



Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»



Кафедра автоматизації
електромеханічних
систем та електроприводу
ФЕА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

СИСТЕМИ АВТОМАТИЗАЦІЇ ЛАБОРАТОРНИЙ ПРАКТИКУМ. ЧАСТИНА 1

Навчальний посібник

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра
за освітньою програмою «Електромеханічні системи автоматизації, електропривод та
електромобільність»
спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»

Укладачі: С.О. Бур'ян, Г.Ю. Землянухіна, Р.С. Волянський

Електронне мережне навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2022

Рецензент *Будько В.І., д-р. техн. наук, доц., зав. каф. відновлювальних джерел енергії КПІ ім. Ігоря Сікорського*

Відповідальний редактор *Красношапка Н.Д., канд. техн. наук, доц.*

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол № 6 від 24.06.2022 р.)
за поданням Вченої ради факультету електроенерготехніки та автоматики
(протокол № 10 від 20.06.2022 р.)*

Навчальний посібник включає методичні матеріали щодо виконання та оформлення лабораторних робіт з дисципліни “Системи автоматизації-1” для освітньої програми “Електромеханічні системи автоматизації, електропривод та електромобільність” всіх форми навчання спеціальності 141 “Електроенергетика, електротехніка та електромеханіка”. Розглядаються питання виконання та оформлення лабораторних робіт, наведені приклади синтезу, програми та шляхи технічної реалізації систем автоматизації за заданими умовами роботи.

Реєстр. № НП 21/22-709. Обсяг 5,7 авт. арк.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Перемоги, 37, м. Київ, 03056
<https://kpi.ua>

Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів
і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

ЗМІСТ

Вступ.....	4
Лабораторна робота № 1.....	5
Лабораторна робота № 2.....	36
Лабораторна робота № 3.....	65
Лабораторна робота № 4.....	82
Лабораторна робота № 5.....	104
Лабораторна робота № 6.....	138
Лабораторна робота № 7.....	168
Лабораторна робота № 8.....	195
Лабораторна робота № 9.....	240
Заходи безпеки під час виконання лабораторних робіт в лабораторії «Автоматизації технологічних процесів, установок і комплексів»	252
Список рекомендованої літератури.....	254
Додаток.....	255

ВСТУП

Навчальний посібник містить опис лабораторних робіт, впровадження яких у освітній процес пов'язане з безперервним вдосконаленням елементної бази систем автоматизації, впровадженням нових технічних засобів, подальшим розвитком методів синтезу дискретних схем автоматики.

В навчальному посібнику, відповідно до силябусу освітнього компонента «Системи автоматизації-1», наведені методичні вказівки для виконання 9-ти лабораторних робіт. Особливістю даних робіт є те, що вони можуть проводитися як на реальному обладнанні лабораторії, так і в режимі симуляції у відповідному програмному забезпеченні.

Мета лабораторних робіт – закріпити теоретичні знання методів синтезу дискретних схем промислової автоматики та сучасної елементної бази пристроїв систем автоматизації, навчитися вирішувати практичні задачі автоматизації методами логічного синтезу, отримати навички реалізації та дослідження схем автоматики на сучасній елементній базі, що включає цифрові інтегральні мікросхеми та програмовані логічні інтегральні схеми FPGA.

У навчальному посібнику до кожної лабораторної роботи поданні основні теоретичні відомості, необхідні для підготовки, виконання та наступного захисту виконаної роботи. Крім того, посібник містить програму роботи, вказівки для її виконання, опис лабораторної установки, зміст звіту та контрольні запитання. Всі лабораторні роботи виконуються на сучасному обладнанні, що використовується у промислових системах автоматизації.

Лабораторні роботи 1, 2, 8 та 9 виконуються здобувачами індивідуально, а лабораторні роботи 3-7 виконуються класичним «бригадним» методом. Звіти з лабораторних робіт оформлюються згідно діючих стандартів до технічної документації.

Лабораторна робота № 1

ЛОГІЧНИЙ СИНТЕЗ ТА ДОСЛІДЖЕННЯ СХЕМ АВТОМАТИКИ НА ДИСКРЕТНИХ ЛОГІЧНИХ ЕЛЕМЕНТАХ

Тривалість лабораторної роботи – 4 год.

Тривалість домашньої роботи – 4 год.

Мета роботи – практично перевірити різні методи синтезу одноктактних та багатотактних схем, набути навичок переходу від алгебраїчних виразів до схем на інтегральних мікросхемах, засвоїти методи синтезу на циклограмах та таблиць переходів, навчитися складати схеми електричні принципові та інтегральних мікросхемах з використанням монтажних плат, кнопок, світлодіодів та резисторів.

1.1. Основні теоретичні відомості

Перед тим як починати синтез схеми слід визначити, одноктактною чи багатотактною є синтезована схема. Схема є одноктактною, якщо стан її виходів визначається тільки комбінацією значень вхідних сигналів і не залежить від послідовності їх надходження. У багатотактних схемах стан виходів залежить не тільки від набору вхідних сигналів у певний момент часу, але й від послідовності їх надходження або від внутрішнього стану схеми.

1.1.1 Синтез одноктактних схем

Робота одноктактної схеми повністю описується таблицею істинності. Тому синтез одноктактної схеми можна виконати у такій послідовності:

- 1) за заданими умовами роботи скласти таблицю істинності;
- 2) за таблицею істинності записати логічні формули для вихідних змінних у довершеній диз'юнктивній нормальній формі (ДДНФ) або у довершеній кон'юнктивній нормальній формі (ДКНФ);
- 3) якщо є можливість, записані формули мінімізувати.

Як приклад розглянемо синтез схеми, виходячи з таких умов її роботи: схема має три вхідні сигнали a, b, c і два вихідні X і Y ; вихідний сигнал $X = 1$, якщо непарна кількість вхідних сигналів дорівнює одиниці; вихідний сигнал $Y = 1$, якщо парна кількість вхідних сигналів дорівнює одиниці (нуль вважати парним числом).

За заданими умовами роботи складаємо таблицю істинності (табл. 1.1), з якої:

$$X = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc;$$

$$Y = \bar{a}\bar{b}\bar{c} + \bar{a}bc + a\bar{b}c + abc.$$

Таблиця 1.1 – Таблиця істинності

Вхідні сигнали			Вихідні сигнали	
a	b	c	X	Y
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

1.1.2 Синтез багатотактних схем на основі таблиць переходів і карт Карно

Розглянемо послідовність синтезу на такому прикладі. Необхідно виконати синтез схеми, виходячи з таких умов її роботи: схема має три вхідні сигнали a, b, c і один вихідний X ; вихідний сигнал $X = 1$ тільки в тому разі, якщо всі три вхідні сигнали дорівнюють одиниці, причому сигнал a подано другим, тобто після сигналу b або c .

Синтез виконується у кілька етапів:

1. *Складання первинної таблиці переходів.* Цей етап починається з визначення кількості вихідних станів. Умови роботи схеми не передбачають жодних обмежень у послідовності надходження вхідних сигналів, тому слід розглянути всі можливі комбінації вхідних сигналів і для кожної з них

визначити кількість можливих станів схеми. Для цього рекомендується скласти таблицю (табл. 1.2)

Таблиця 1.2 – Рекомендований вигляд таблиці істинності

Номер стану схеми	Комбінації вхідних сигналів			Значення вихідного сигналу	Коментар
	<i>a</i>	<i>b</i>	<i>c</i>		
1	0	0	0	0	-
2	0	0	1	0	-
3	0	1	0	0	-
4	0	1	1	0	-
5	1	0	0	0	-
6	1	0	1	0	<i>a</i> подано першим
7	1	0	1	0	<i>a</i> подано другим
8	1	1	0	0	<i>a</i> подано першим
9	1	1	0	0	<i>a</i> подано другим
10	1	1	1	0	<i>a</i> подано не другим
11	1	1	1	1	<i>a</i> подано другим

Кожній з перших п'яти комбінацій вхідних сигналів відповідає свій однозначний (1 – 5) стан схеми. Кожній з решти трьох комбінацій сигналів *a, b, c* (101,110,111) відповідають по два стани схеми, оскільки схема повинна відрізняти черговість надходження сигналу *a*. Отже, загальна кількість станів схеми – 11.

Після визначення кількості станів схеми будуємо таблицю переходів (рис. 1.1), яка має 11 рядків по одному для кожного стану схеми. У крайньому лівому стовпці записуємо номери вихідних станів, у крайньому правому – відповідні їм значення вихідної змінної, у стовпцях середньої частини таблиці записують номери тих станів, у які переходить схема з вихідного, якщо набір вхідних сигналів відповідає стовпцям.

Заповнюючи середню частину таблиці переходів, необхідно мати на увазі, що багатотактні схеми можуть перебувати в двох станах – стійкому і нестійкому. Стійкий стан характеризується тим, що в разі незмінного набору вхідних змінних він також не змінюється протягом скільки завгодно тривалого

часу. Номери стійких станів записують у кутових дужках $\langle \rangle$. Нестійкий стан спостерігається в тому разі, коли набір вхідних сигналів змінюється, а внутрішній стан схеми, тобто стан вихідних і проміжних змінних, ще не набув відповідності новому наборові вхідних сигналів. Тривалість нестійкого стану визначається часом запізнення у передаванні вхідної дії на вихід схеми. Номери нестійких станів записують без дужок.

Номер вихідного стану	Наступні стани								Вихідна змінна X
		a		b			c		
1	$\langle 1 \rangle$	5	–	3	–	–	–	2	0
2	1	–	–	–	4	–	7	$\langle 2 \rangle$	0
3	1	–	9	$\langle 3 \rangle$	4	–	–	–	0
4	–	–	–	3	$\langle 4 \rangle$	10	–	2	0
5	1	$\langle 5 \rangle$	8	–	–	–	6	–	0
6	–	5	–	–	–	10	$\langle 6 \rangle$	2	0
7	–	5	–	–	–	11	$\langle 7 \rangle$	2	0
8	–	5	$\langle 8 \rangle$	3	–	10	–	–	0
9	–	5	$\langle 9 \rangle$	3	–	11	–	–	0
10	–	–	8	–	4	$\langle 10 \rangle$	6	–	0
11	–	–	9	–	4	$\langle 11 \rangle$	7	–	1

Рисунок 1.1 – Таблиця переходів

Під час заповнення середньої частини таблиці переходів у першу чергу записуємо номери стійких станів на перетині рядків з тим самим номером вихідного стану і стовпців, яким відповідає набір вхідних змінних для цього стану схеми. Номери вихідних станів і відповідні їм набори вхідних змінних наведено в табл. 1.2. Стійкий стан $\langle 1 \rangle$ записуємо у першому рядку і стовпці, для якого $a = 0, b = 0, c = 0$; стійкий стан $\langle 2 \rangle$ – у другому рядку і стовпці, для якого $a = 0, b = 0, c = 1$ тощо.

Після запису стійких станів визначаємо всі можливі переходи з кожного стійкого стану в інші і записуємо номери нестійких станів. З кожного стійкого стану можливі тільки три переходи в разі змінювання кожної вхідної змінної окремо. Наприклад, у стані $\langle 1 \rangle$ вхідні змінні $a = 0, b = 0, c = 0$ (набір 000), тому слід розглядати переходи схеми зі змінюванням кожної змінної a, b, c з нуля на одиницю.

Якщо a змінюється з 0 на 1 (набір 100), схема переходить у стан 5 (табл. 1.2), якщо b змінюється з 0 на 1 (набір 010), – у стан 3, а якщо c змінюється з 0 на 1 (набір 001), – у стан 2. Ці номери нестійких станів записуємо у першому рядку таблиці переходів. Решту клітинок у цьому рядку заповнюємо ризками.

Запишемо тепер переходи зі стану $\langle 2 \rangle$ (набір вхідних змінних 001). Можливі змінювання вхідних сигналів: a з 0 на 1, b з 0 на 1, c з 1 на 0. Коли a змінюється з 0 на 1 (набір 101), схема переходить у стан 7, а не 6, оскільки у стані $\langle 2 \rangle$ сигнал c вже дорівнював одиниці і, отже, сигнал a подається другим. Якщо сигнал b змінюється з 0 на 1 (набір 011), схема переходить у стан 4, а якщо c змінюється з 1 на 0 (набір 000) – у стан 1.

Аналогічно заповнюємо решту рядків таблиці переходів.

2. *Стиснення таблиці переходів.* Таблиця переходів, складена безпосередньо за умовами роботи схеми, називається первинною. Стисненням первинної таблиці переходів називають скорочення кількості її рядків шляхом сполучення кількох рядків в один і поєднання еквівалентних станів.

Рядки об'єднують за такими правилами:

- 1) кілька рядків можна об'єднати в один, якщо номери станів у них у тих самих стовпцях однакові або замість номеру стану стоїть ризка;
- 2) якщо об'єднують стійкі і нестійкі стани, то в об'єднаний рядок записують стійкі стани;
- 3) можна об'єднувати рядки як з однаковими значеннями вихідних змінних, так і з різними;
- 4) у стисненій таблиці переходів значення вихідних змінних не записують.

У розглядуваному прикладі кількість рядків таблиці переходів можна зменшити до двох, об'єднавши рядки 1, 2, 3, 7, 9, 11 і 4, 5, 6, 8, 10. Тоді стиснена таблиця переходів матиме вигляд, показаний на рис 1.2

3. *Визначення кількості та розміщення станів проміжних змінних.* Кількість проміжних змінних визначають виходячи з того, що кожному рядку

стисненої таблиці переходів має відповідати своя певна комбінація значень проміжних змінних. Стиснена таблиця переходів на рис 1.2 має два рядки, тому достатньо ввести тільки одну проміжну змінну P . Значення проміжної змінної для першого і другого рядків таблиці можна обрати довільне. Нехай у станах 1, 2, 3, 7, 9, 11 (перший рядок) змінна $P = 1$, а в станах 4, 5, 6, 8, 10 (другий рядок) $P = 0$.

Номери вихідних станів	Наступні стани							
	a		b			c		
1, 2, 3, 7, 9, 11	< 1 >	5	< 9 >	< 3 >	4	< 11 >	< 7 >	< 2 >
4, 5, 6, 8, 10	1	< 5 >	< 8 >	3	< 4 >	< 10 >	< 6 >	2

Рисунок 1.2 – Стиснена таблиця переходів

4. Складання карт Карно для проміжних і вихідних змінних та визначення за ними алгебричних виразів. Карти Карно складають так, щоб вхідні змінні розміщувалися зверху карти так само, як і у середній частині таблиці переходів, а проміжні змінні – ліворуч. Карту Карно для проміжної змінної P показано на рис. 1.3, а. Заповнюють її так. Спочатку визначають рядки карти, що відповідають рядкам стисненої таблиці переходів. Верхній рядок карти ($P = 0$) відповідає другому рядку таблиці, що позначено символом $S_{4,5,6,8,10}$ проти верхнього рядка. Нижній рядок карти ($P = 1$) відповідає першому рядку таблиці. Його позначено символом $S_{1,2,3,7,9,11}$. Після цього замість номеру стану в таблиці переходів у клітинках карти Карно записують значення проміжної змінної, яке вона має у цьому стані, тобто замість станів 1, 2, 3, 7, 9, 11 записують одиницю, а замість станів 4, 5, 6, 8, 10 – нуль.

Карту Карно для вихідної змінної заповнюють аналогічно. Відмінність полягає лише в тому, що враховують тільки стійкі стани в таблиці переходів, а номери нестійких станів замінюють рисками. Карту Карно для вихідної змінної

показано на рис. 1.3, б. Тут стійкий стан <11> замінено на одиницю, решту стійких станів – на нулі, нестійкі стани – на rischi.

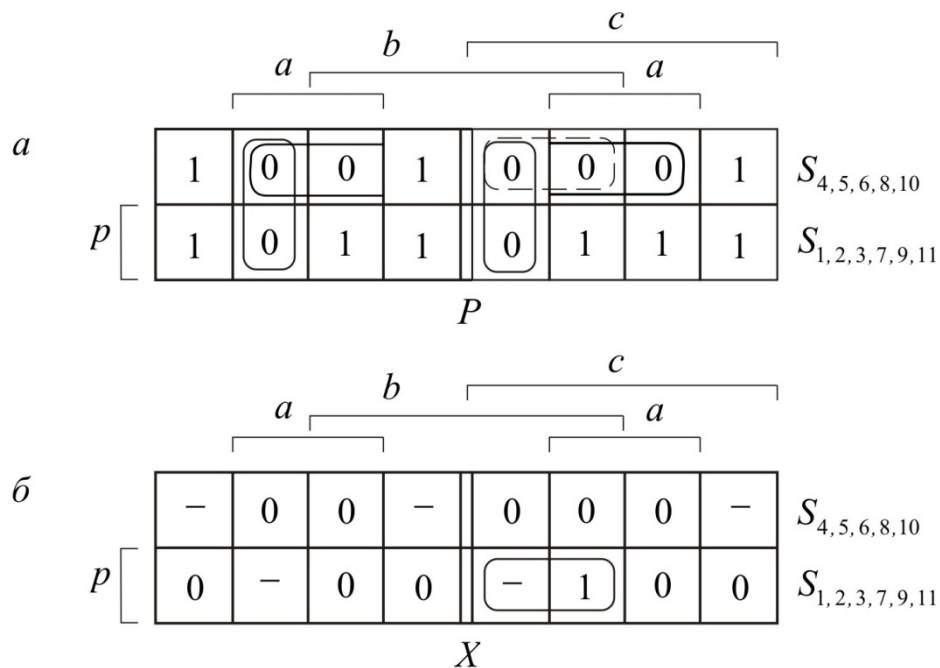


Рисунок 1.3 – Карти Карно: а – для проміжної змінної P ; б – для вихідної змінної X

За картами Карно рис. 1.3 складають алгебричні вирази для проміжної змінної P і вихідної X :

$$P = (\bar{a} + p)(\bar{a} + b + c)(a + \bar{b} + \bar{c})(\bar{b} + \bar{c} + p);$$

$$X = pbc.$$

У виразі для P введено додатковий член $(\bar{b} + \bar{c} + p)$, зайвий за умовами мінімізації. Цей член не залежить від a і зберігає значення $P = 0$ у разі, коли сигнал a надходить останнім, причому a стає рівним 1 раніше, ніж \bar{a} набуває значення 0.

Для складання виразу для P у контури об'єднували клітинки з нулями і функцію отримано у кон'юнктивній нормальній формі, тому що схему необхідно реалізувати на елементах АБО-НІ. Функцію $X = pbc$ доцільно реалізувати на діодній приставці І.

1.1.3 Мінімізація логічних функцій за допомогою карт Карно

Карта Карно – один з графічних способів подання логічних функцій. Для функцій n змінних вона складається з 2^n клітинок, причому кожна клітинка

відповідає певному набору змінних. Вигляд карт Карно для функцій 2,3 і 4-ти змінних зображений на рисунку 1.1. Вхідні змінні розміщуються з зовнішніх сторін карти проти її рядків або стовпців.

Значення вхідної змінної стосується усіх клітинок у рядку або стовпці і дорівнює 1, якщо проти рядка або стовпця є дужка з позначенням цієї змінної. Для решти рядків і стовпців значення змінної дорівнює 0. У клітинках карти записується те значення функції, яке вона має при наборах вхідних змінних, що відповідають цим клітинкам.

