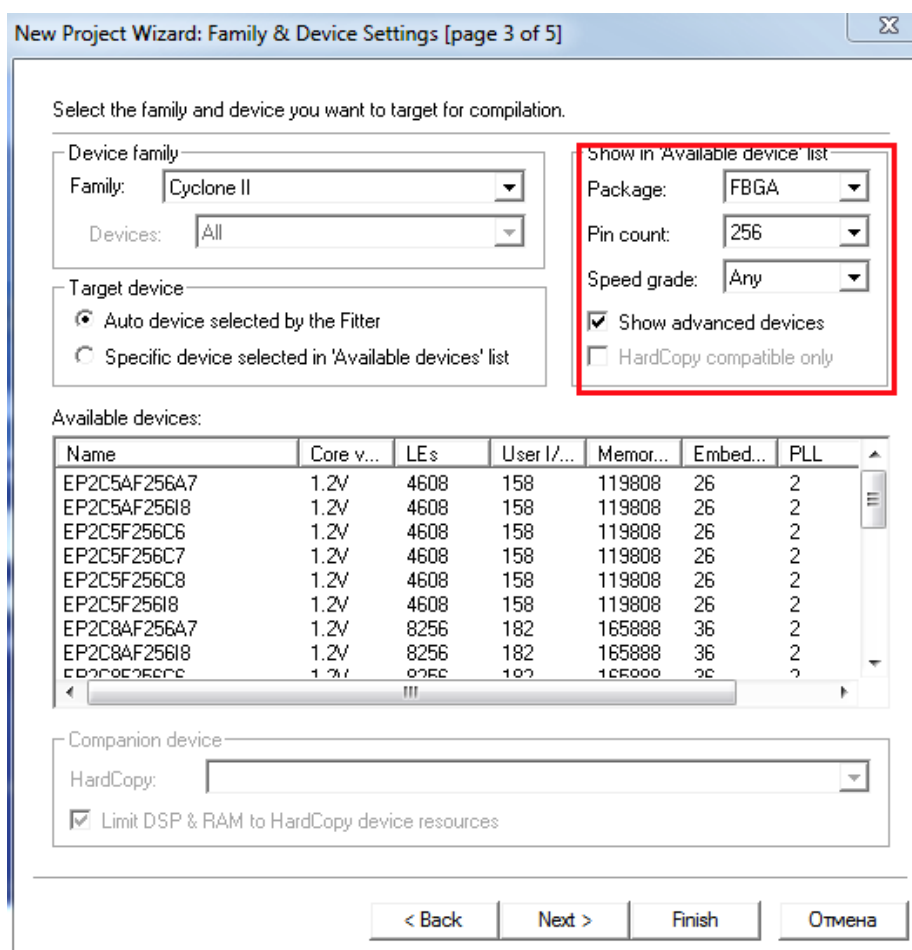


Рисунок 2. Меню вибору необхідної ПЛІС

### 2.3. Робота в графічному редакторі

Графічний редактор призначений для створення файлу, який буде містити принципову схему пристрою, який розробляється. Після створення проекту необхідно виконати команду *File > New* (Файл > Новий) і у діалоговому вікні *New* (рис.3) на вкладці *Design Files* (Проектування файлу) вибрати тип файлу *Block Diagram/Schematic File* (Блок діаграм/схематичний файл) і натиснути *OK*.

В результаті відкриється вікно графічного редактора з файлом *Block1.bdf*, в якому буде створюватися схема (рис.4).