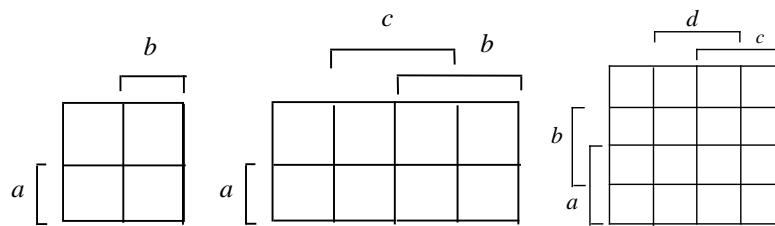


Рисунок 1.4 – Форма запису карт Карно

Функцію, задану алгебраїчним виразом, можна подати у вигляді карти Карно. Для цього передусім задану функцію треба подати в диз'юнктивній (ДНФ) або кон'юнктивній (КНФ) нормальній формі. При використанні карт Карно для мінімізації логічних функцій необхідно побудувати карту для відповідної кількості змінних і нанести на неї задану функцію. Потім слід об'єднати сусідні клітинки з одиницями в контури, записати вирази для контурів і скласти їх диз'юнкцію.

1.1.4 Синтез схем на основі циклограм

Циклограма – це графічне зображення послідовності роботи окремих елементів схеми у часі [1]. Робота елемента і відповідна йому змінна (сигнал) зображується на циклограмі відрізком горизонтальної прямої. Позначення змінної або елемента ставиться ліворуч від відрізка. Товстою лінією позначаються сигнали командних і виконавчих елементів, тонкою – додаткових проміжних, пунктиром – умовне вмикання елементів. Умовне вмикання елемента відповідає пасивному стану, при якому він не впливає на стан вихідних елементів схеми незалежно від того, ввімкнтий цей елемент або ні.

Черговість вмикання елементів визначається положенням лівих кінців відрізків, черговість вимикання – правих. Дія одного елемента на інший зображується на циклограмі стрілкою, що зазначає напрям дії. Наприклад, циклограма на рисунку 1.5 відповідає такій послідовності роботи елементів: спочатку надходить сигнал a , який діє на вихідний X і проміжний P елементи і вмикає їх (зникнення сигналу a не впливає на вихідний і проміжний елементи); потім надходить сигнал b , який діє на елемент P і вмикає його; після зникнення сигналу b вмикається елемент X і схема повертається у вихідний стан [2].

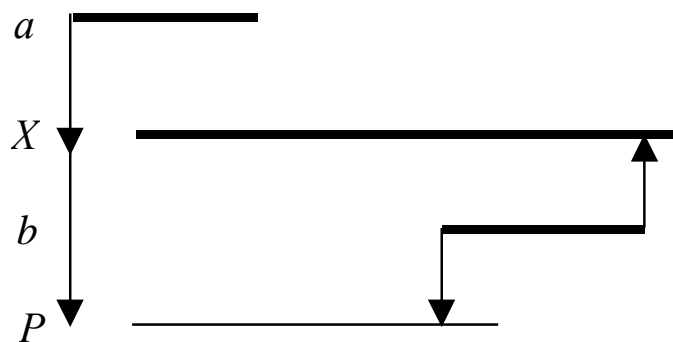


Рисунок 1.5 – Циклограма

1.2 Програма роботи

Перед початком лабораторної роботи група ділиться на бригади відповідно до кількості робочих місць. Кожному студенту бригади необхідно вибрати один із варіантів (1-6), який містить 4 завдання, з таблиці 1.3. Для кожного з чотирьох завдань необхідно:

- 1) виконати логічний синтез схеми керування за заданими умовами роботи (**виконується вдома під час СРС**);
- 2) мінімізувати кількість логічних операцій у отриманих алгебричних виразах (**виконується вдома під час СРС**);
- 3) за мінімізованими у п.2 алгебраїчними виразами скласти принципову схему на дискретних логічних елементах АБО, І, АБО-НІ, І-НІ та НІ та зробити перелік елементів до неї (**виконується вдома під час СРС**);

- 4) зібрати одну схему за вибором викладача на лабораторному стенді та перевірити її працездатність*;
- 5) показати зібрану схему викладачу та продемонструвати її роботу (**викладач підтверджує своїм підписом у протоколі працездатність схеми**)*;
- 6) зібрати схему у програмному середовищі Multisim та виконати симуляцію її роботи (**виконується безпосередньо під час лабораторних робіт**);
- 7) оформити звіт по лабораторній роботі та зробити висновки.

* - виконується тільки під час очного навчання

Таблиця 1.3 – Варіанти завдань до лабораторної роботи №1

Бригада 1	Номери завдань
1	1; 20; 40; 57;
2	2; 23; 39; 58;
3	3; 30; 38; 59;
4	4; 26; 37; 60;
5	5; 18; 35; 50;
6	6; 25; 35; 61;
Бригада 2	Номери завдань
1	5; 17; 36; 61;
2	6; 21; 35; 62;
3	7; 29; 34; 63;
4	8; 25; 33; 64;
5	9; 19; 38; 52;
6	10; 20; 39; 53;
Бригада 3	Номери завдань
1	9; 19; 41; 49;
2	10; 31; 42; 50;
3	11; 22; 43; 51;
4	12; 27; 44; 52;
5	13; 18; 45; 53;
6	14; 21; 46; 54;
Бригада 4	Номери завдань
1	13; 32; 45; 53;
2	14; 24; 46; 54;
3	15; 18; 47; 55;
4	16; 28; 48; 56;
5	1; 17; 34; 50;
6	2; 18; 44; 51;
Бригада 5	Номери завдань
1	4; 19; 40; 50;
2	5; 20; 41; 51;
3	6; 21; 42; 52;
4	7; 22; 43; 43;
5	8; 23; 44; 54;
6	9; 24; 45; 55;
Бригада 6	Номери завдань
1	12; 25; 46; 56;
2	13; 26; 47; 57;
3	14; 27; 48; 58;
4	15; 28; 39; 59;
5	16; 29; 38; 60;
6	11; 30; 37; 61;

Умови роботи схем

Скласти схему по заданому логічному виразу

$$1. f = c\bar{d}\bar{e} + bcd + a\bar{c}\bar{e}$$

$$2. f = \bar{b}\bar{c}\bar{d}\bar{e} + abcd + \bar{a}\bar{b}de$$

$$3. f = (a + d + e)(\bar{a} + d + \bar{e})(\bar{b} + c + \bar{e})$$

$$4. f = \bar{b}ce + ad\bar{e} + \bar{d}\bar{e}(\bar{a} + b)$$

$$5. f = \bar{c}\bar{d}\bar{e} + \bar{b}cd + a\bar{c}\bar{e}$$

$$6. f = (\bar{a} + c + d + e)(a + \bar{c} + d + \bar{e})(a + c + \bar{d} + \bar{e})$$

$$7. f = b + \bar{b}cd + a\bar{b}e(\bar{c} + d)$$

$$8. f = \bar{b}\bar{c}\bar{d}\bar{e} + \overline{abcd} + \overline{acde}$$

$$9. f = \bar{c}\bar{d}\bar{e} + \bar{b}\bar{c}\bar{d} + a\bar{c}\bar{e}$$

$$10. f = \bar{d}\bar{e}(a + \bar{b}) + \bar{c}\bar{d}\bar{e} + abcd$$

$$11. f = \bar{a}\bar{c}\bar{d}\bar{e} + acd\bar{e} + \bar{a}\bar{b}\bar{c}\bar{e}$$

$$12. f = c\bar{d}\bar{e} + \bar{b}\bar{c}\bar{d} + \bar{a}\bar{d}\bar{e}$$

$$13. f = a\bar{c}\bar{d}\bar{e} + \bar{a}bcd + abcd + \bar{a}\bar{c}\bar{d}\bar{e}$$

$$14. f = (\bar{a} + b + \bar{c} + d)(\bar{a} + \bar{b} + c + \bar{e})(a + \bar{b} + \bar{c} + \bar{d})$$

$$15. f = \bar{a}\bar{c}\bar{d}\bar{e} + \bar{a}\bar{b}\bar{d}\bar{e} + a\bar{c}\bar{d}\bar{e} + b\bar{c}\bar{d}\bar{e}$$

$$16. f = ab + \bar{a}\bar{b} + ad\bar{e} + a\bar{d}\bar{e}$$

Виконати синтез одноктактної схеми

17. Схема має 4 вхідних сигнали a, b, c, d . Вихідний сигнал схеми дорівнює одиниці тільки в тому разі, коли один з вхідних сигналів дорівнює одиниці (дорівнює одиниці один будь-який із вхідних сигналів).

18. Схема має 4 вхідних сигнали a, b, c, d . Вихідний сигнал схеми дорівнює одиниці тільки в тому разі, коли два вхідних сигнали дорівнюють одиниці (дорівнює одиниці два будь-яких вхідних сигнали).

19. Схема має 4 вхідних сигнали a, b, c, d . Вихідний сигнал схеми дорівнює одиниці тільки в тому разі, коли три вхідних сигналів дорівнюють одиниці (дорівнює одиниці три будь-яких вхідних сигнали).

20. Схема має три вхідних сигнали a, b, c . Вихідний сигнал схеми дорівнює одиниці тільки в тому разі, коли непарна кількість вхідних сигналів

дорівнює одиниці (дорівнюють одиниці один або три будь-яких вхідних сигнали).

21. Схема має три вхідних сигнали a , b , c . Вихідний сигнал схеми дорівнює одиниці тільки в тому разі, коли парна кількість вхідних сигналів дорівнює одиниці (дорівнюють одиниці два будь-яких вхідних сигнали, коли всі вхідні сигнали дорівнюють 0 вихідний також дорівнює 1).

22. Схема має 4 вхідних сигнали a , b , c , d . Вихідний сигнал схеми дорівнює одиниці тільки в тому разі, коли три або всі 4 вхідних сигналів дорівнюють одиниці (дорівнює одиниці три будь-яких вхідних сигнали або всі вхідні сигнали дорівнюють одиниці).

23. Схема має 4 вхідних сигнали a , b , c , d . Вихідний сигнал схеми дорівнює одиниці тільки в тому разі, коли один або три вхідних сигналів дорівнюють одиниці (дорівнює одиниці один або три будь-яких вхідних сигнали, коли одиниці дорівнює два вхідних сигнали вихідний дорівнює нулю).

24. Схема має 4 вхідних сигнали a , b , c , d . Вихідний сигнал схеми дорівнює одиниці тільки в тому разі, коли два або всі чотири вхідні сигнали дорівнюють одиниці (дорівнює одиниці два або чотири будь-яких вхідних сигнали).

25. Схема має 4 вхідних сигнали a , b , c , d . Вихідний сигнал схеми дорівнює одиниці тільки в тому разі, коли всі чотири або будь-який один вхідний сигнал дорівнює одиниці (дорівнює одиниці будь-який один або чотири вхідних сигнали).

26. Схема виконує вибір “за більшістю” з чотирьох сигналів a , b , c , d . Сигнал на виході схеми збігається зі значеннями більшості вхідних сигналів.

27. Схема виконує вибір “за меншістю” з чотирьох сигналів a , b , c , d . Сигнал на виході схеми збігається зі значеннями меншості вхідних сигналів.

28. Схема порівнює за величиною два двійкових дворозрядних числа ab і cd , де a і c – старші розряди. Вихідний сигнал схеми дорівнює одиниці тільки у тому разі, якщо число ab більше числа cd .

29. Схема порівнює за величиною два двійкових дворозрядних числа ab і cd , де a і c – старші розряди. Вихідний сигнал схеми дорівнює одиниці тільки у тому разі, якщо число cd більше числа ab .

30. Схема порівнює за величиною два двійкових дворозрядних числа ab і cd , де a і c – старші розряди. Вихідний сигнал схеми дорівнює одиниці тільки у тому разі, якщо число cd дорівнює числу ab .

31. Схема порівнює за величиною два двійкових дворозрядних числа ab і cd , де a і c – старші розряди. Вихідний сигнал схеми дорівнює одиниці тільки у тому разі, якщо число cd не дорівнює числу ab .

32. Схема виконує функції однорозрядного повного суматора, тобто має три вхідних сигнали і два вихідних. Вхідні сигнали: a , b – доданки, c – сигнал переносу з попереднього розряду. Вихідні сигнали: X – сума; Y – сигнал переносу в наступний розряд.

Виконати синтез багатотактної схеми

33. Схема має два вхідні сигнали a і b та один вихідний сигнал X . Якщо $b=1$, то значення X збігається зі значенням a , якщо $b=0$, то X зберігає своє останнє значення, яке він мав до того моменту часу, коли сигнал b набув значення 0.

34. Схема має три вхідні сигнали a , b , c і один вихідний сигнал X . Сигнал $X=1$ тільки в тому разі, якщо є сигнал a і його подано раніше, ніж сигнали b і c .

35. Схема має три вхідні сигнали a , b , c і один вихідний сигнал X . Сигнал $X=1$, якщо всі три вхідні сигнали дорівнюють одиниці, причому сигнал a набуває значення 1 першим.

36. Схема має три вхідні сигнали a , b , c і один вихідний сигнал X . Сигнал $X=1$ тільки в тому разі, коли всі три вхідні сигнали дорівнюють одиниці, причому сигнал a набуває значення 1 останнім. Після того, як сигнал X набуває значення 1, він зберігає це значення доти, поки $a=1$, незалежно від значення сигналів b і c .

37. Схема має три вхідні сигнали a , b , c і один вихідний сигнал X . Вихідний сигнал набуває значення 1 тільки в тому разі, якщо всі три вхідні

сигнали дорівнюють одиниці і вони надходили в такій послідовності: спочатку сигнал a , потім b і останнім c .

38. Схема має три вхідні сигнали a, b, c і один вихідний сигнал X . Вихідний сигнал X набуває значення одиниця, якщо всі три вхідні сигнали дорівнюють одиниці, причому сигнал a подано другим, тобто після сигналу b або c . Після того, як сигнал X набуває значення 1, він зберігає це значення доти, поки $a = 1$, незалежно від значень сигналів b і c .

39. Схема має три вхідні сигнали a, b, c і один вихідний X ; вихідний сигнал $X = 1$ тільки в тому разі, якщо всі три вхідні сигнали дорівнюють одиниці, причому сигнал a подано другим, тобто після сигналу b або c .

40. Схема має три вхідні сигнали a, b, c і один вихідний сигнал X . Вихідний сигнал набуває значення 1 тільки в тому разі, якщо всі три вхідні сигнали дорівнюють одиниці і вони надходили в такій послідовності: спочатку сигнал c , потім b і останнім a .

41. Схема має три вхідні сигнали a, b, c і один вихідний сигнал X . Вихідний сигнал набуває значення 1 тільки в тому разі, якщо сигнал a надходить після сигналу b і зберігається до надходження сигналу c .

42. Схема має два вхідні сигнали a і b та один вихідний сигнал X . Якщо $b = 1$, то значення X збігається з інверсією значенням a , якщо $b = 0$, то X зберігає своє останнє значення, яке він мав до того моменту, коли сигнал b набув значення 0.

43. Схема має три вхідні сигнали a, b, c і один вихідний сигнал X . Сигнал $X = 1$ тільки в тому разі, якщо є сигнал c і його подано раніше, ніж сигнали a і b .

44. Схема має три вхідні сигнали a, b, c і один вихідний сигнал X . Сигнал $X = 1$, якщо всі три вхідні сигнали дорівнюють одиниці, причому сигнал c набуває значення 1 першим.

45. Схема має три вхідні сигнали a, b, c і один вихідний сигнал X . Сигнал $X = 1$, якщо всі три вхідні сигнали дорівнюють одиниці, причому сигнал c набуває значення 1 останнім.

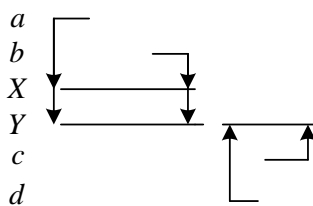
46. Схема має три вхідні сигнали a, b, c і один вихідний X ; вихідний сигнал $X = 1$ тільки в тому разі, якщо всі три вхідні сигнали дорівнюють одиниці, причому сигнал b подано другим, тобто після сигналу a або c .

47. Схема має три вхідні сигнали a, b, c і один вихідний сигнал X . Вихідний сигнал набуває значення 1 тільки в тому разі, якщо всі три вхідні сигнали дорівнюють одиниці і вони надходили в такій послідовності: спочатку сигнал c , потім b і останнім a .

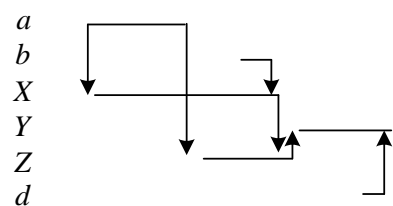
48. Схема має три вхідні сигнали a, b, c і один вихідний сигнал X . Сигнал $X = 1$ тільки в тому разі, коли всі три вхідні сигнали дорівнюють одиниці, причому сигнал a набуває значення 1 останнім. Після того, як сигнал X набуває значення 1, він зберігає це значення доти, поки $c = 1$, незалежно від значення сигналів a і b .

Умови роботи схем подано циклограмами

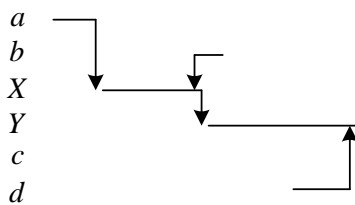
49



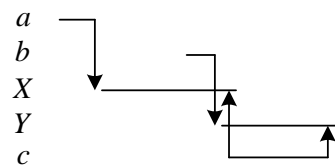
50



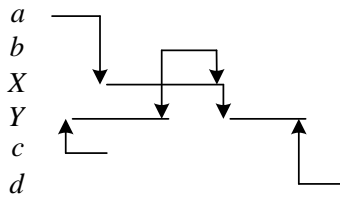
51



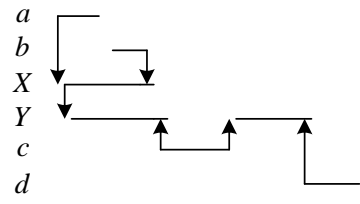
52



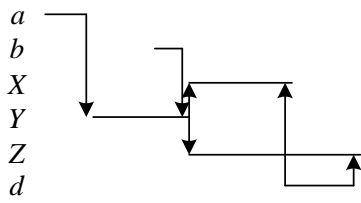
53



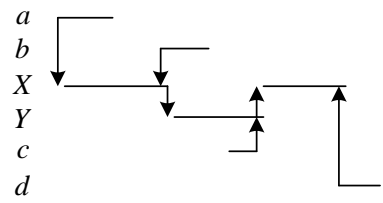
54



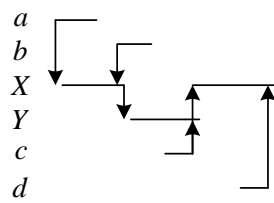
55



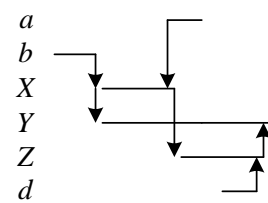
56



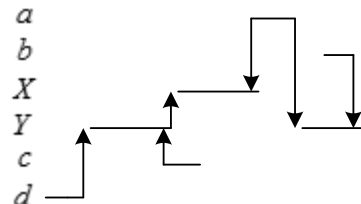
57



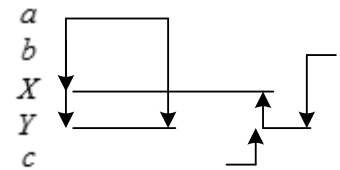
58



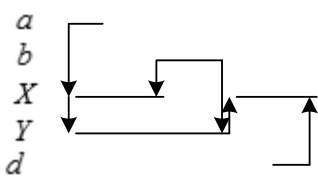
59



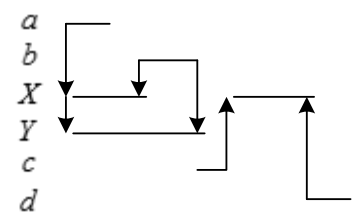
60



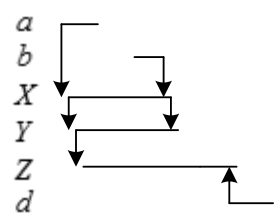
61



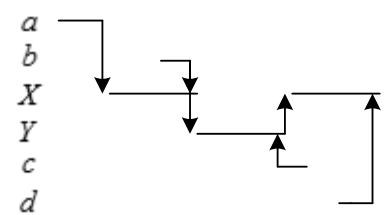
62



63



64



1.3 Опис лабораторної установки

Функціональну схему лабораторного стенду представлено на рисунку 1.6. За допомогою широкого набору мікросхем складається робоча схема, яка описує роботу системи відповідно до заданої умови роботи схеми.

За допомогою тумблерів SB1, SB2 подається напруга живлення на макетні плати. Про подачу живлення сповіщають сигнальні лампи HL1, HL2, які розміщені над тумблерами.

Схема збирається за допомогою мікросхем, тактових кнопок, світлодіодів, резисторів на монтажній платі. Про її роботу можуть свідчити світлодіоди які потрібно встановити на вихідних сигналах схеми.

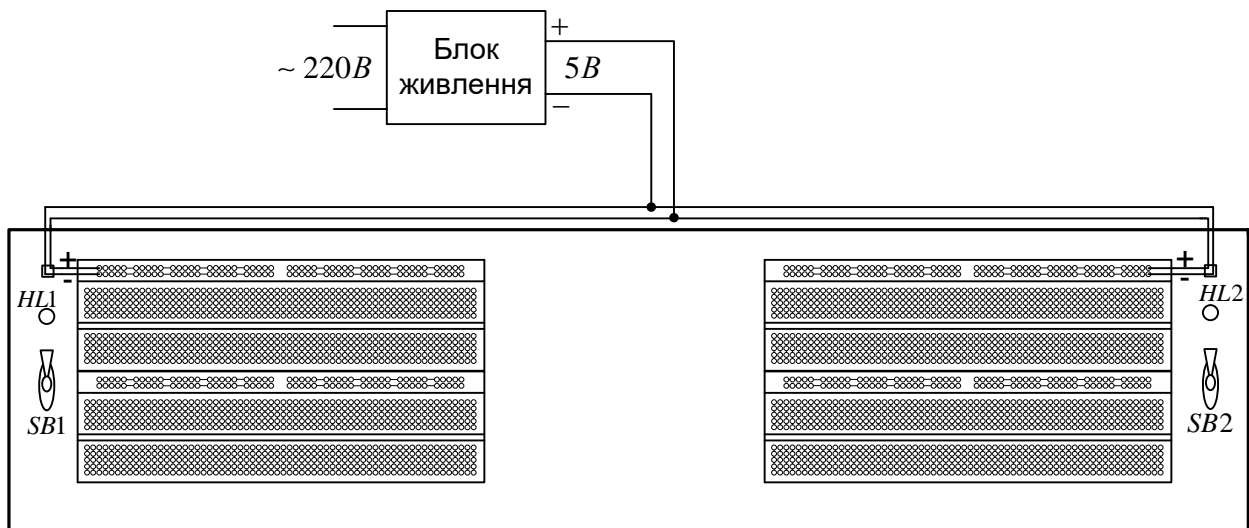
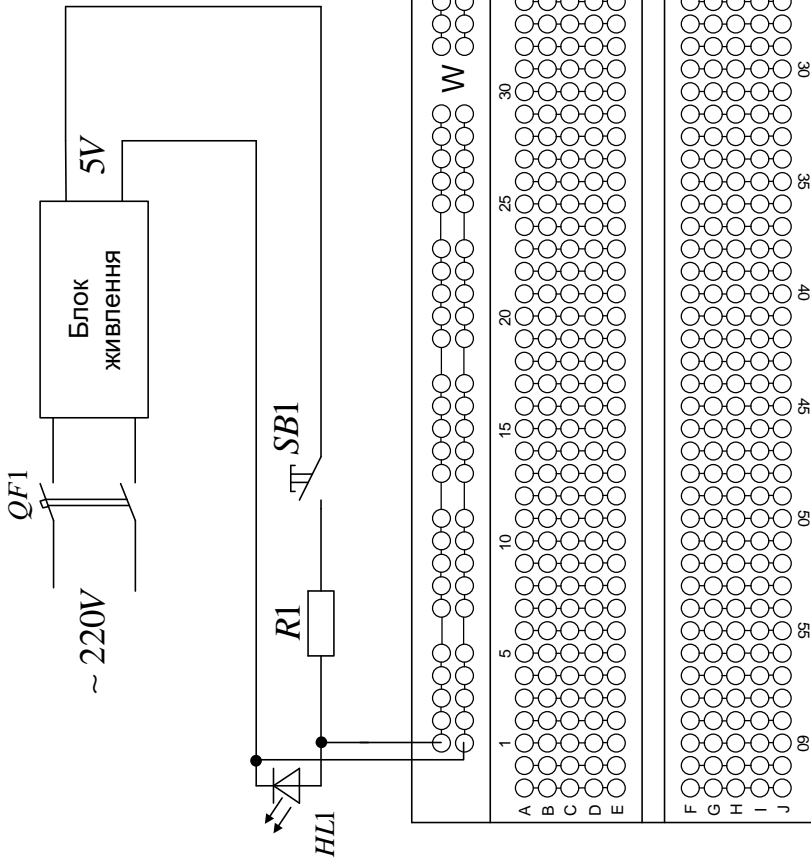
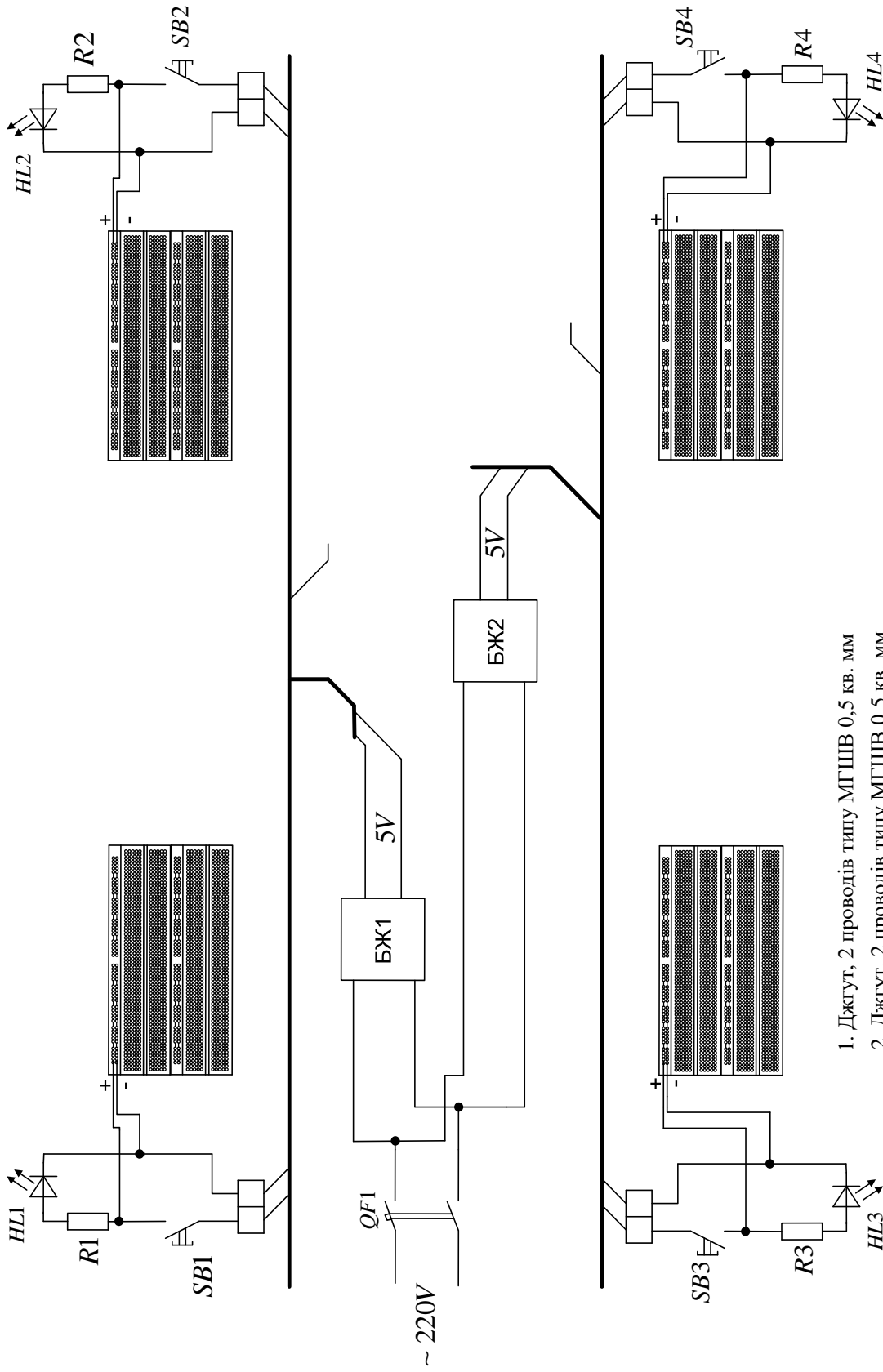


Рисунок 1.6 – Функціональна схема лабораторного стенду

На панелі розташовано два робочих місця, кожне з яких містить дві макетні плати, що забезпечують достатньо місця для монтування мікросхем. Вихідний струм блоку живлення – 500 мА. Один блок живлення використаний для двох робочих місць.

На рис. 1.7 зображено електричну принципову схему одного робочого місця та схему підключень лабораторного стенду на рис. 1.8.





- 1. Джгут, 2 проводів типу МГШВ 0,5 кв. мм
- 2. Джгут, 2 проводів типу МГШВ 0,5 кв. мм

Рисунок 1.8 – Схема підключень

1.4 Методичні вказівки до виконання лабораторної роботи

Для виконання лабораторних робіт використовуються дискретні мікросхеми серії 74НС02, 74НС04, 74НС10, 74НС11, 74НС20, 74НС32, 74НС153, RS-тригери серії CD4043В, та мультиплектори на вісім інформаційних входів серії К555КП15 та на чотири входи 74НС153.

1.4.1 Логічна мікросхема 74НС02

Мікросхема містить 4 дискретні елементи по 2 входи, які виконують функцію «АБО-НІ» та споживає струм 20 мА. Живлення мікросхеми – 5 В. Розташування входів/виходів мікросхеми показано на рис. 1.9.

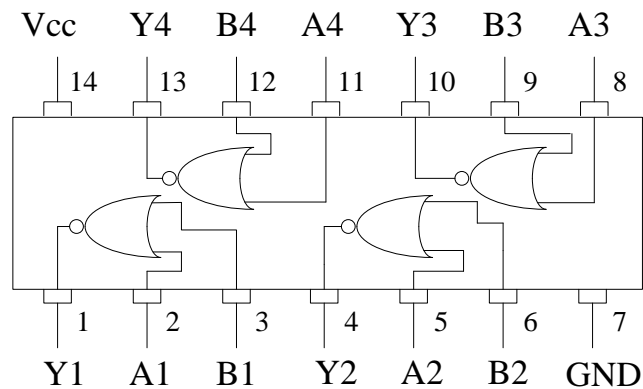


Рисунок 1.9 – Мікросхема 74НС02

На рисунку прийняті наступні позначення:

Vcc – живлення мікросхеми +5 В;

GND – «земля» живлення;

A1-A4, B1-B4 – входи дискретних логічних елементів;

Y1-Y4 – виходи дискретних логічних елементів.

1.4.2 Логічна мікросхема 74НС04

Мікросхема містить 8 інверторів, що на виході дають інверсію вхідного сигналу. Струм живлення – 25 мА, напруга живлення – 5 В. Розташування входів/виходів мікросхеми показано на рис. 1.10.

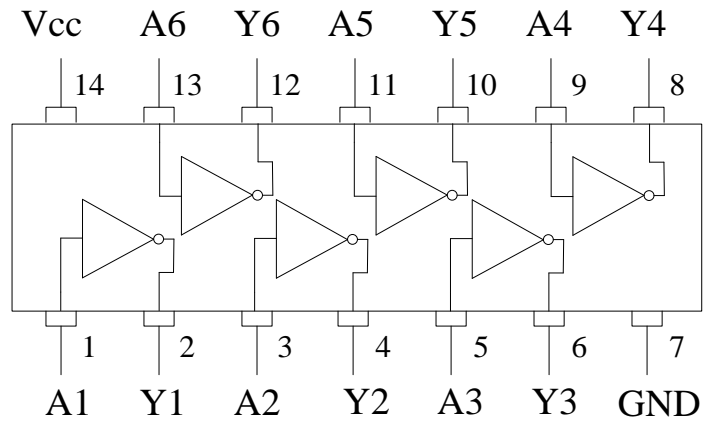


Рисунок 1.10 – Мікросхема 74НС04

1.4.3 Логічна мікросхема 74НС10

Мікросхема містить 3 дискретних елементи по 3 входи, які виконують функцію «І-НІ» та споживає 15мА. Живлення мікросхеми – 5 В. Розташування входів/виходів мікросхеми показано на рис. 1.11.

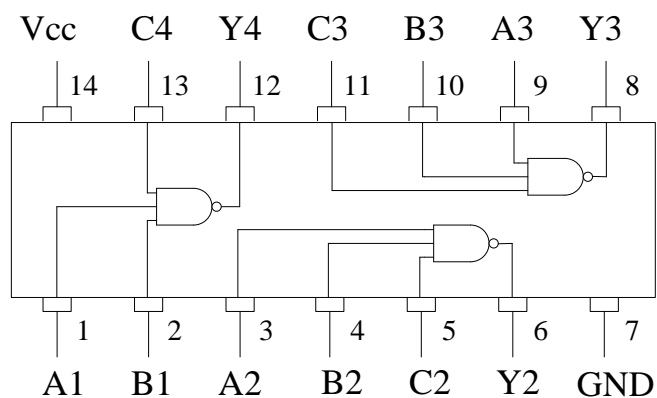


Рисунок 1.11 – Мікросхема 74НС10

1.4.4 Логічна мікросхема 74НС11

Мікросхема містить 3 дискретних елементи по 3 входи, які виконують функцію «І» та споживає 50 мА. Живлення мікросхеми – 5 В. Розташування входів/виходів мікросхеми показано на рис. 1.12.

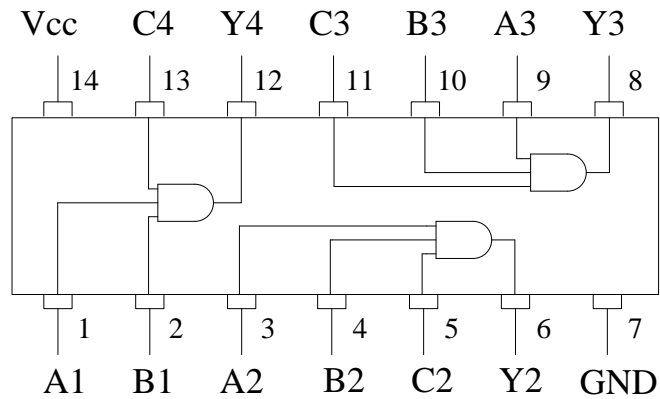


Рисунок 1.12 – Мікросхема 74НС11

1.4.5 Логічна мікросхема 74НС20

Мікросхема містить 2 дискретних елементи по 4 входи, які виконують функцію «І-НІ» та споживає 50 мА. Живлення мікросхеми – 5 В. Розташування входів/виходів мікросхеми показано на рис. 1.13.

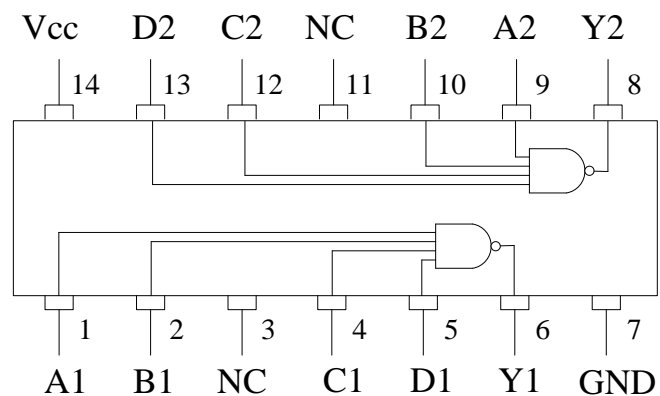


Рисунок 1.13 – Мікросхема 74НС20

1.4.6 Логічна мікросхема 74НС32

Мікросхема містить 4 дискретні елементи по 2 входи, які виконують функцію «АБО» та споживає 50 мА. Живлення мікросхеми – 5 В. Розташування входів/виходів мікросхеми показано на рис. 1.14.

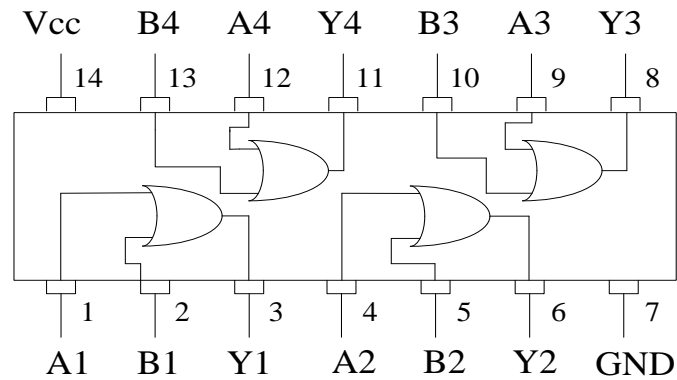


Рисунок 1.14 – Мікросхема 74НС32

1.4.7 Приклади синтезу та складання схем

Приклад 1. Скласти схему за заданим рівнянням:

$$f = abc + \bar{c}de + ace$$

Якщо скласти карту Карно до даного рівняння, видно що функція не мінімізується, тому складаємо схему на основі рівняння:

Імітацію спрацювання вихідної функції f здійснюємо за допомогою світлодіоду VD1. Резистори R1-R5 використовуються для «підтяжки» входів мікросхем до логічного «0», коли кнопки a , b , c , d та e розімкнені. Резистор R6 розрахований для зниження струму, що проходить через світлодіод VD1, для уникнення його перегорання.