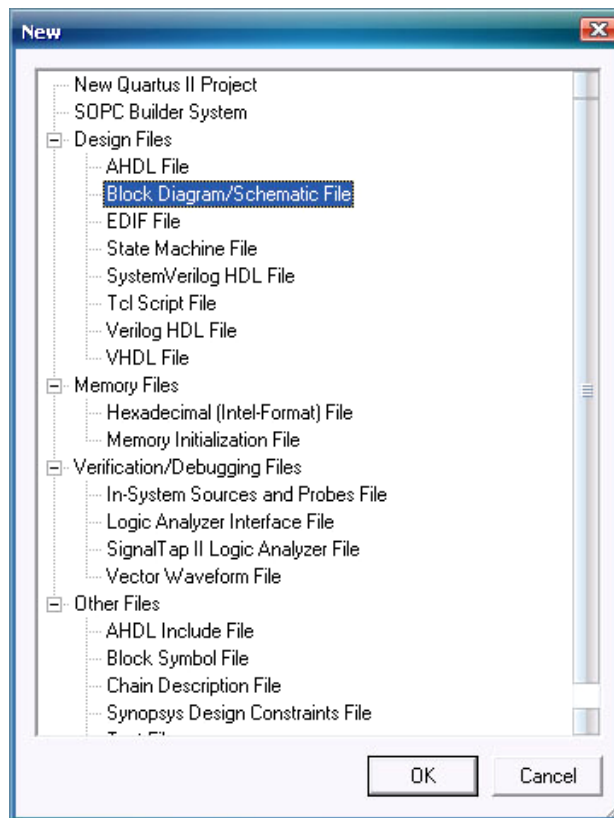


Рисунок 3. Створення схемного файлу проекту

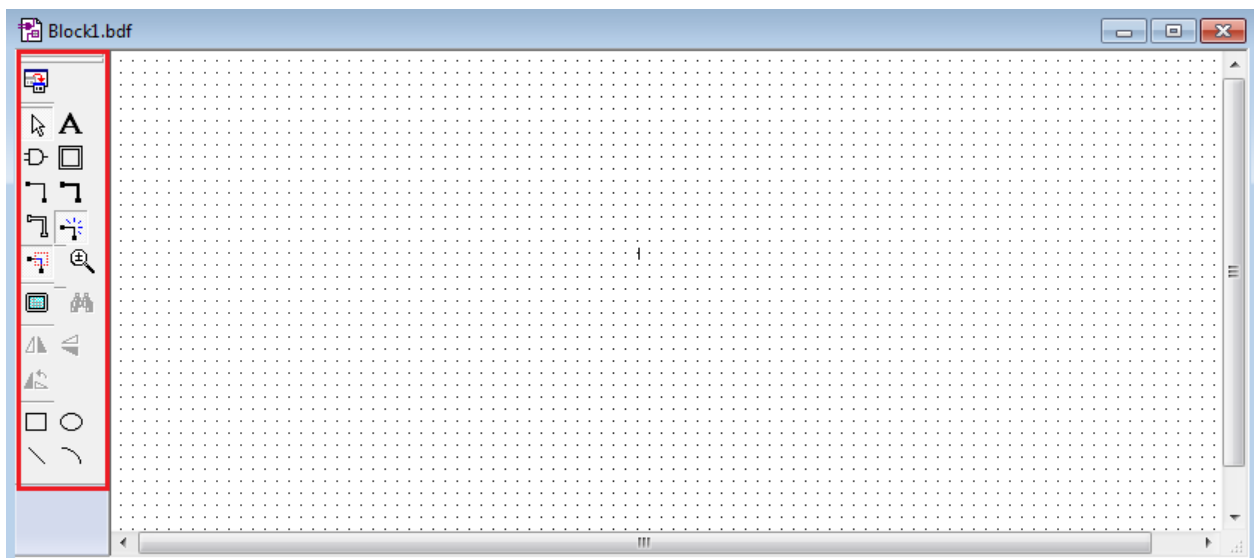


Рисунок 4. Вікно графічного редактора

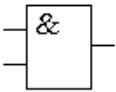
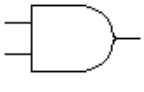
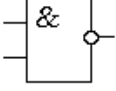
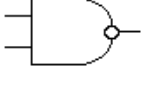
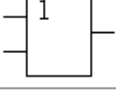

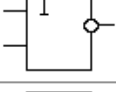

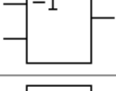

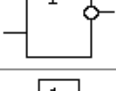


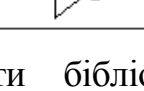
### 2.3.1. Створення простої комбінаційної схеми

Після створення файлу проекту стає активною панель інструментів, розташована ліворуч від робочої області вікна. Для введення елемента схеми слід клацнути по *Symbol Tools* (Символьні інструменти) (рис.4., червона область). В результаті відкриється вікно з бібліотеками елементів (рис.5).

Слід звернути увагу, що графічні зображення елементів схеми в системі *Quartus II* відрізняються від позначень, прийнятих стандартами в Україні. У таблиці 1 наведені зображення елементів схеми за вимогами ГОСТ і відповідні їм позначення елементів у системі *Quartus II*.

Таблиця 1

Відповідність загальних позначень елементів за вимогами ГОСТ і позначень в Quartus II.

Логічний елемент	Загальне зображення	Логічний елемент	Зображення в Quartus
"И"		and	
"И-НЕ"		nand	
"ИЛИ"		or	
«ИЛИ-НЕ»		nor	
«ИСКЛЮЧАЮЩЕЕ ИЛИ»		xor	
«ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ»		xnor	
«НЕ»		not	

Для введення логічного елемента «2І» слід обрати бібліотеку *primitives/logic* (примітиви/логіка). Розмістити необхідні компоненти на полі графічного редактора. Після цього слід розмістити вхідні (вхід) і вихідні (вихід) контакти, які знаходяться в папці *primitives/pin* (примітиви/контакт).

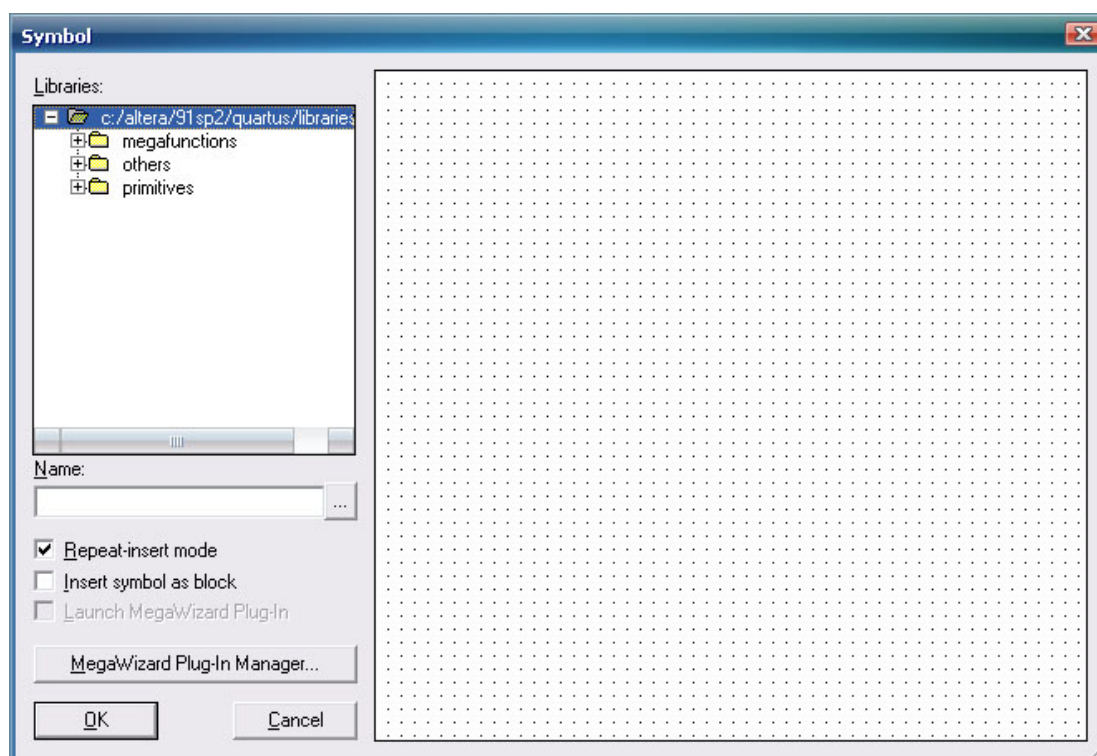


Рисунок 5. Вікно вибору бібліотеки елемента

Після цього провести з'єднання компонентів наступним чином: перемістити курсор в одну з двох точок схеми, які потрібно з'єднати, натиснути ліву кнопку миші і, не відпускаючи її, переміщати курсор до другої точки, що з'єднується. Далі слід перейменувати вхідні і вихідні контакти.

Приклад створеної простої комбінаційної схеми приведено на рис.6.

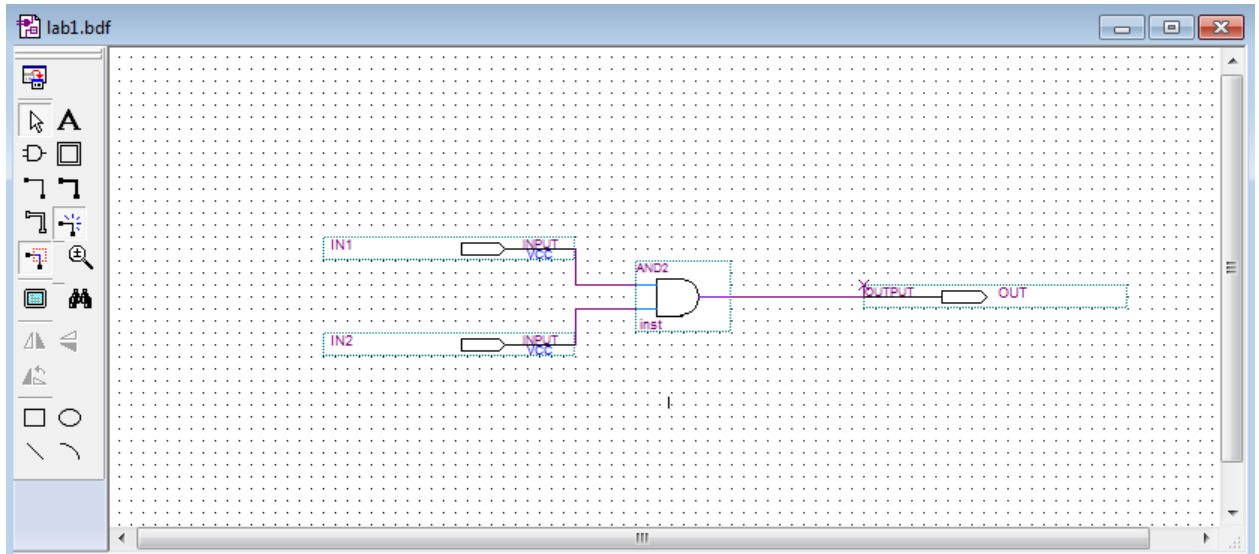



Рисунок 6. Вигляд створеної комбінаційної схеми

Для подальшої роботи зі схемою, вона має бути перевірена на наявність помилок.

**Увага!!! Для уникнення накопичення помилок виконувати перевірку логічних схем на кожному етапі проектування.**

Отже для перевірки створеної схеми на наявність помилок і визначення кількості логічних чарунок для її реалізації натиснути на іконку  – *Start Compilation* (Початок компіляції) в верхній панелі елементів. За виконанням процесу можна спостерігати в лівому вікні станів задач (рис.7). Успішно виконана перевірка завершується сповіщенням (рис.8).

Task	Time
2% Compile Design	00:00:11
9% Analysis & Synthesis	00:00:11
0% Fitter (Place & Route)	
0% Assembler (Generate programming files)	
0% Classic Timing Analysis	
EDA Netlist Writer	
Program Device (Open Programmer)	

Рисунок 7.

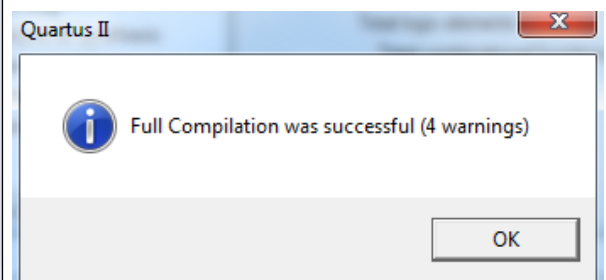


Рисунок 8.

Далі програма згенерує звіт перевірки, в якому відобразить дані про кількість зайнятих та вільних комірок у відсотках (рис.9).

Flow Summary	
Flow Status	Successful - Tue Feb 24 18:56:23 2015
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Web Edition
Revision Name	lab1
Top-level Entity Name	lab1
Family	Cyclone II
Met timing requirements	Yes
Total logic elements	1 / 4,608 (< 1 %)
Total combinational functions	1 / 4,608 (< 1 %)
Dedicated logic registers	0 / 4,608 (0 %)
Total registers	0
Total pins	3 / 158 (2 %)
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

Рисунок 9. Звіт перевірки схемного файлу

### 2.3.2. Підготовка процесу симуляції

Щоб перевірити правильність раніше створеної схеми, потрібно симулювати її роботу. Середовище *Quartus* підтримує два типи симуляції: часову і функціональну.

**Функціональна симуляція** дозволяє перевірити саме логіку роботи. З її допомогою ми можемо переконатися, що проект може і повинен працювати так як задумано. Перш за все потрібно робити саме функціональну симуляцію.

**Часова симуляція** дозволяє побачити сигнали з урахуванням всіх виникаючих затримок сигналу між елементами, входами і виходами. Тимчасову симуляцію роблять в останню чергу, вже після функціональної симуляції, щоб переконатися, що схема може працювати на потрібній заданій частоті.

Для невеликих проектів цілком достатньо робити тільки функціональну симуляцію.