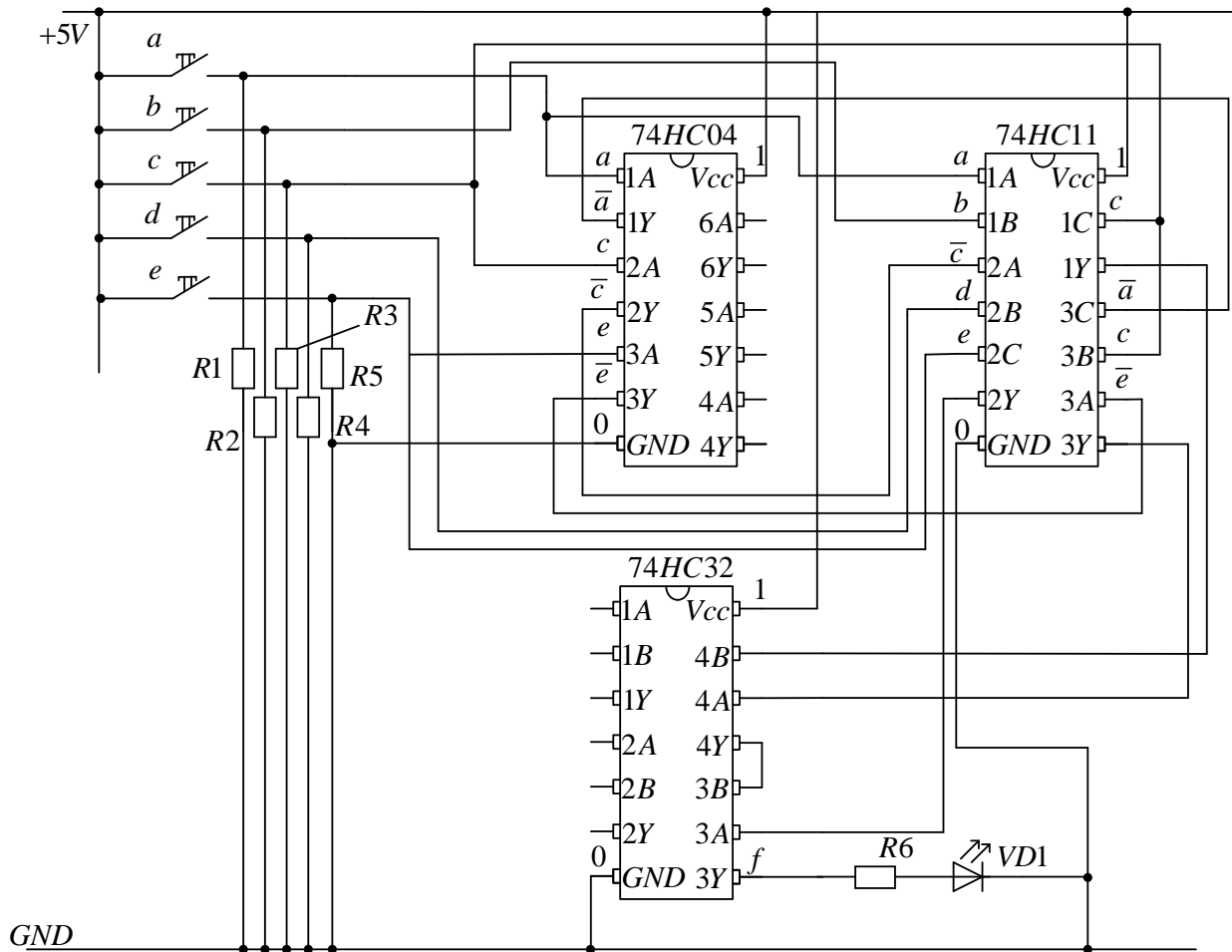


Рисунок 1.15– Схема на дискретних логічних елементах для прикладу 1

Приклад 2. Виконати синтез однократної схеми

Схема виявляє неприпустимі комбінації в двійково-десятковому коді 8421. Умова роботи схеми: схема має чотири входи і один вихід. На входи надходять сигнали 0 і 1, які є розрядами двійкового числа. Якщо це число менше або дорівнює 9, то вихідний сигнал схеми дорівнює нулю, якщо числа на вході більше 9, вихідний сигнал дорівнює одиниці.

За заданими умовами роботи складаємо таблицю істинності (табл. 1.4), в якій прийнято, що *a*- старший, а *d* - молодший розряди вхідного двійкового числа.

Таблиця 1.4 – Таблиця істинності за умовою

Вхідні сигнали				Вихідний сигнал
<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>f</i>
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

За даними табл. 1.4 записуємо вираз функції f у доконаній диз'юнктивній нормальній формі (ДДНФ):

$$f = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}cd + ab\bar{c}\bar{d} + ab\bar{c}d + abc\bar{d} + abcd.$$

Застосувавши закон склеювання, мінімізуємо записану функцію

$$f = \bar{a}\bar{b}c + ab\bar{c} + abc = ac + ab.$$

Імітацію спрацювання вихідної функції f здійснюємо за допомогою світлодіоду VD1. Резистори R1-R3 використовуються для «підтяжки» входів мікросхем до логічного «0», коли кнопки a , b та c розімкнені. Резистор R4 розрахований для зниження струму, що проходить через світлодіод VD1, для уникнення його перегорання.

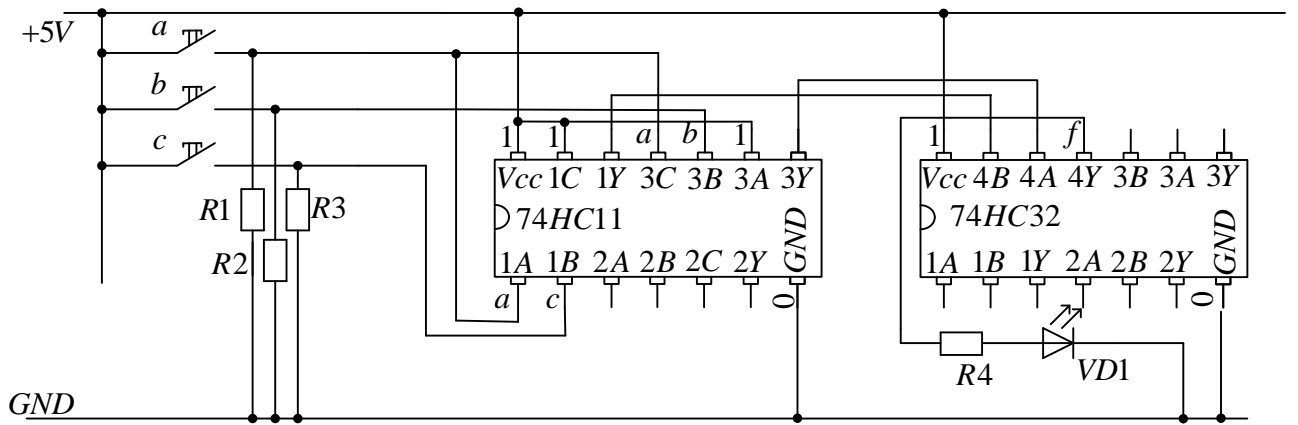


Рисунок 1.16 – Схема на дискретних логічних елементах для прикладу 2

Приклад 3. Умову роботи схеми задано циклограмою на рис. 1.17.

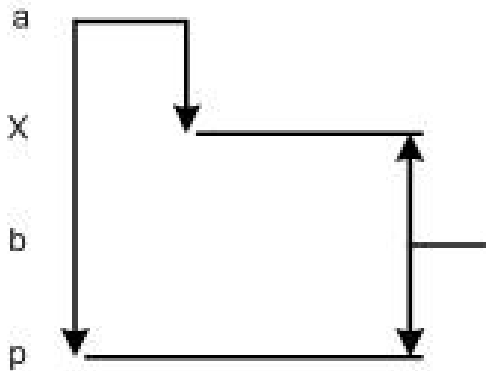


Рисунок 1.17 –

Задана циклограма

Умова не спрацювання проміжної змінної:

$$f_p'' = \bar{b}.$$

Функція вмикання проміжної змінної:

$$p = (a + p)\bar{b}.$$

Імітацію спрацювання вихідної функції f здійснюємо за допомогою світлодіоду VD1. Резистори R1, R2 використовуються для «підтяжки» входів мікросхем до логічного «0», коли кнопки a та b розімкнені. Резистор R3 розрахований для зниження струму, що проходить через світлодіод VD1, для уникнення його перегорання.

З отриманих результатів будемо схему на логічних елементах.

Визначаємо умова спрацювання:

$$f_x' = \bar{a}.$$

Умова неспрацювання:

$$f_x'' = \bar{b}.$$

Формула вмикання:

$$X = f' \cdot \bar{f}'' = \bar{a}\bar{b}p.$$

Умова спрацювання проміжної змінної:

$$f_p' = a + p.$$

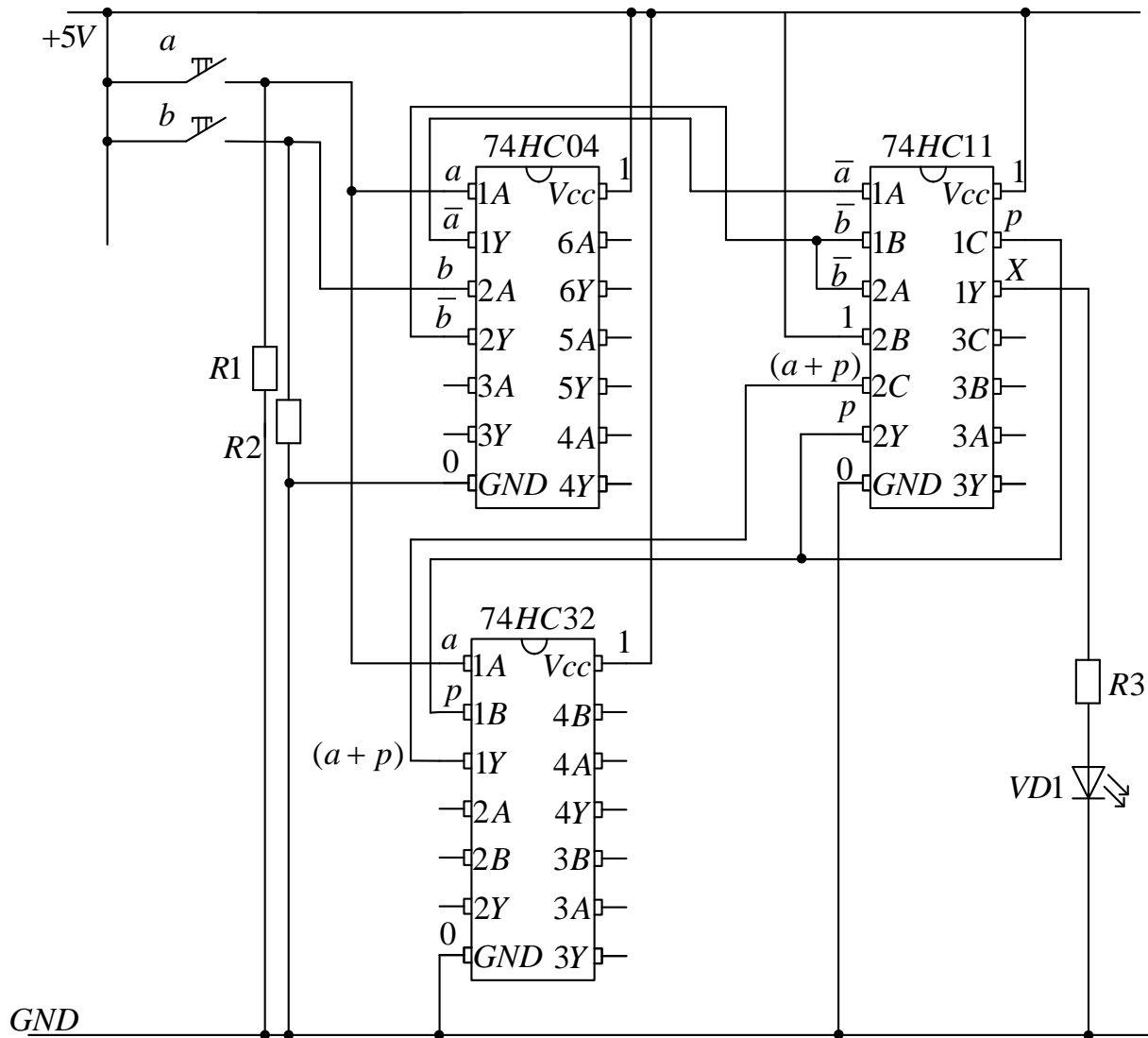


Рисунок 1.18 – Схема на дискретних логічних елементах для прикладу 3

1.4.8 Основи роботи в середовищі Multisim

Multisim – це система автоматизованого проектування (САПР) схем електричних принципів від фірми National Instruments, яка дозволяє також виконувати симуляцію режимів роботи спроектованих схем.

Загальний вигляд робочого вікна середовища Multisim представлено на рис. 1.19. Воно складається з наступних основних елементів: 1 – меню пристроїв; 2 – меню вимірювальних приладів; 3 – вікно робочих файлів; 4 – робоча область для створення схем.

Для того, щоб розпочати роботу, необхідно у меню пристроїв 1 вибрати необхідний розділ, а в ньому знайти необхідно мікросхему або потрібний елемент. Перелік розділів наведений на рис. 1.20.

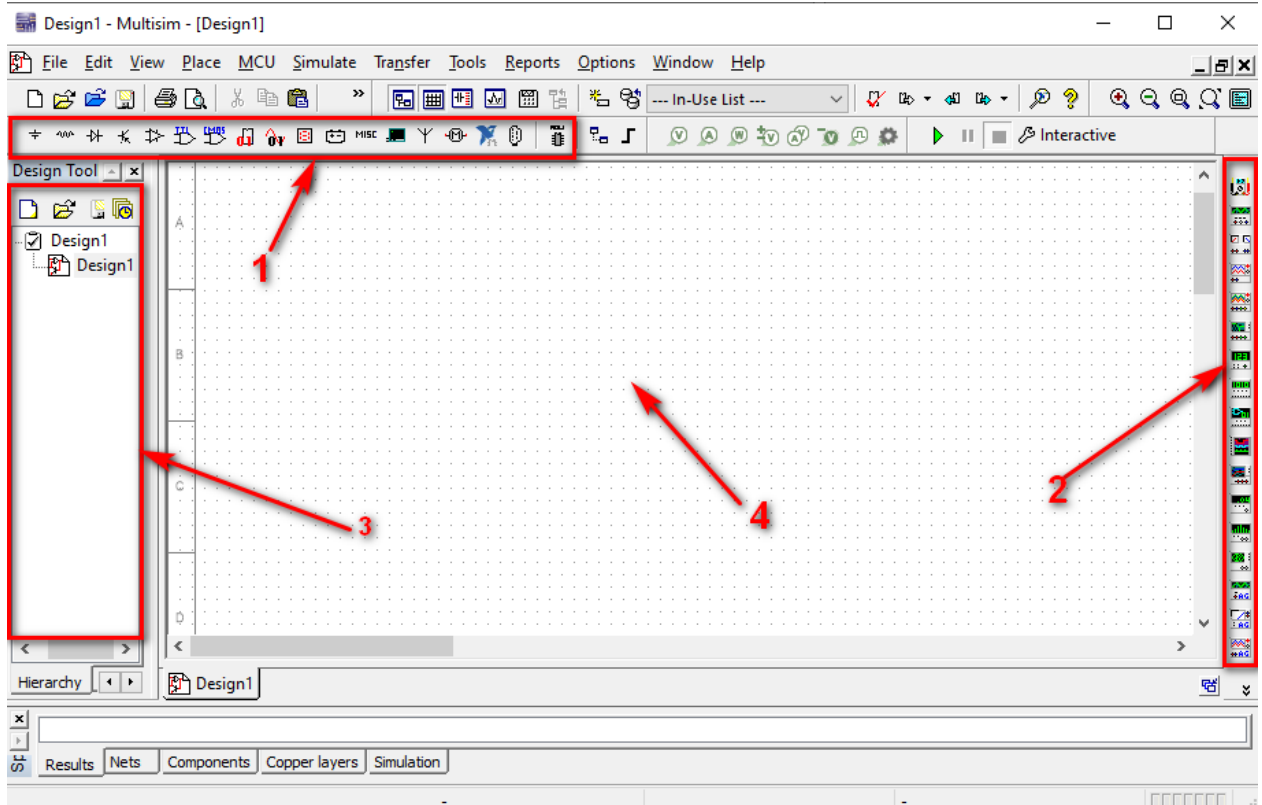


Рисунок 1.19 – Робоче вікно середовища Multisim

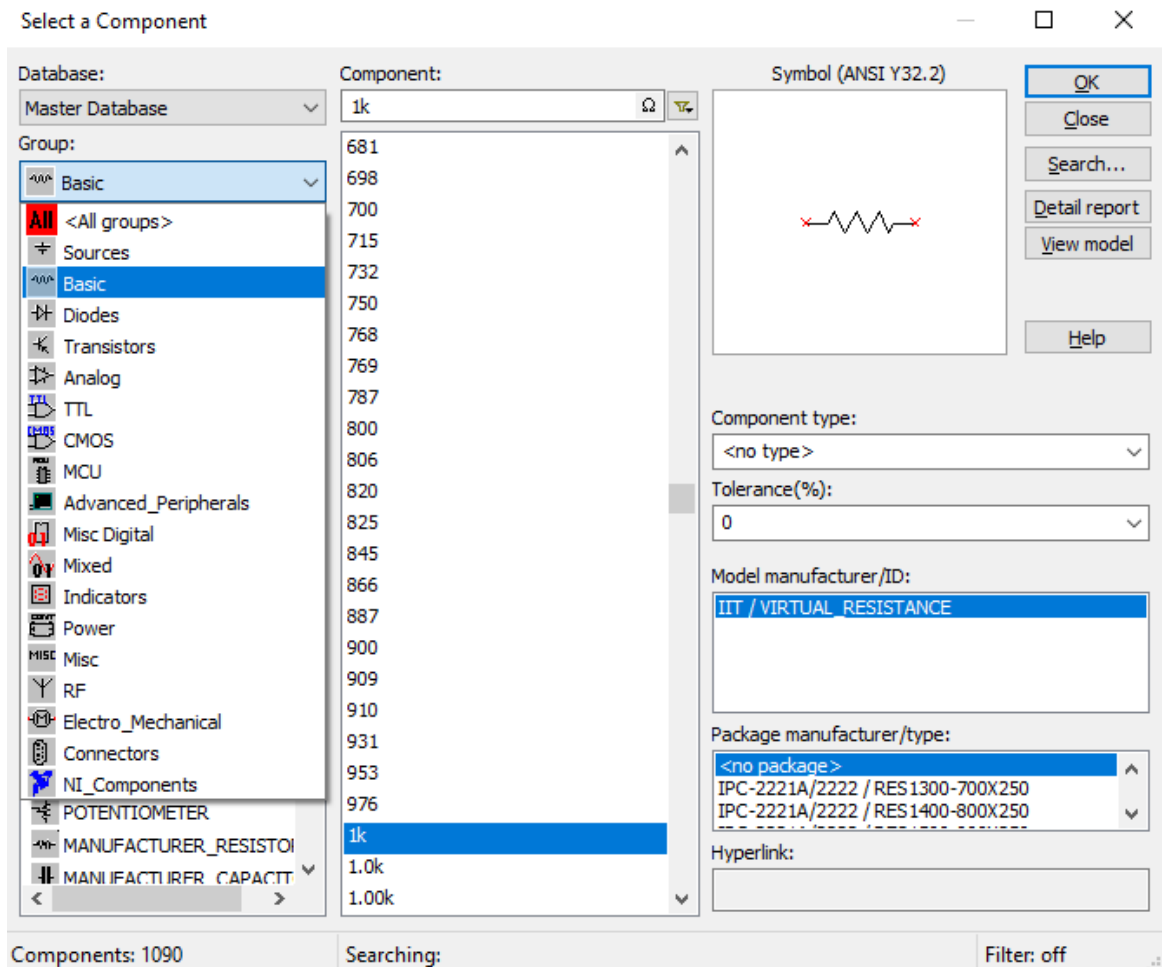



Рисунок 1.20 – Меню пристроїв із вибраною вкладкою «Basic»

Знайти необхідний елемент можна також за допомогою функції пошуку. Для цього необхідно вибрати у меню пристроїв пункт «All groups», а потім у вікні «Component» ввести назву необхідної мікросхеми або приладу.