Перш ніж робити симуляцію потрібно провести компіляцію проекту, для чого обрати команду *Processing > Start Compilation* (Обробка > Пуск компіляції). По завершенню компіляції потрібно створити спеціальний файл – *netlist* (Список) для функціональної симуляції. Вибираємо пункт меню *Processing/Generate Functional Simulation Netlist* (Обробка > Генерація файлу функціонального моделювання) (рис.10).

Для створення самого файлу симуляції, у якому описуються вхідні сигнали і задається їх список для перегляду, обрати пункт меню *File > New* (Файл > Новий) і потім в діалозі *Verification / Debugging Files* (Перевірка / Інструментальний файл) > *Vector Waveform File* (Файл векторної форми представлення) (рис.11).



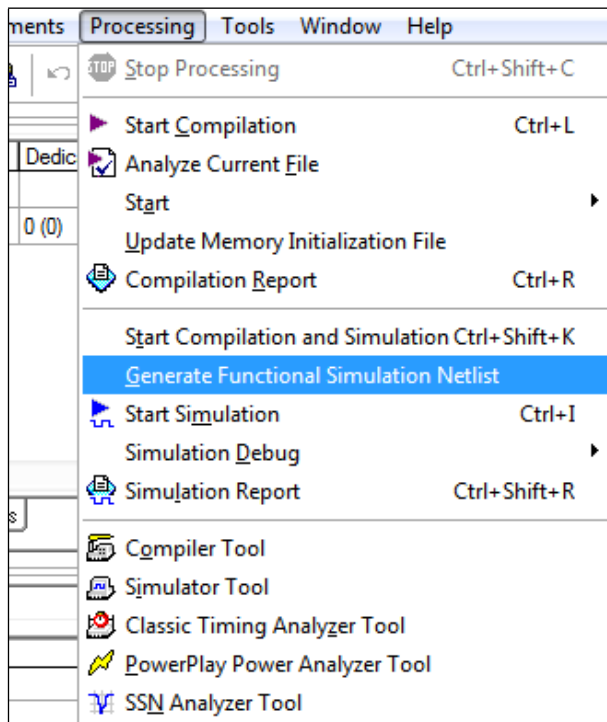


Рисунок 10.

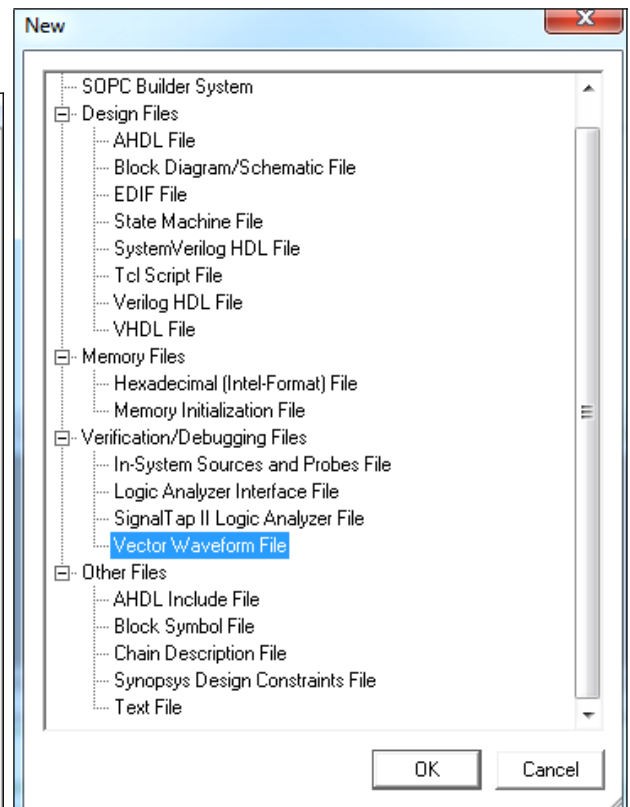


Рисунок 11.

Зберегти файл симуляції під ім'ям, наприклад, **lab1.vwf**. Тепер необхідно вибрати сигнали, для перегляду. Клацнути правою клавiшею миші на лівій панелі *Name* (Ім'я) і в випадаючому меню обрати пункт *Insert / Insert Node or Bus* (Установка/Установити вузол або шину) (рис.12).

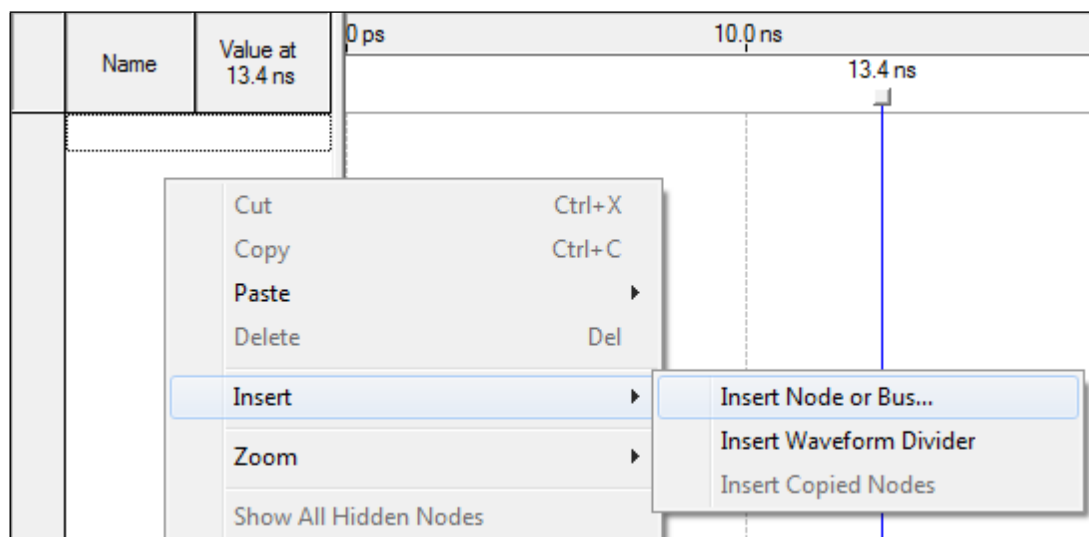


Рисунок 12. Вибір сигналів комбінаційної схеми

З'явилося діалогове вікно. У ньому можна набрати ім'я виходу сигналу або елемента.

*Повне ім'я сигналу може бути дуже довгим і не завжди зрозумілим. Тому краще скористатися пошуком сигналу в проекті.*

Натиснути кнопку *Node Finder* (Пошук сигналу), з'являється діалогове вікно для пошуку сигналів у проекті (рис.13).



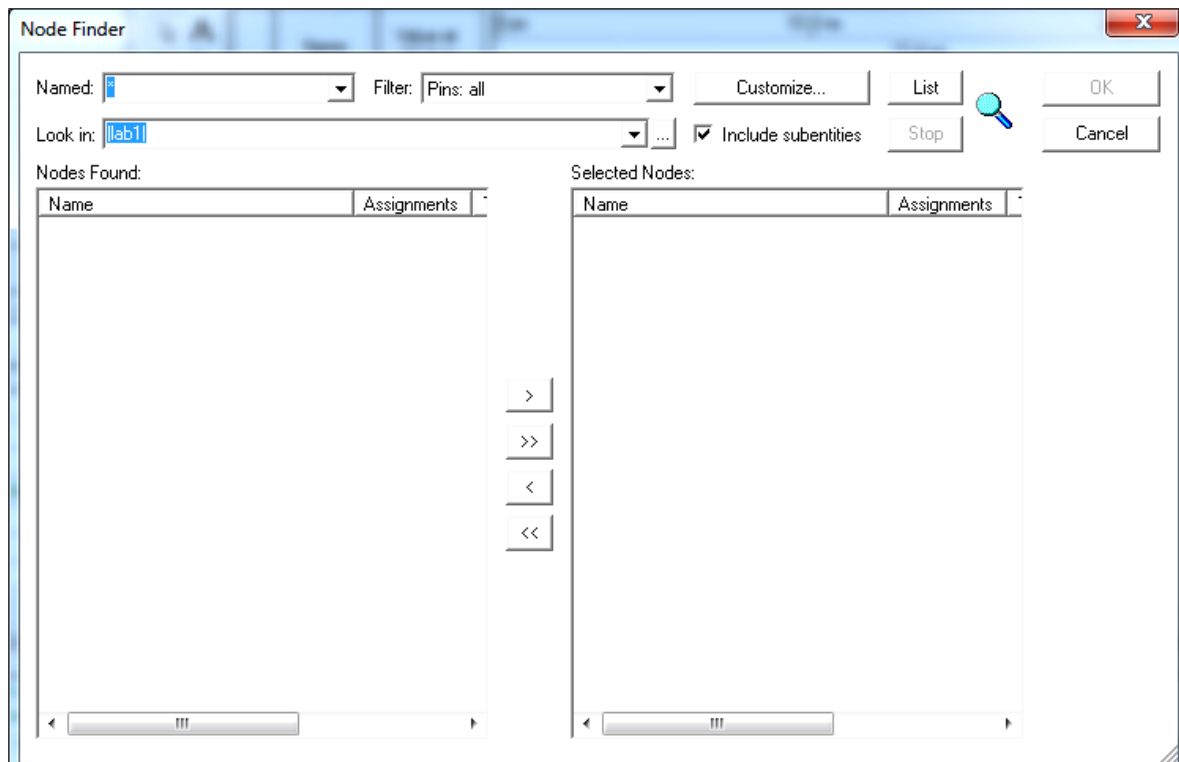


Рисунок 13. Пошук потрібних сигналів

У полі *Look in* (Дивитись в) вказано ім'я поточного проекту – пошук сигналу буде проводитись в ньому. Тут нічого змінювати не потрібно. У полі *Named* (Найменування) можна задати частину імені сигналу або проводу.

З натисканням кнопки *List* (Список) з'явиться список входів і виходів (рис.14).

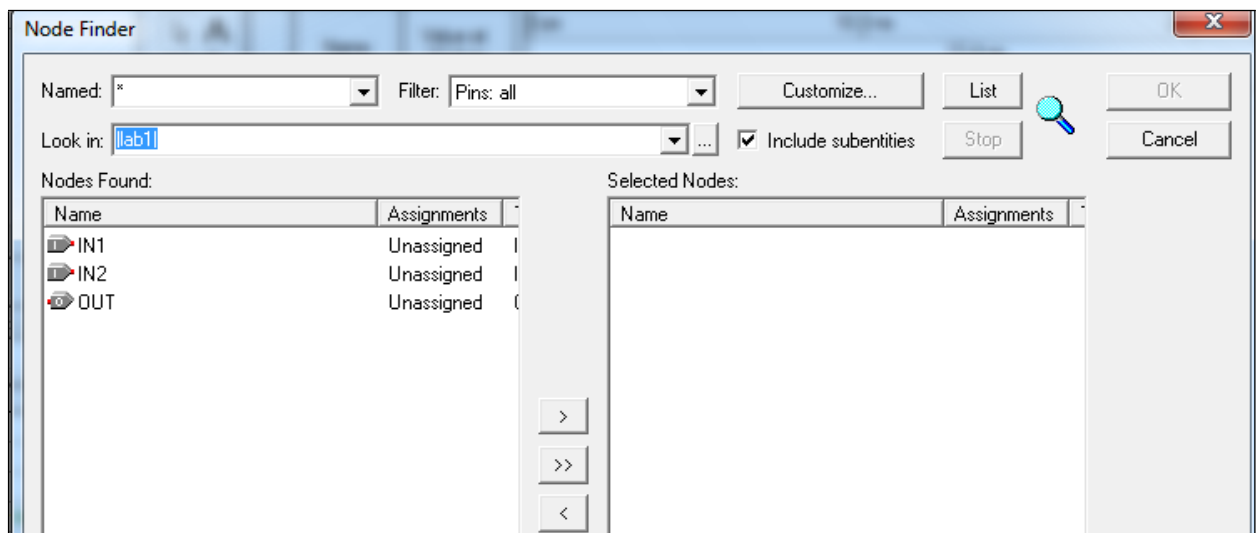


Рисунок 14. Вивід списку входів і виходів комбінаційної схеми

Вибрати всі потрібні сигнали зліва в таблиці знайдених сигналів *Nodes Found* (Наявні сигнали) і перенести їх вправо в таблицю обраних сигналів *Selected Nodes* (Обрані сигнали). Натиснути кнопку *OK*.