Для симуляції розробленої схеми необхідно натиснути на кнопку у вигляді зеленої стрілки  «Run». Для зупинки симуляції треба натиснути кнопку «Stop».

Приклад проекту схеми електричної принципової на інтегральних мікросхемах наведений на рис. 1.21.

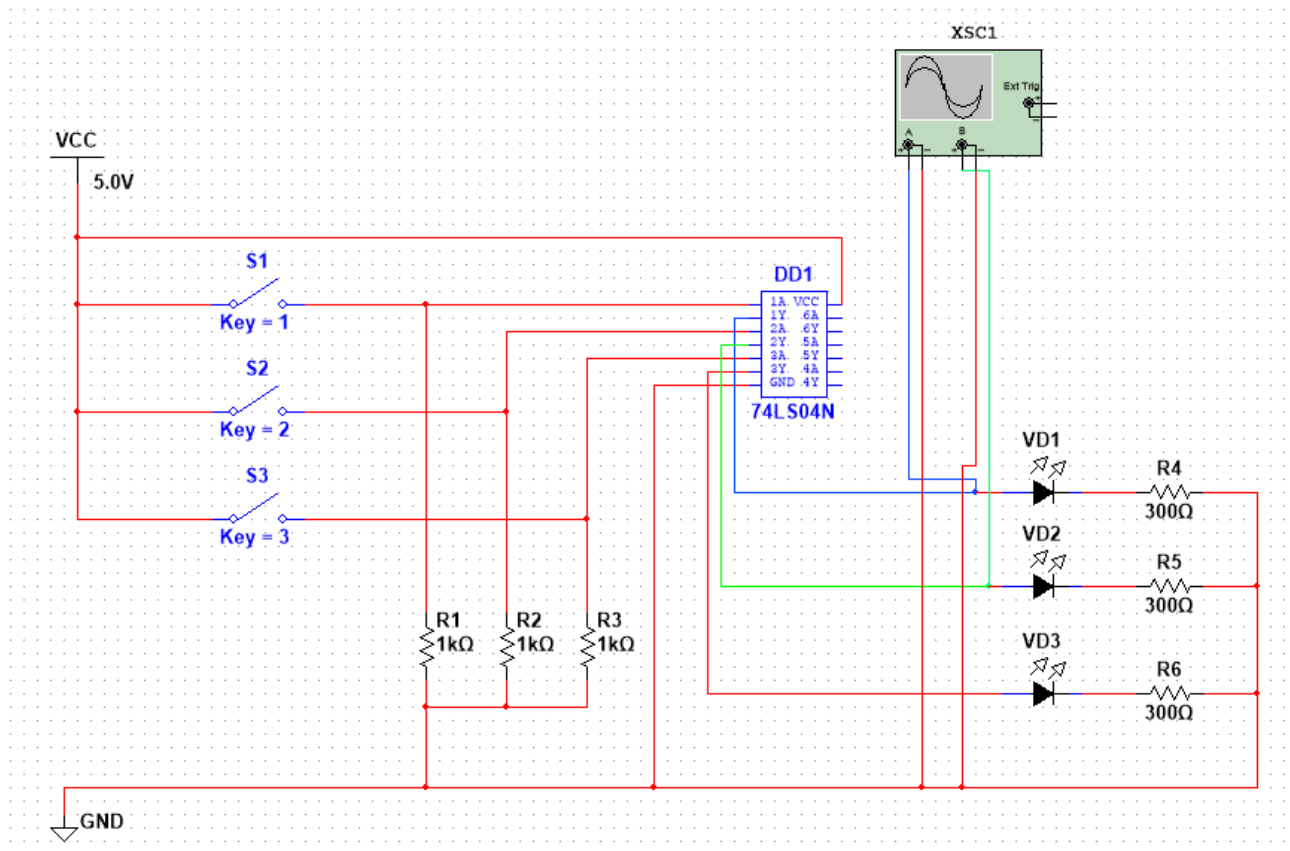


Рисунок 1.21 – Приклад схеми електричної принципової на інтегральних мікросхемах в середовищі Multisim

Процедуру створення проекту схеми можна подивитися у відео-уроці до лабораторної роботи №1 за посиланням <https://youtu.be/4e8h-ntdmwY>.

Процедуру складання схеми складання схеми на логічних мікросхемах та монтажних платах можна подивитися за посиланням <https://youtu.be/FSBpo0BFFmY>.

1.5 Зміст звіту

Звіт має містити такі матеріали дослідження кожної схеми:

- 1) титульний аркуш із зазначенням назви дисципліни та лабораторної роботи (див. Додаток А);
- 2) мету роботи, програму роботи та умови роботи схем;
- 3) логічний синтез схеми;
- 4) принципові схеми на наявних у даній лабораторній роботі логічних елементах, з підписами викладача;
- 5) перелік елементів до кожної схеми;
- 6) схеми у програмному забезпеченні Multisim з результатами її симуляції;
- 7) висновки про відповідність роботи схеми заданим умовам.

1.6 Контрольні запитання

1. Чим відрізняються одноктактні схеми від багатотактних?
2. Як виконується синтез одноктактних схем?
3. Чому не можна за допомогою таблиці істинності описати роботу багатотактної схеми?
4. Якими методами можна виконувати синтез багатотактних схем?
5. У чому полягає суть перевірок реалізованості циклограм?
6. Як обрати кількість проміжних змінних, виконуючи синтез багатотактних схем методом таблиць переходів і карт Карно?
7. Чим відрізняються правила складання карт Карно для проміжних і вихідних змінних?

Лабораторна робота № 2

ЛОГІЧНИЙ СИНТЕЗ ТА ДОСЛІДЖЕННЯ СХЕМ АВТОМАТИКИ НА ТРИГЕРАХ ТА МУЛЬТИПЛЕКСОРАХ

Тривалість лабораторної роботи – 4 год.

Тривалість домашньої роботи – 4 год.

Мета роботи – практично перевірити різні методи синтезу одноканальних та багатоканальних схем на тригерах та мультиплексорах, навчитися складати схеми електричні принципи та інтегральних мікросхемах тригерів та мультиплексорів з використанням монтажних плат, кнопок, світлодіодів та резисторів.

2.1. Основні теоретичні відомості

2.1.1 Синтез схем на мультиплексорах-селекторах

Роботу мультиплексора-селектора описують такою матричною рівністю

$$[A][Z]=[f],$$

де $[A]$ – матриця вхідних даних; $[Z]$ – матриця селектування; $[f]$ – матриця вихідної змінної.

Матриця вхідних даних має вигляд матриці-рядка. Наприклад, для мультиплексора-селектора з двома селекторними лініями

$$[A]=[a_0a_1a_2a_3],$$

а з трьома

$$[A]=[a_0a_1a_2a_3a_4a_5a_6a_7],$$

де a_0, a_1, a_2, \dots – значення змінних на відповідних вхідних лініях.

Матриця селектування – це прямокутна матриця, кількість рядків якої дорівнює N , а кількість стовпців – 2^N , де N – кількість селекторних ліній. Стовпці матриці являють собою комбінації сигналів на селекторних лініях і розташовуються у такому порядку, що зображують послідовний запис двійкових N -розрядних чисел від 0 до $2^N - 1$. При цьому старші розряди чисел записуються у верхньому рядку.

Так, для мультиплексора з трьома селекторними лініями

$$[Z] = \begin{bmatrix} 00001111 \\ 00110011 \\ 01010101 \end{bmatrix}.$$

Функція

$$[f] = [A][Z],$$

яку виконує мультиплексор, визначається матрицею вхідних даних $[A]$, тому синтез схем на мультиплексорах по суті зводиться до визначення компонентів цієї матриці.

При синтезі однотактних схем для визначення матриці $[A]$ використовують карту Карно, складену за умовами роботи схеми.

При побудові багатотактних схем на мультиплексорах-селекторах їх вихідні сигнали використовуються як проміжні змінні. Кожний стан схеми визначається власною комбінацією цих змінних. Вихідні змінні схеми в цьому випадку є комбінаційними функціями проміжних змінних. В багатотактних схемах повинна зберігатися інформація про попередні стани схеми, тобто повинні бути зворотні зв'язки, що забезпечують запам'ятовування значень проміжних змінних. При застосуванні мультиплексорів-селекторів зворотний зв'язок одержують за рахунок з'єднання виходів мультиплексорів-селекторів з селекторними лініями.

Схему на мультиплексорах-селекторах 8×1 із зворотними зв'язками за трьома змінними показано на рис. 2.1. Тут великими літерами P_1 , P_2 , P_3

позначено вихідні сигнали мультиплексорів-селекторів , а малими p_1 , p_2 , p_3 – відповідні сигнали на селекторних лініях.

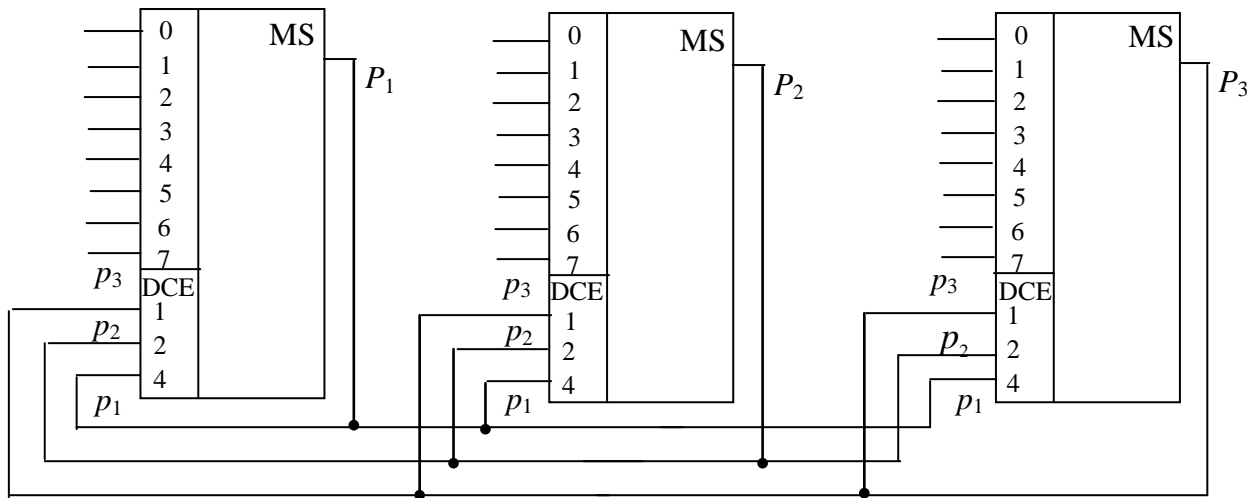


Рисунок 2.1 – Схема на мультиплексорах-селекторах із зворотними зв'язками за трьома змінними

2.1.2 Синтез схем автоматики на тригерах

При синтезі схем на тригерах умови роботи багатотактних схем подаються у вигляді графів переходів. Граф переходів – це графічне зображення послідовності роботи багатотактної схеми. Елементами графа є *вершини* і *ребра*. Вершини відповідають станам схеми і позначаються кружками. Ребра – це лінії із стрілками, що з'єднують вершини і показують напрям переходу з одного стану схеми в інший.

Кількість вершин графа при синтезі асинхронних схем на RS – тригерах визначається з умови $2^n \geq S$, де S - кількість станів схеми; 2^n – кількість вершин графа; n – кількість тригерів.

Вершини графа рекомендується розміщувати так, щоб при $n=2$ вони створювали конфігурацію 2×2 , при $n=3$ – конфігурацію 4×2 , а при $n=4$ - конфігурацію 4×4 . Вигляд графа переходів при $n=3$ показано на рис. 2.2.

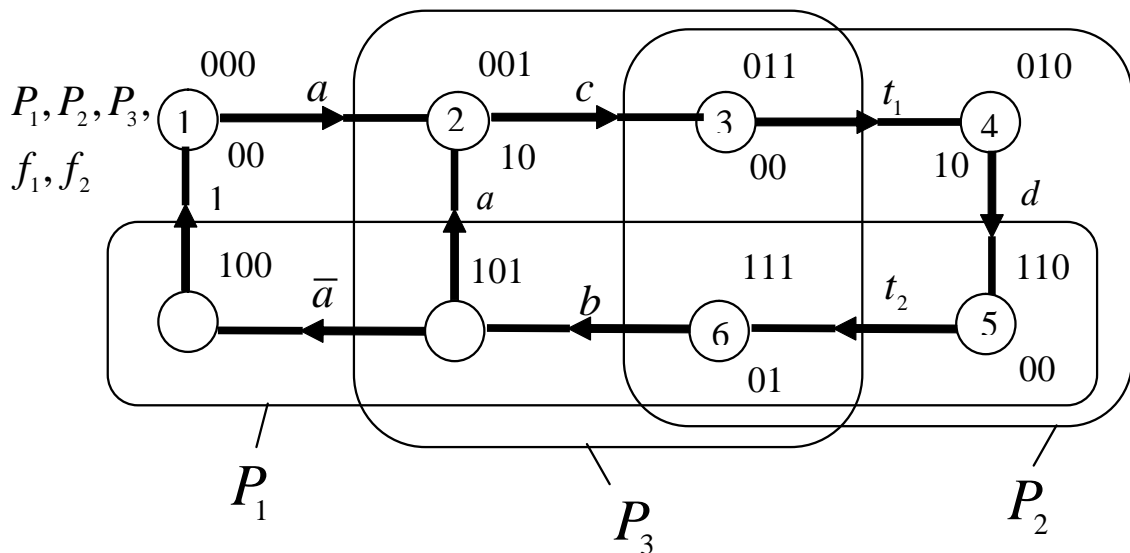


Рисунок 2.2 – Вигляд графа переходів при $n=3$

Вихідні сигнали тригерів виконують роль проміжних змінних, тому тригери та їх вихідні сигнали позначаються буквами P_1, P_2, \dots . Ці позначення записують зверху ліворуч від графа.

Кожна вершина графа кодується набором значень вихідних сигналів тригерів. Коди вибираються так, щоб для сусідніх вершин вони відрізнялися значенням тільки однієї змінної. У вихідному стані схеми (стан очікування) звичайно приймають, що усі проміжні змінні дорівнюють нулеві. Значення проміжних змінних для кожної вершини записуються над кружками у тій черговості, в якій записані позначення тригерів.

Вершини, між якими повинні відбуватися переходи, з'єднують ребрами із стрілками. Над стрілками або праворуч від них, якщо ребра спрямовано вертикально, записують позначення вхідних сигналів, що спричинюють ці переходи.

При побудові схем на асинхронних RS – тригерах переходи можна робити тільки між сусідніми вершинами. Якщо ця умова не виконується, то необхідно передбачити так звані природні переходи (за рахунок подавання вхідного сигналу одиниця) через проміжні нестійкі стани.

Як приклад побудуємо граф переходів, що описує роботу схеми керування механізмом M , який працює в режимі циклів, що повторюються. У вихідному положенні механізму натиснутий кінцевий вимикач $SQ1$. Цикл роботи починається після надходження команди "Пуск". Механізм переміщується в

положення, що фіксується кінцевим вимикачем $SQ2$, стоїть протягом часу Δt_1 , переміщується в положення, що фіксується кінцевим вимикачем $SQ3$, стоїть протягом часу Δt_2 , а потім повертається в вихідне положення. Для повторення циклу необхідно знов подати команду "Пуск". Якщо ця команда надходить безперервно, то після відпрацьовування одного циклу автоматично починається наступний.

Прийmemo такі позначення вхідних і вихідних сигналів, а також сигналів таймерів, які необхідно розглядати як вхідні сигнали для графа переходів. Вхідні сигнали: a – команда "Пуск"; b, c, d – сигнали кінцевих вимикачів $SQ1, SQ2, SQ3$ відповідно; t_1, t_2 – сигнали таймерів, що дають затримки Δt_1 і Δt_2 . Вихідні сигнали: f_1 – команда на переміщення механізму з вихідного положення, f_2 – команда на повернення механізму в вихідне положення.

Побудову графа переходів починаємо з визначення кількості станів, в яких може перебувати схема автоматичного керування. Таких станів шість: 1– вихідне положення; 2– переміщення з вихідного положення; 3– стоянка протягом часу Δt_1 ; 4 – подальше переміщення; 5– стоянки протягом часу Δt_2 ; 6 – повернення в вихідне положення.

Виходячи з кількості станів схеми $6 < 2^3$, визначаємо кількість тригерів $n=3$ і кількість вершин графа переходів $2^3=8$. Позначаємо проміжні змінні P_1, P_2, P_3 , будуємо 8 вершин графа і кодуємо їх комбінаціями значень проміжних змінних (рис. 2.2).

Кожному стану схеми ставимо у відповідність одну з вершин графа, причому стани, між якими повинен відбуватися перехід згідно з умовами роботи схеми, розміщуємо у сусідніх вершинах. Вільні вершини використовуємо для переходу з стану 6 в стан 2 під час роботи схеми в режимі циклів, що повторюються, або в стан 1 при відпрацьовуванні поодиноких циклів.

Позначення вихідних сигналів f_1 і f_2 записано під позначеннями проміжних змінних. Значення f_1 і f_2 для кожного стану схеми записуємо під відповідними вершинами графа.

Синтез схеми полягає у записі умов вмикання і скидання кожного тригера. Для цього охоплюють замкнутою лінією всі стани на графі переходів, в яких значення вихідного сигналу даного тригера дорівнює одиниці. Вхідні сигнали схеми, позначення яких стоять на ребрах, що заходять в одержану замкнуту область, встановлюють тригер в стан 1, а вхідні сигнали на ребрах, що виходять з цієї області, скидають тригер в стан 0.

Умови вмикання тригера записуються у вигляді добутку сигналу на ребрі, що заходить в область, і сигналів решти тригерів, стан яких не змінюється при переході, позначеному ребром. Наприклад, якщо сигнал на ребрі, що заходить в область з одиничним значенням вихідного сигналу тригера P_1 , дорівнює a , а тригери P_2 і P_3 не перемикаються, а зберігають стан $P_2=1$, $P_3=0$, то умова вмикання тригера P_1 записується у вигляді

$$S_{P_1} = ap_2 \bar{p}_3.$$

Якщо в замкнуту область входить кілька ребер, то умова вмикання тригера записується у вигляді суми добутків відповідних сигналів, складених для кожного ребра.

Умова скидання тригера записується аналогічно для кожного ребра, що виходить з даної області, і подається у вигляді формули R_{P_i} . Описану процедуру виконують для кожного тригера і визначають для них умови вмикання і скидання.

Застосувавши описану процедуру визначення умов вмикання і скидання тригерів, для графа переходів на рис. 2.2 отримаємо:

$$S_{P_1} = dp_2 \bar{p}_3;$$

$$R_{P_1} = a\bar{p}_2 p_3 + \bar{p}_2 \bar{p}_3;$$

$$S_{P_2} = c\bar{p}_1 p_3;$$

$$R_{P_2} = bp_1 p_3;$$

$$S_{P_3} = a\bar{p}_1 \bar{p}_2 + t_2 p_1 p_2;$$

$$R_{P_3} = t_1 \bar{p}_1 p_2 + \bar{a} p_1 \bar{p}_2.$$

Формули для вихідних сигналів f_1 і f_2 записуються як комбінаційні функції вихідних сигналів тригерів P_1, P_2, P_3 . Дійсно, функція $f_1 = 1$ в станах 2 і 4, тобто в станах, яким відповідають такі комбінації значень вихідних сигналів тригерів: $p_1=0, p_2=0, p_3=1$ і $p_1=0, p_2=1, p_3=0$. Тому

$$f_1 = \bar{p}_1 \bar{p}_2 p_3 + \bar{p}_1 p_2 \bar{p}_3.$$

Аналогічно, функція $f_2 = 1$ в стані 6 ($p_1 p_2 p_3 = 111$), тобто

$$f_2 = p_1 p_2 p_3.$$

Таймер T_1 вмикається в стані 3 ($p_1 p_2 p_3 = 011$), а таймер T_2 – в стані 5 ($p_1 p_2 p_3 = 110$), тому

$$T_1 = \bar{p}_1 p_2 p_3;$$

$$T_2 = p_1 p_2 \bar{p}_3.$$

Синтез синхронних схем на JK -тригерах виконується аналогічно. Різниця полягає у тому, що для синхронних схем при побудові графа переходів є припустимими переходи між будь-якими вершинами графа, а не обов'язково сусідніми, тобто природні переходи не потрібні. Умови вмикання тригерів записують у вигляді формули Jp_i , умови скидання – у вигляді K_{p_i} .