В файл симуляції додалися необхідні сигнали (рис.15).

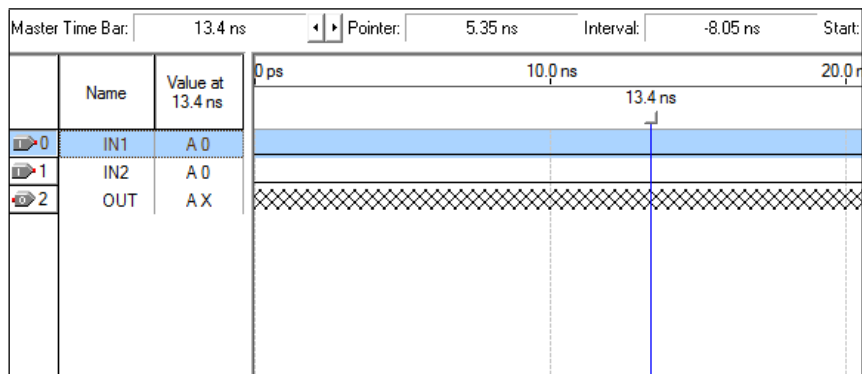


Рисунок 15. Зображення сигналів комбінаційної схеми

Перш ніж редагувати самі сигнали потрібно задати часову сітку (для зручності відтворення). Для цього зайти в пункт меню *Edit > Grid Size* (Правка > Розмір сітки). Крім того потрібно задати тривалість симуляції, для цього відкрити меню *Edit > End Time* (Правка > Час закінчення) (рис.16). Час симуляції визначити – 1сек.

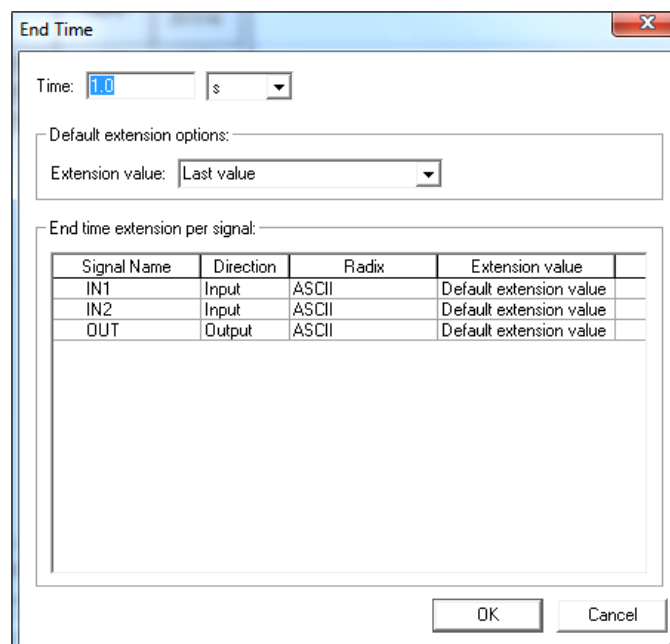


Рисунок 16. Встановлення часу симуляції

Редагувати вхідні сигнали проекту можна за допомогою кнопок панелі інструментів. Вибрати сигнали *IN1* та *IN2* і натиснути на панелі інструментів кнопку *Overwrite Clock* (Встановлення годинника). Зараз у цьому діалоговому вікні можна задати поведінку сигналу як тактової частоти. Задати період тактової частоти 10нс (рис.17).

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), функціональна тому, що зараз важливо перевірити саме логіку роботи, а не часові затримки всередині мікросхеми (рис. 18). Задати ім'я вхідного файлу симуляції *flipflop.vwf* натиснути *OK*.

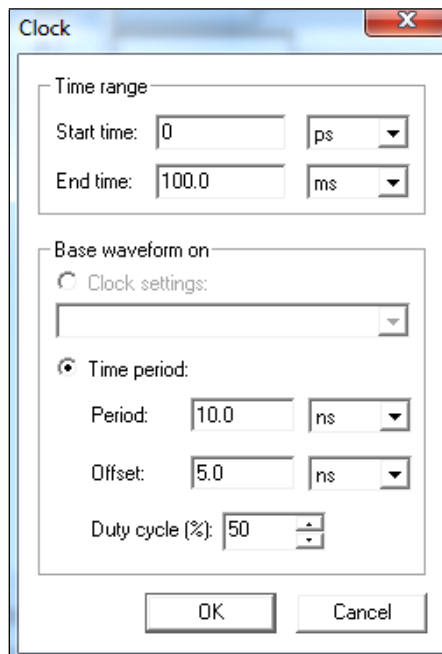


Рисунок 17. Встановлення періоду сигналу

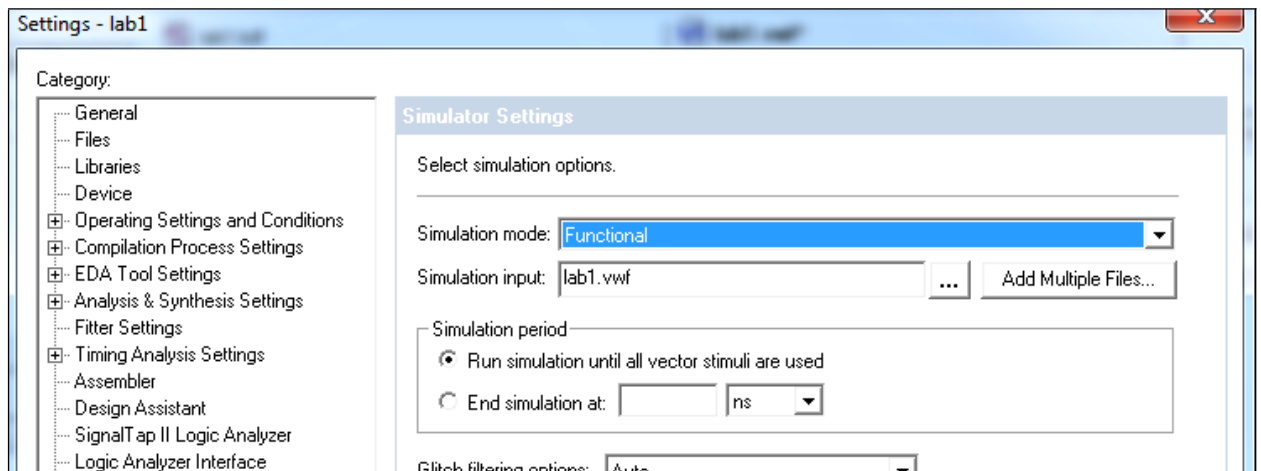


Рисунок 18. Встановлення типу симуляції

### 2.3.3. Проведення симуляції

Запустити симулятор з пункту меню *Processing > Start Simulation* (Обробка > Початок симуляції). Після успішної симуляції можна розглянути сигнали, які цікавлять (рис.19).

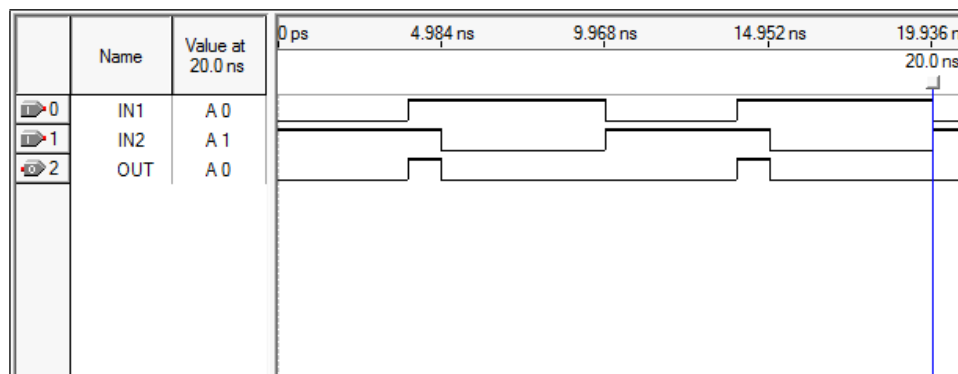


Рисунок 19. Результат функціонального моделювання

## 2.4. Виведення результату розробки на друк

Виведення результатів розробки на друк здійснюється за допомогою загальних можливостей *Quartus II*.

Для відображення створених файлів треба вибирати за необхідністю з меню *File > Print Preview* (Файл Предперегляд), *Page > Setup* (Налаштування сторінки) або *Print* (Друк) потрібні команди та використовуючи їх можливості виводити інформацію на друк, або у файл друку.

## 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.

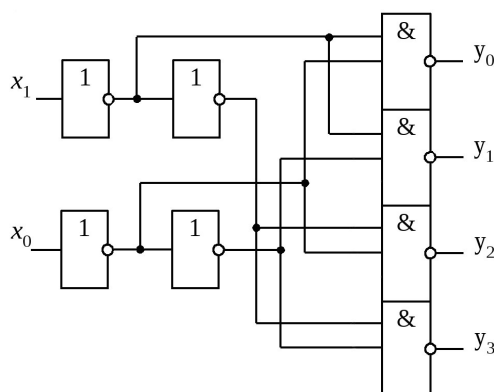
Для захисту звіту має бути представлено іменний файл проекту розробки у програмі *Quartus II*.

## 2.6. Завдання до лабораторної роботи

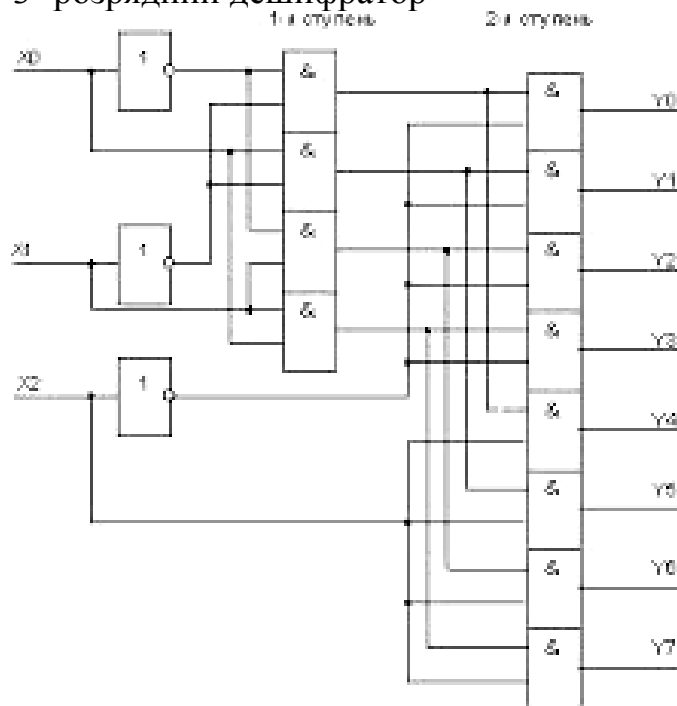
Розробити комбінаційну схему за варіантом та провести часову симуляцію розробленої схеми (відтворити таблицю істинності).

Варіанти завдань наведені нижче. Варіант завдання обирається за останньою цифрою залікової книжки.