2.2 Програма роботи

Перед початком лабораторної роботи група ділиться на бригади відповідно до кількості робочих місць. Кожному студенту бригади необхідно вибрати один із варіантів (1-6), який містить 4 завдання, з таблиці 2.1. Для кожного з чотирьох завдань необхідно:

1) виконати логічний синтез схеми керування, використовуючи метод графопереходів для завдань 1-32 та метод мультиплексорів-селекторів для завдань 33-64 (**виконується вдома під час СРС**);

2) за отриманими алгебраїчними виразами скласти принципову схему на RS - тригерах для завдань 1-32 та на мультиплексорах-селекторах для завдань 33-64. Скласти переліки елементів для кожної схеми (**виконується вдома під час СРС**);

3) зібрати схему за вибором викладача на лабораторному стенді та перевірити її працездатність*;

4) показати зібрану схему викладачу та продемонструвати її роботу **(викладач підтверджує своїм підписом у протоколі працездатність схеми)***;

5) зібрати схему у програмному середовищі Multisim та виконати симуляцію її роботи **(виконується безпосередньо під час лабораторних робіт)**;

6) оформити звіт по лабораторній роботі та зробити висновки.

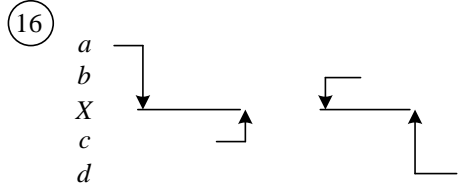
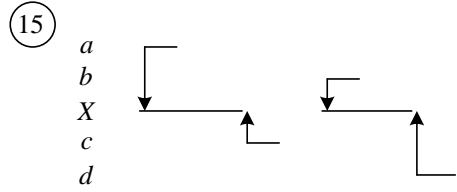
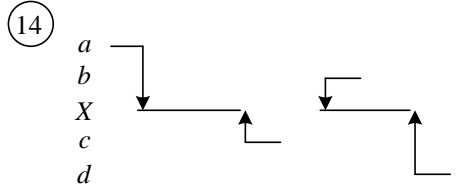
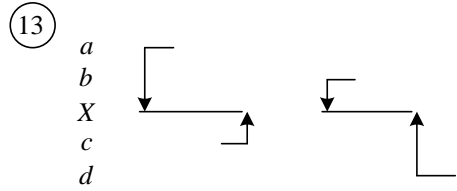
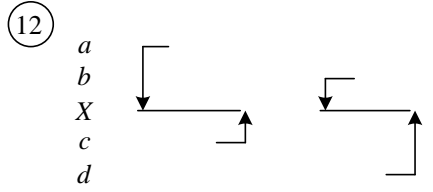
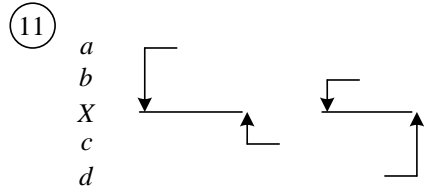
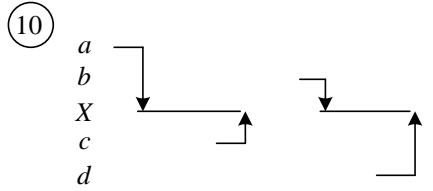
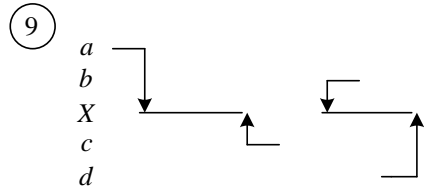
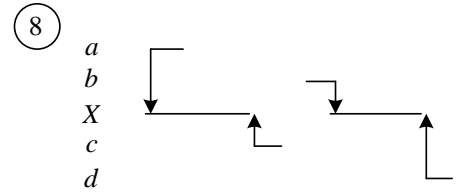
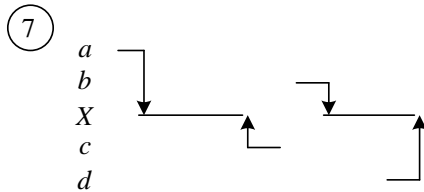
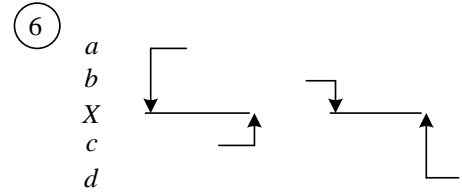
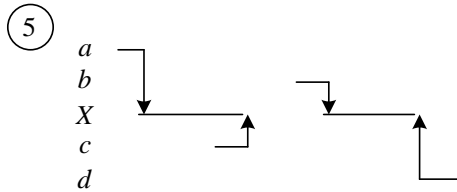
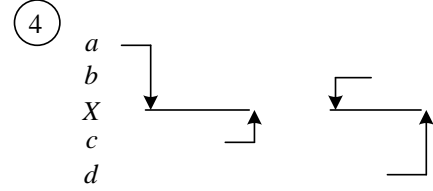
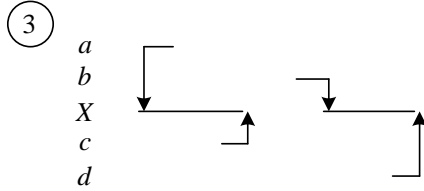
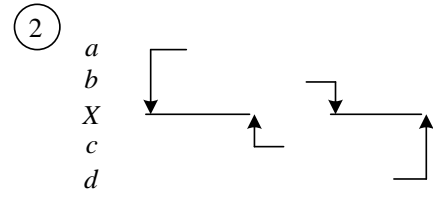
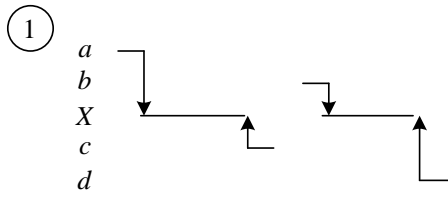
* - виконується тільки під час очного навчання

Таблиця 2.1 – Варіанти завдань до лабораторної роботи №2

Бригада 1	Номери завдань
1	1; 20; 38; 57;
2	2; 23; 46; 58;
3	3; 30; 42; 59;
4	4; 26; 34; 60;
5	5; 18; 35; 50;
6	6; 25; 34; 61;
Бригада 2	Номери завдань
1	5; 17; 37; 61;
2	6; 21; 45; 62;
3	7; 29; 41; 63;
4	8; 25; 33; 64;
5	9; 18; 35; 50;
6	10; 25; 34; 59;
Бригада 3	Номери завдань
1	9; 19; 40; 49;
2	10; 31; 48; 50;
3	11; 22; 44; 51;
4	12; 27; 36; 52;
5	5; 18; 35; 50;
6	6; 25; 33; 61;
Бригада 4	Номери завдань
1	13; 32; 39; 53;
2	14; 24; 47; 54;
3	15; 18; 43; 55;
4	16; 28; 35; 56;
5	5; 19; 36; 50;
6	6; 25; 37; 61;
Бригада 5	Номери завдань
1	4; 21; 39; 56;
2	5; 22; 41; 57;
3	6; 32; 43; 58;
4	8; 24; 35; 59;
5	9; 17; 38; 60;
6	10; 23; 35; 61;
Бригада 6	Номери завдань
1	12; 19; 37; 52;
2	5; 25; 45; 53;
3	6; 28; 41; 54;
4	1; 29; 33; 55;
5	2; 26; 35; 56;
6	13; 24; 34; 57;

Умови роботи схем

Умови роботи схем подано циклограммами



Виконати синтез багатоактної схеми, яка має 4 вхідних сигнали a , b , c , d та 7 вихідних які представляють собою сегменти семисегментного світлодіодного індикатора (див. рис. 2.5):

17. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 1,2,3,4. Кнопка «Викл» вимикає схему коли активна цифра 4.

18. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 1,3,5,7. Кнопка «Викл» вимикає схему коли активна цифра 7.

19. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 2,4,6,8. Кнопка «Викл» вимикає схему коли активна цифра 8.

20. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 2,3,4,5. Кнопка «Викл» вимикає схему коли активна цифра 5.

21. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 3,4,5,6. Кнопка «Викл» вимикає схему коли активна цифра 6.

22. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 4,5,6,7. Кнопка «Викл» вимикає схему коли активна цифра 7.

23. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 5,6,7,8. Кнопка «Викл» вимикає схему коли активна цифра 8.

24. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 6,7,8,9. Кнопка «Викл» вимикає схему коли активна цифра 9.

25. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 7,8,9,0. Кнопка «Викл» вимикає схему коли активна цифра 0.

26. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 0,1,2,3. Кнопка «Викл» вимикає схему коли активна цифра 3.

27. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 1,2,4,8. Кнопка «Викл» вимикає схему коли активна цифра 1 або 8.

28. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 5,6,7,8. Кнопка «Викл» вимикає схему коли активна цифра 5 або 8.

29. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 1,3,5,7. Кнопка «Викл» вимикає схему коли активна цифра 1 або 7.

30. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 1,3,7,9. Кнопка «Викл» вимикає схему коли активна цифра 1 або 9.

31. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками

«Більше», «Менше» реалізується послідовне чергування цифр 0,3,6,9. Кнопка «Викл» вимикає схему коли активна цифра 0 або 9.

32. Схема має 4 кнопки «Вкл.», «Викл.», «Більше», «Менше». Схема починає роботу після надходження сигналу кнопки «Вкл.». Далі кнопками «Більше», «Менше» реалізується послідовне чергування цифр 1,4,7,0. Кнопка «Викл» вимикає схему коли активна цифра 1 або 0.

Умови роботи схеми задано логічною формулою

$$33. f = \bar{a}\bar{b} + a\bar{c} + \bar{a}d$$

$$34. f = ab + bd + \bar{b}c$$

$$35. f = \bar{a}\bar{b} + \bar{c}\bar{d} + bc$$

$$36. f = \bar{c}d + a\bar{d} + b\bar{d}$$

$$37. f = \bar{c}\bar{b} + ab + ad$$

$$38. f = \bar{a}\bar{c} + bc + ad$$

$$39. f = cd + \bar{c}\bar{d} + a\bar{b}$$

$$40. f = \bar{c}d + bc + ac$$

$$41. f = c\bar{b} + ad + \bar{c}\bar{d}$$

$$42. f = (\bar{a} + c)(\bar{b} + \bar{d})(a + b)$$

$$43. f = (\bar{a} + \bar{c})(b + d)(a + b)$$

$$44. f = (\bar{b} + d)(\bar{a} + \bar{c})(c + d)$$

$$45. f = (\bar{b} + \bar{d})(a + \bar{c})(b + c)$$

$$46. f = (\bar{b} + \bar{c})(a + \bar{c})(a + d)$$

$$47. f = (c + d)(c + \bar{a})(\bar{d} + \bar{b})$$

$$48. f = (a + d)(\bar{a} + \bar{c})(d + \bar{b})$$

Багатотактний пристрій має два вхідних сигнали (a,b) та два вихідних (f_1, f_2) і відпрацьовує таку послідовність змінювання вхідних і вихідних сигналів:

49. ab 00 → 10 → 11 → 10 → 00 → 10 → 11 → 10 → 00
 f_1f_2 00 → 10 → 01 → 00 → 10 → 00 → 10 → 00 → 00

50. ab 00 → 10 → 11 → 10 → 00 → 10 → 11 → 01 → 00
 f_1f_2 00 → 01 → 11 → 00 → 01 → 00 → 10 → 00 → 00

51. ab 00 → 10 → 11 → 10 → 00 → 01 → 11 → 10 → 00
 f_1f_2 11 → 10 → 01 → 00 → 10 → 01 → 10 → 00 → 11

52. ab 00 → 10 → 11 → 10 → 00 → 01 → 11 → 01 → 00
 f_1f_2 00 → 01 → 00 → 10 → 00 → 10 → 11 → 01 → 00

53. ab 00 → 10 → 11 → 01 → 00 → 10 → 11 → 10 → 00
 f_1f_2 00 → 10 → 11 → 01 → 00 → 01 → 10 → 00 → 00

54. ab 00 → 10 → 11 → 01 → 00 → 10 → 11 → 01 → 00
 f_1f_2 11 → 10 → 01 → 00 → 10 → 00 → 01 → 10 → 11

55. ab 00 → 10 → 11 → 01 → 00 → 01 → 11 → 10 → 00
 f_1f_2 00 → 01 → 00 → 10 → 11 → 01 → 00 → 10 → 00

56. ab 00 → 10 → 11 → 01 → 00 → 01 → 11 → 01 → 00
 f_1f_2 11 → 01 → 11 → 10 → 11 → 00 → 10 → 01 → 11

57. ab 00 → 01 → 11 → 10 → 00 → 10 → 11 → 10 → 00
 f_1f_2 00 → 10 → 01 → 00 → 10 → 11 → 01 → 00 → 00

58. ab 00 → 01 → 11 → 10 → 00 → 10 → 11 → 01 → 00
 f_1f_2 00 → 00 → 10 → 01 → 00 → 01 → 10 → 11 → 00

59. ab 00 → 01 → 11 → 10 → 00 → 01 → 11 → 10 → 00
 f_1f_2 00 → 11 → 01 → 00 → 10 → 00 → 01 → 10 → 00

60. ab 00 → 01 → 11 → 10 → 00 → 01 → 11 → 01 → 00
 f_1f_2 00 → 11 → 01 → 01 → 00 → 10 → 11 → 10 → 00

61. ab 00 → 01 → 11 → 01 → 00 → 10 → 11 → 10 → 00
 f_1f_2 00 → 00 → 11 → 10 → 00 → 11 → 01 → 00 → 00

62. ab 00 → 01 → 11 → 01 → 00 → 10 → 11 → 01 → 00
 f_1f_2 11 → 00 → 10 → 00 → 01 → 00 → 11 → 01 → 11

63. ab 00 → 01 → 11 → 01 → 00 → 01 → 11 → 10 → 00
 f_1f_2 11 → 10 → 00 → 10 → 11 → 10 → 00 → 01 → 11

64. ab 00 → 01 → 11 → 01 → 00 → 01 → 11 → 01 → 00
 f_1f_2 00 → 01 → 10 → 11 → 10 → 01 → 00 → 01 → 00

2.3 Опис лабораторної установки

Функціональну схему лабораторного станду, його опис та схему електричну принципову наведено у методичних вказівках до лабораторної роботи №1. Для виконання лабораторної роботи №2 використовуються аналогічні робочі місця.

2.4 Методичні вказівки до виконання лабораторної роботи

Для виконання лабораторних робіт використовуються дискретні мікросхеми серії 74НС02, 74НС04, 74НС10, 74НС11, 74НС20, 74НС32, 74НС153, RS-тригери серії CD4043В, та мультиплектори на вісім інформаційних входів серії К555КП15 та на чотири входи 74НС153.

Опис мікросхем 74НС02, 74НС04, 74НС10, 74НС11, 74НС20 та 74НС32 наведено у методичних вказівках до виконання лабораторної роботи №1.

2.4.1 Логічна мікросхема 74НС153

Мікросхема містить 2 мультиплектора кожен з яких містить 4 селекторних входи та споживає 50 мА. Живлення мікросхеми – 5 В. Розташування входів/виходів мікросхеми показано на рис. 2.3.

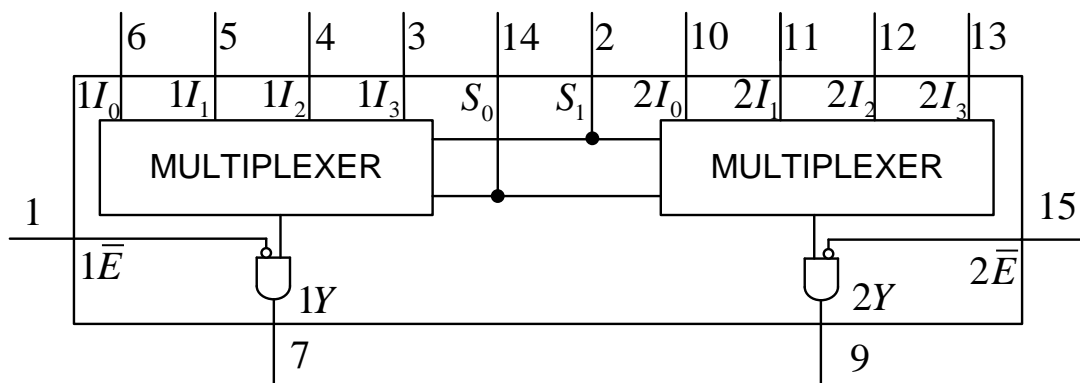


Рисунок 2.3 – Мікросхема 74НС153

На рисунку прийняті наступні позначення:

I0-I3 – інформаційні входи мультиплексорів; S0-S1 – селекторні входи; $1\bar{E}$, $2\bar{E}$ – стробуючі входи; Y1, Y2 – виходи мультиплексорів.

2.4.2 Логічна мікросхема CD4043B

Мікросхема містить 4 RS-тригери та споживає 14 мА. Живлення мікросхеми 5 В. Розташування входів/виходів мікросхеми показано на рис. 2.4.

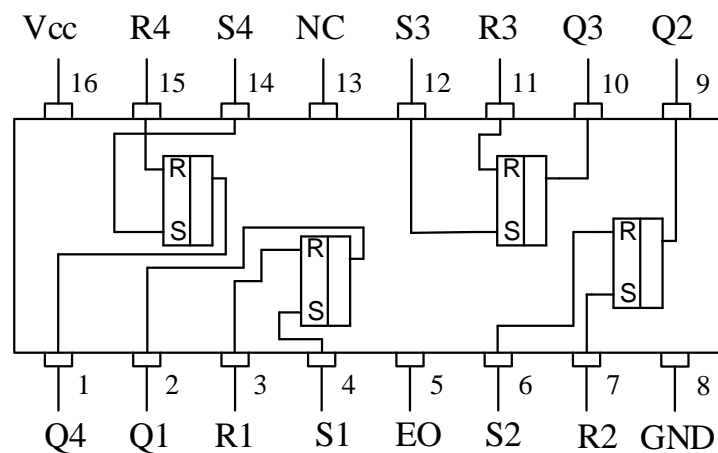


Рисунок 2.4 – Мікросхема CD4043B

На рисунку прийняті наступні позначення:

Vcc – живлення мікросхеми +5 В;

GND – «земля» живлення;

R1-R4, S1-S4 – входи даних;

Q1-Q4 – виходи тригерів;

EO – дозвіл роботи (для дозволу роботи виходів тригерів на цей вхід необхідно подати логічну одиницю).

2.4.3 Світлодіодний індикатор, виробник KINGBRIGHT

Високоєфективне джерело червоного світла зроблене з фосфідових діодів. Схема підключення виконана із загальним анодом, зображена на рис. 2.5.