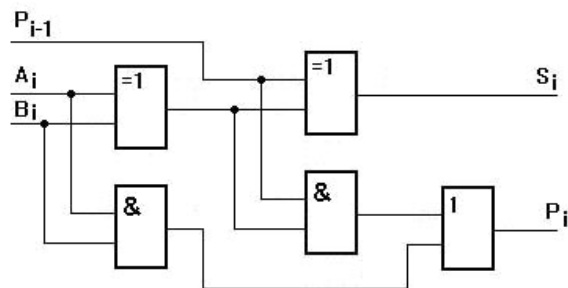
### 1. 2-розрядний дешифратор



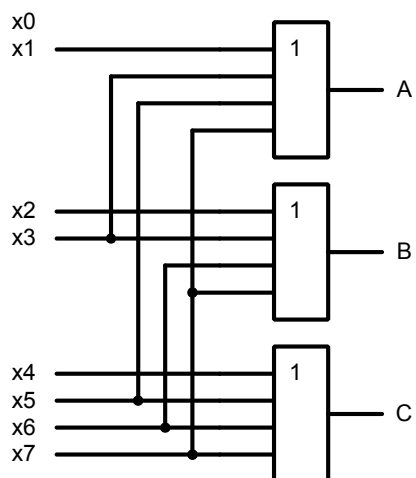
## 2. 3-розрядний дешифратор



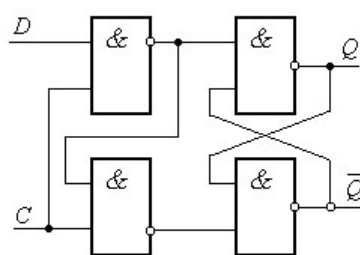
## 3. 4-розрядний двійковий суматор



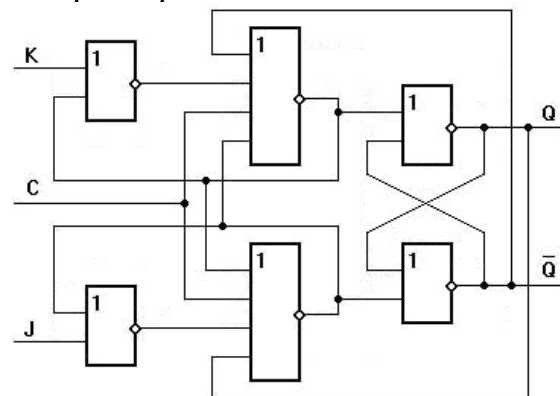
## 4. 8-входовий двійковий шифратор



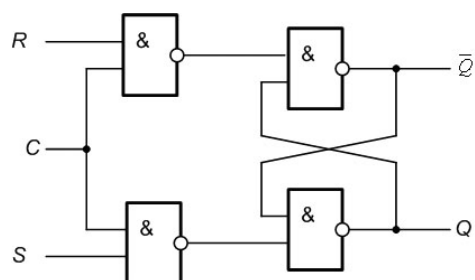
## 5. D-триггер



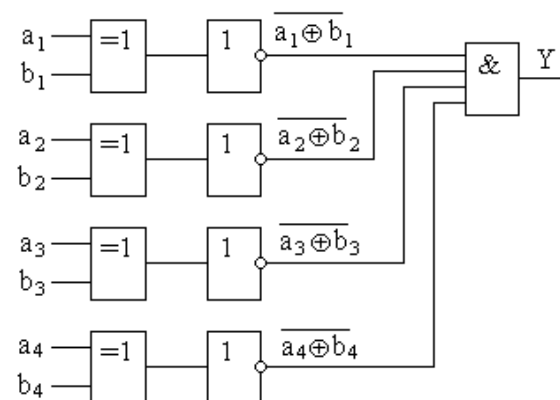
## 6. JK-триггер



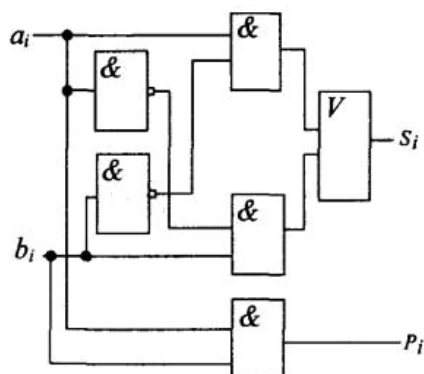
## 7. RS-триггер



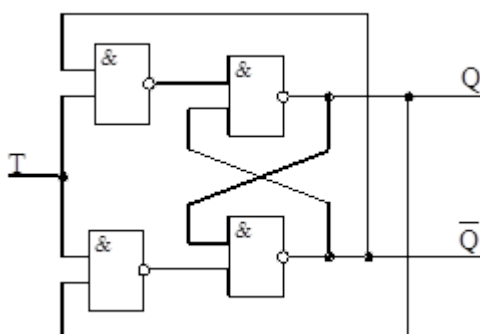
## 8. Компаратор



## 9. Однорозрядний суматор



## 10. Т-триггер



### Питання для самоперевірки

1. Розкажіть що таке ПЛІС.
2. Розкажіть про переваги ПЛІС.
3. Що таке файл конфігурації ПЛІС?
4. Яким чином проводиться конфігурування ПЛІС?
5. Призначення системи автоматизованого проектування *Quartus II*.
6. Етапи створення проекту в системі *Quartus II*.
7. Робота з графічним редактором *Quartus II*.

### Список літератури

1. Обзор элементной базы фирмы ALTERA. Антонов А.П., Мелехин В.Ф., Филлипов А.С. СПб.: ИД Fine Street, 1997;
2. Системы автоматизированного проектирования фирмы Altera MAX+Plus II и *Quartus II*. Краткое описание и самоучитель. Комолов Д.А., Мяльк Р.А., Зобенко А.А., Филлипов А.С. М.: ИП ИП Радио Софт 2002;
3. Архитектура и функционирование ПЛИС. Матюшин О.Т., под редакцией Баскакова А.И. М.: МЭИ 2003;
4. <http://ahdl.winglion.ru/ahdl.htm>. Електронний довідник AHDL.