- розсіювання потужності – 75 мВт;
- номінальний струм – 30 мА;
- номінальна напруга – 5В;
- робоча температура - -40...+85°С.

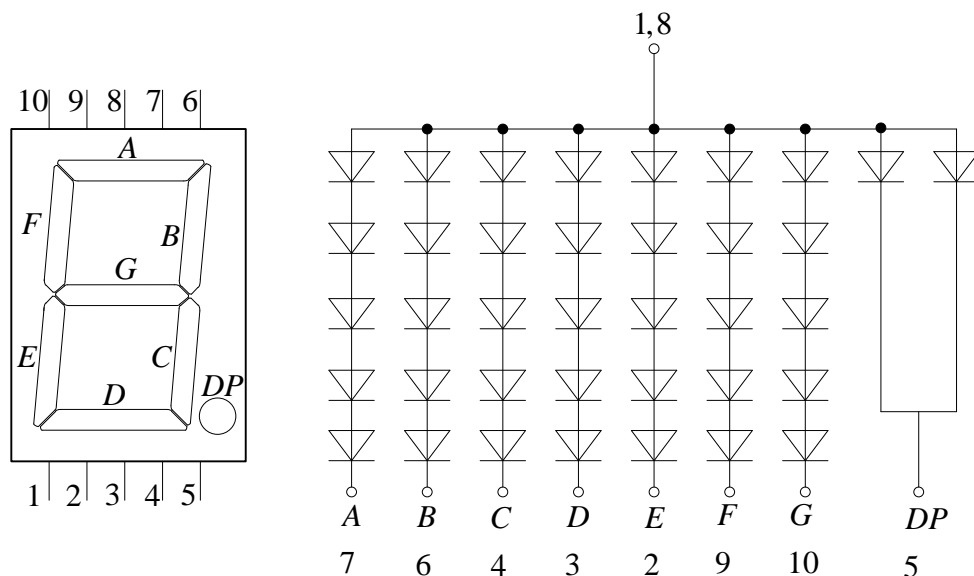


Рисунок 2.5 – Схема підключення світлодіодного індикатора KINGBRIGHT

2.4.4 Приклади синтезу та складання схем

Приклад 1. Керування двома двигунами М1, М2 здійснюється за допомогою кнопок «Пуск», «Перший двигун», «Другий двигун» і «Стоп». При натисненні кнопки «Пуск» вмикається двигун М1, потім за допомогою кнопок «Перший двигун» та «Другий двигун» можна переключати який двигун буде працювати. Кнопка «Стоп» вимикає роботу першого або другого двигуна і зупиняє роботу поки не буде натиснена кнопка «Пуск».

Вводимо позначення: M1- перший двигун, M2 – другий двигун, а – кнопка «Пуск», b – кнопка «Стоп», S1 – кнопка «Перший двигун», S2 – кнопка «Другий двигун». Графоперехід за умовами роботи схеми представлений на рисунку 2.6.

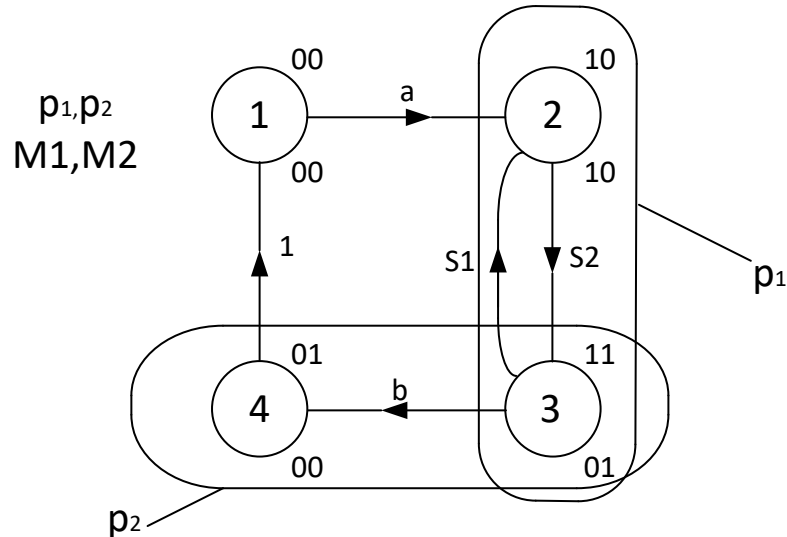


Рисунок 2.6 – Графоперехід за умовами роботи схеми

Запишемо рівняння які визначають умови вмикання тригерів:

$$S_{p1} = a \cdot \bar{p}_2;$$

$$S_{p2} = S2 \cdot p_1.$$

Умови скидання тригерів:

$$R_{p1} = b \cdot p_2 + b \cdot \bar{p}_2 = b;$$

$$R_{p2} = \bar{p}_1 + S1 \cdot p_1.$$

Формули для вихідних змінних:

$$M1 = p_1 \cdot \bar{p}_2;$$

$$M2 = p_1 \cdot p_2.$$

Імітацію ввімкнення силових контакторів керування двигунами M1 та M2 здійснюємо за допомогою світлодіодів VD1 та VD2 відповідно. Резистори R1- R4 використовуються для «підтяжки» входів мікросхем до логічного «0», коли кнопки a, b, S1 та S2 розімкнені. Резистори R5 та R6 розраховані для зниження струму, що проходить через світлодіоди VD1 та VD2, для уникнення їхнього перегорання.

На основі виразів будемо схему на дискретних логічних елементах, що зображена на рисунку 2.7.

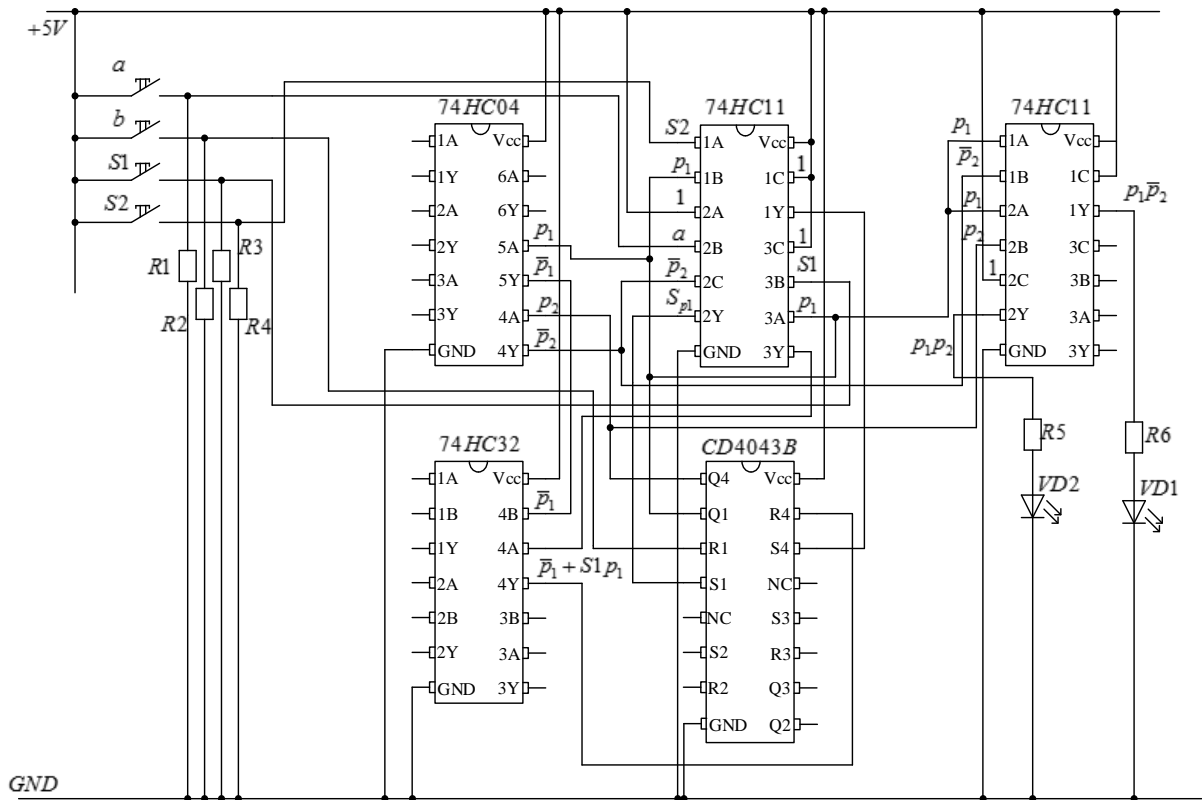


Рисунок 2.7 – Схема на дискретних логічних елементах

На рисунку 2.8 зображена схема, зібрана на стенді.

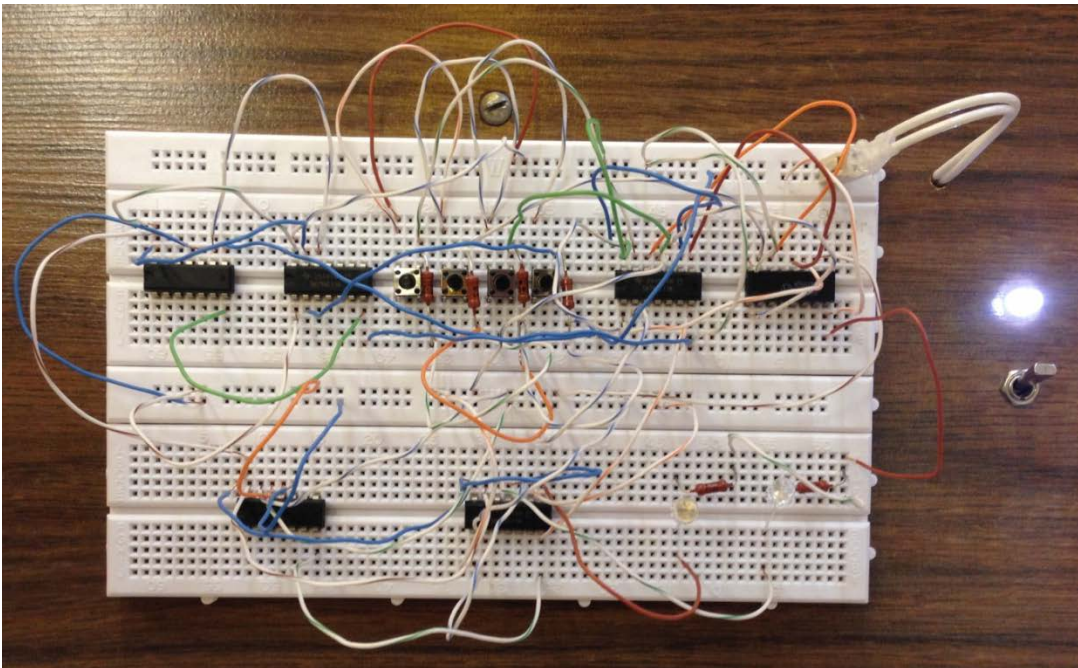


Рисунок 2.8 – Схема зібрана на стенді

Приклад 2. Розглянемо керування дозуючим пристроєм зображеним на рисунку 2.9. У вихідному положенні ємність пуста і всі крани перекриті. Кнопка «Пуск» запускає роботу пристрою на один цикл. При подачі сигналу a – кнопки «Пуск», спрацьовує датчик першого крану $K1$ та у дозуючий пристрій наливається рідина 1 поки не спрацює датчик $d1$, датчик потрібного об'єму рідини 1. Після спрацьовування датчика, $K1$ закривається і відкривається $K3$, через який рідина один зливається у ємність. Після спрацьовування датчика $d0$, який показує, що дозуючий пристрій пустий, відкривається $K2$, що наповнює пристрій рідиною 2 до рівня $d2$. Після цього рідина 2 також зливається у ємність. Якщо кнопка «Пуск» натиснена то цикл повторюється спочатку, якщо ні то повертаємося у вихідне положення.

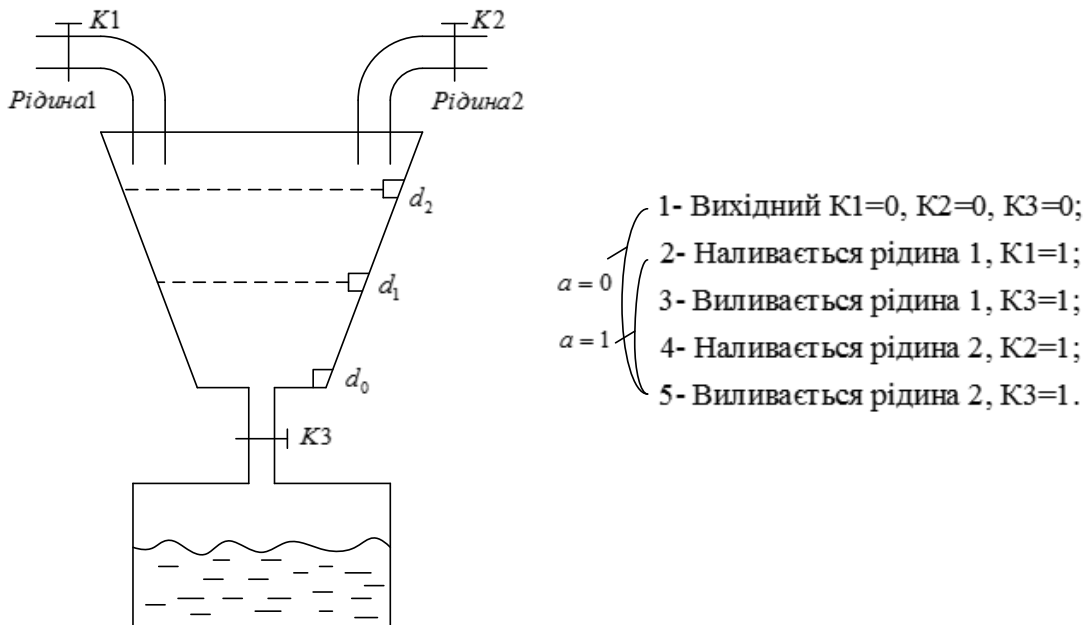


Рисунок 2.9 – Дозуючий пристрій

Графоперехід за умовою задачі зображений на рис. 2.10.

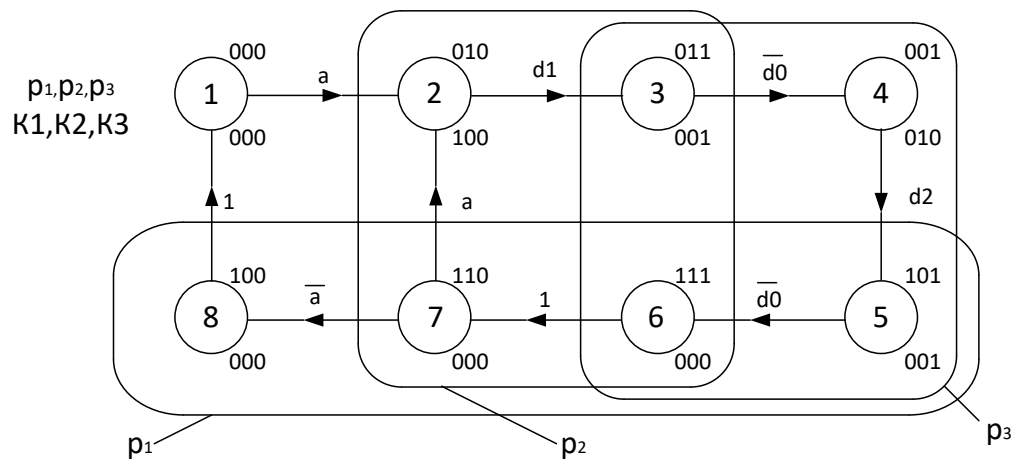


Рисунок 2.10 – Графоперехід за умовою задачі

Запишемо рівняння які визначають умови вмикання тригерів:

$$S_{p_1} = d_2 \cdot \bar{p}_2 \cdot p_3;$$

$$S_{p_2} = a \cdot \bar{p}_1 \cdot \bar{p}_3 + \bar{d}_0 \cdot p_1 \cdot p_3;$$

$$S_{p_3} = d_1 \cdot \bar{p}_1 \cdot p_2.$$

Умови скидання тригерів:

$$R_{p_1} = a \cdot p_2 \cdot \bar{p}_3 + \bar{p}_2 \cdot \bar{p}_3;$$

$$R_{p_2} = \bar{a} \cdot p_1 \cdot \bar{p}_3 + \bar{d}_0 \cdot \bar{p}_1 \cdot p_3;$$

$$R_{p_3} = p_1 \cdot p_2.$$

Формули для вихідних змінних:

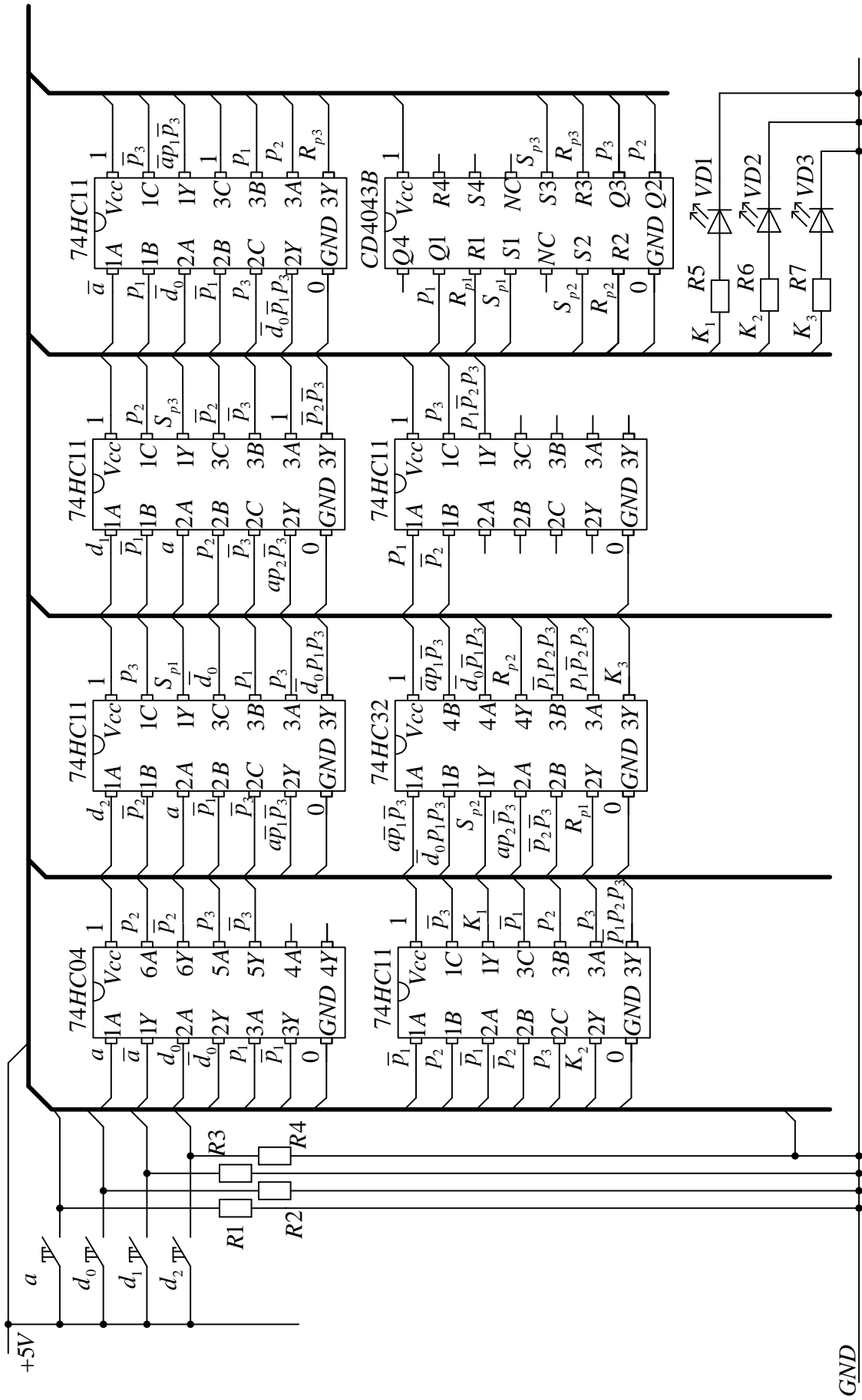
$$K_1 = \bar{p}_1 \cdot p_2 \cdot \bar{p}_3;$$

$$K_2 = \bar{p}_1 \cdot \bar{p}_2 \cdot p_3;$$

$$K_3 = \bar{p}_1 \cdot p_2 \cdot p_3 + p_1 \cdot \bar{p}_2 \cdot p_3.$$

Імітацію ввімкнення силових контакторів керування кранами K1, K2 та K3 здійснюємо за допомогою світлодіодів VD1, VD2 та VD3 відповідно. Резистори R1-R4 використовуються для «підтяжки» входів мікросхем до логічного «0», коли кнопки a , d_0 , d_1 та d_2 розімкнені. Резистори R5, R6 та R7 розраховані для зниження струму, що проходить через світлодіоди VD1, VD2 та VD3, для уникнення їхнього перегорання.

З отриманих рівнянь будемо схему яка зображена на рис. 2.11.



Приклад 3. Умови роботи схеми задано формулою $f = \bar{a}b + a\bar{c} + \bar{a}d$.

Потрібно спроектувати схему на мультиплексорі, що має 3 селекторні лінії.

При синтезі одноканальних схем для визначення матриці $[A]$ використовують карту Карно, складену за умовою роботи схеми. Карта Карно для заданої роботи схеми зображена на рис. 2.12.

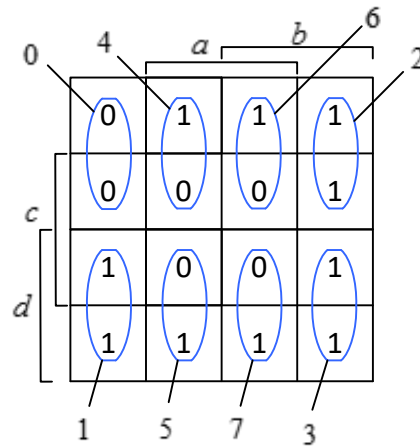


Рисунок 2.12 – Карта Карно

Як селекторні вибираємо змінні a , b , d , тоді матриця селектування має вигляд

$$[Z] = \begin{bmatrix} a \\ b \\ d \end{bmatrix} = \begin{bmatrix} 00001111 \\ 00110011 \\ 01010101 \end{bmatrix}.$$

Кожному стовпцю матриці відповідає двоклітинковий контур карти Карно. На рис.2.12 контури позначено цифрами, що дорівнюють номерам стовпців в матриці селектування. Кожний двоклітинковий контур – це карта Карно, яка визначає один з сигналів a_0, a_1, \dots, a_7 на входних лініях мультиплексора-селектора як функцію вхідного сигналу c .

Отже, матриця вхідних даних має вигляд:

$$[A] = [a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7] = [0 \ 1 \ 1 \ 1 \ \bar{c} \ \bar{c} \ \bar{c} \ \bar{c}].$$

Імітацію спрацювання вихідної функції f здійснюємо за допомогою світлодіоду VD1. Резистори R1-R4 використовуються для «підтяжки» входів мікросхем до логічного «0», коли кнопки a , b , c та d розімкнені. Резистор R5

розрахований для зниження струму, що проходить через світлодіод VD1, для уникнення його перегорання.

Схема побудована за одержаними виразами, показано на рис. 2.13.

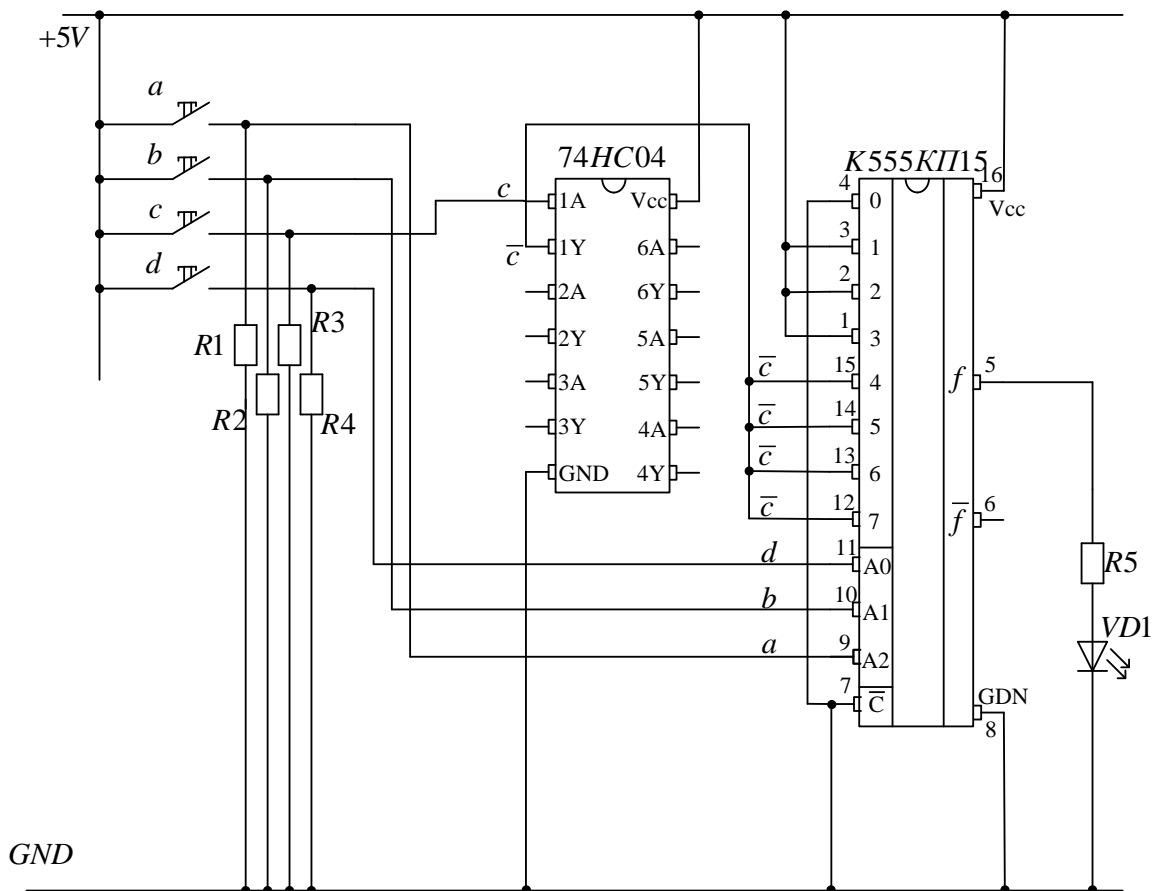


Рисунок 2.13 – Схема на мультиплексорі

На рисунку 2.14 представлена схема зібрана на стенді.

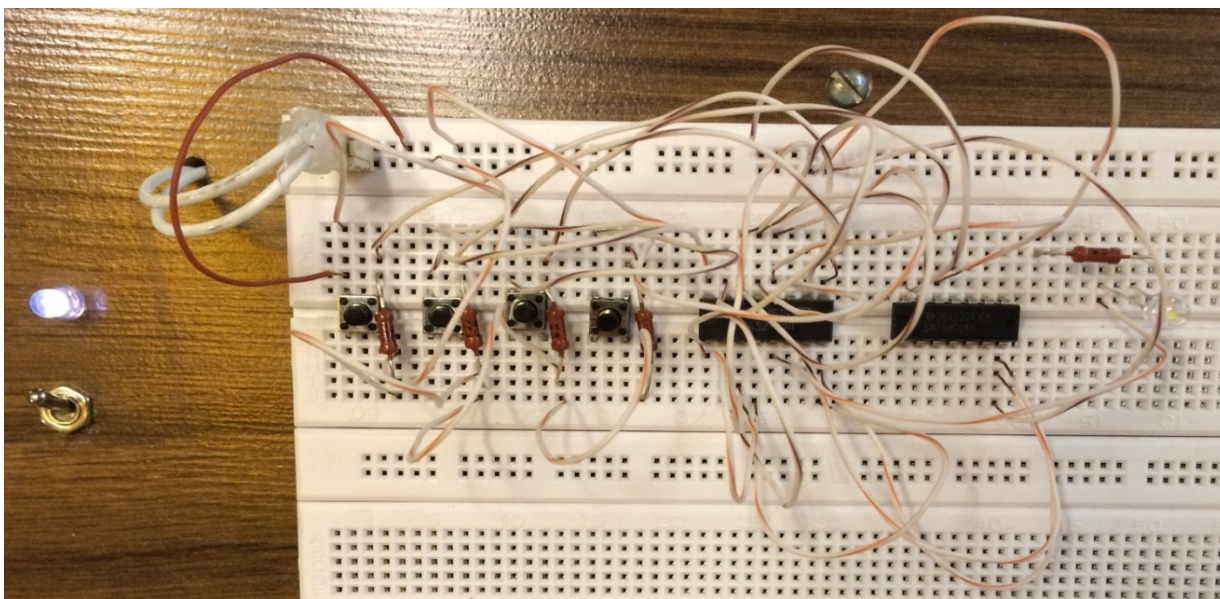


Рисунок 2.14 – Схема на мультиплексорі зібрана на стенді

Приклад 4. Умову роботи схеми задано циклограмою, зображеною на рис.

2.15. Спроекувати схему на мультиплексах.

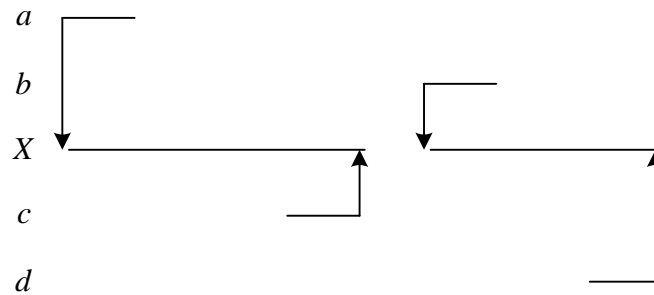


Рисунок 2.15 – Циклограма

При побудові багатотактних схем на мультиплексах-селекторах їх вихідні сигнали використовуються як проміжні змінні. Кожний стан схеми визначається власною комбінацією цих змінних. Вихідні змінні схеми в цьому випадку є комбінаційними функціями проміжних змінних.

Оскільки, вихідні змінні мультиплексорів подаються на селекторні входи то матриця селектування має вигляд:

$$[Z] = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 00001111 \\ 00110011 \\ 01010101 \end{bmatrix}.$$

Для визначення матриці $[A]$ потрібно побудувати графоперехід за умовою роботи схеми, який зображено на рис. 2.16.

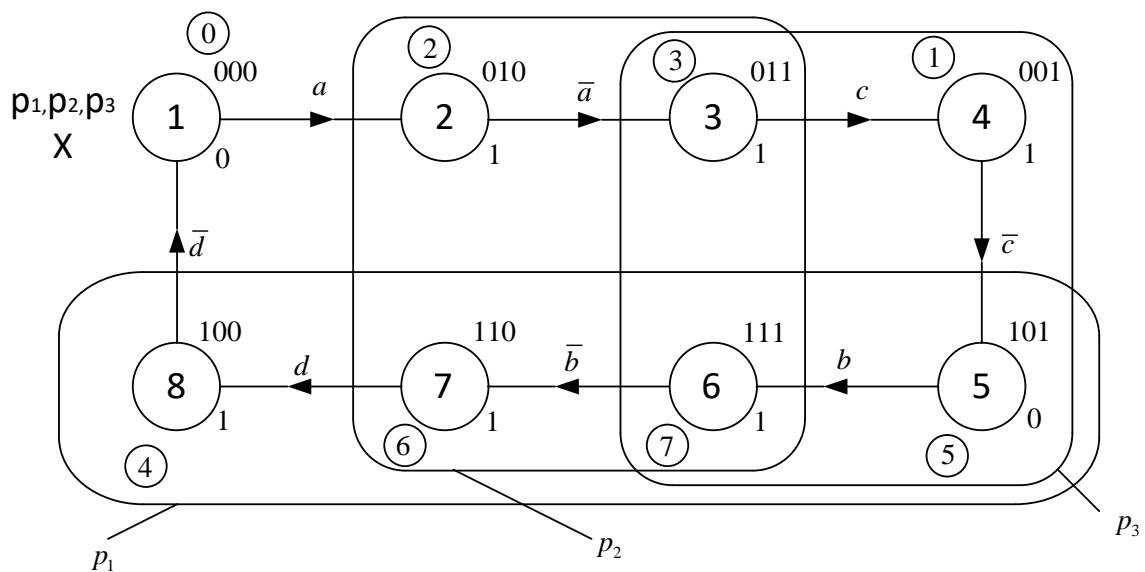


Рисунок 2.16 – Графоперехід за умовою задачі

Матриця вхідних даних $[A]$ складається таким чином:

1. Для кожного стовпця матриці вказується відповідний йому стан на графі переходів. Нульовому стовпцю відповідає стан з набором проміжних змінних 000, першому стовпцю – 001, другому – 010 і т.д.

2. Перевіряється, які проміжні змінні не змінюють своє значення при переході з даного стану в наступний. Якщо будь-яка проміжна змінна не змінює свого значення, то це значення стає елементом стовпця матриці $[A]$.

3. Якщо значення проміжної змінної змінюється з 0 на 1 під час переходу з даного стану в наступний, то елементом стовпця стає значення вхідної змінної, що спричинює цей перехід. Якщо ж проміжна змінна змінюється з 1 на 0, то елементом стовпця буде інверсія значення вхідного сигналу.

Матриця $[A]$, складена за цими правилами для графопереходу на рис. 2.16, має вигляд:

$$[A] = \begin{bmatrix} 0 & \bar{c} & 0 & 0 & d & 1 & 1 & 1 \\ a & 0 & 1 & \bar{c} & 0 & b & \bar{d} & 1 \\ 0 & 1 & \bar{a} & 1 & 0 & 1 & 0 & b \end{bmatrix}$$

Вихідні сигнали схеми визначаються так само, як і при синтезі схем на RS -тригерах

$$X = p_2 + \bar{p}_1 p_3 + p_1 \bar{p}_3$$

Імітацію ввімкнення вихідного сигналу X здійснюємо за допомогою світлодіоду VD1. Резистори R1-R4 використовуються для «підтяжки» входів мікросхем до логічного «0», коли кнопки a , b , c та d розімкнені. Резистор R5 розрахований для зниження струму, що проходить через світлодіод VD1, для уникнення його перегорання.

Схема побудована за даними формулами зображена на рис. 2.17.

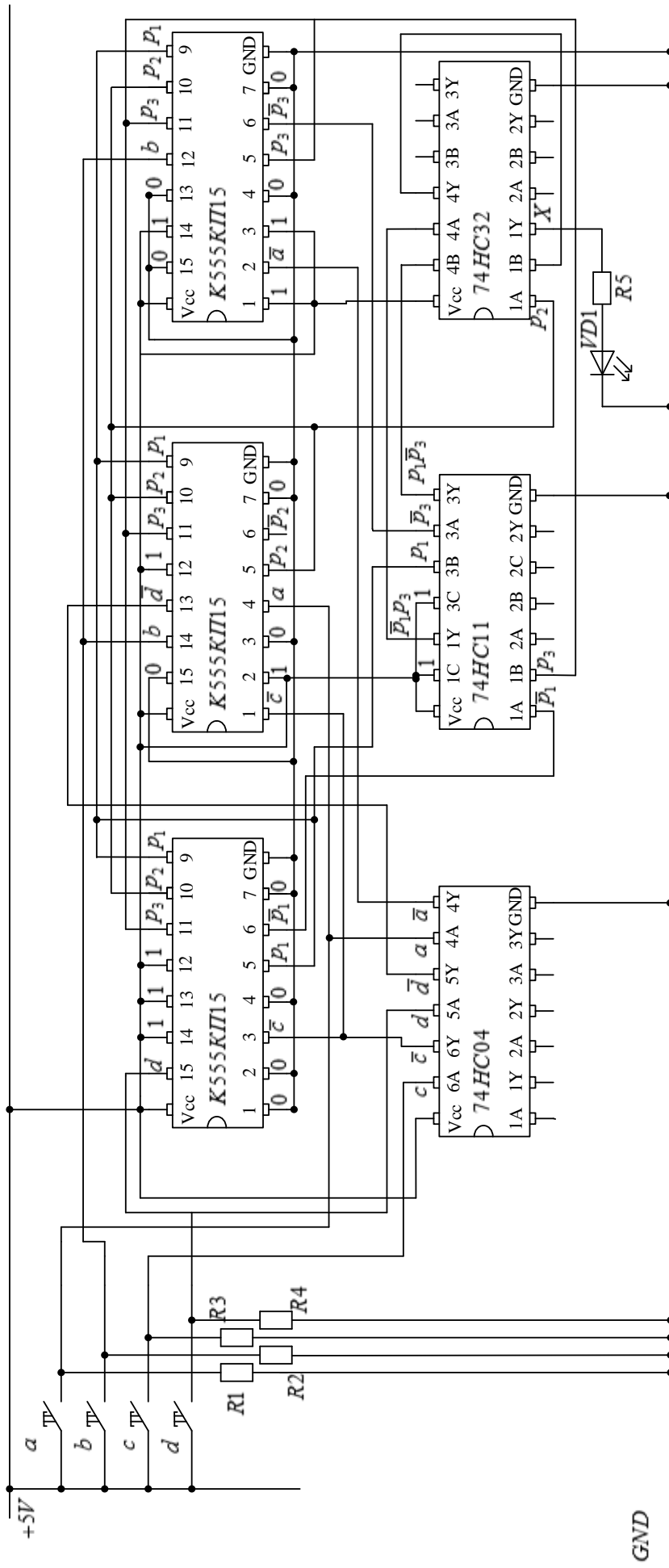


Рисунок 2.17 - Схема на мультиплексорах селектора для прикладу 4

Процедуру створення проєкту схеми в середовищі Multisim з використанням двох RS-тригерів можна подивитися у відео-уроці до лабораторної роботи №2 за посиланням <https://youtu.be/87-ItZRa7Dw>.

Процедуру створення проєкту схеми в середовищі Multisim з використанням трьох RS-тригерів можна подивитися у відео-уроці до лабораторної роботи №2 за посиланням <https://youtu.be/w7wiBIMHhDM>.

Приклад синтезу та створення проєкту схеми в середовищі Multisim з для завдання 2 можна подивитися у відео-уроці до Лекції 9 за посиланням <https://youtu.be/qP8QYBJgMXY>.

2.5 Зміст звіту

Звіт має містити такі матеріали дослідження кожної схеми:

- 1) титульний аркуш із зазначенням назви дисципліни та лабораторної роботи (див. Додаток А);
- 2) мету роботи, програму роботи та умови роботи схем;
- 3) логічний синтез схеми;
- 4) принципові схеми на наявних у даній лабораторній роботі логічних елементах, з підписами викладача;
- 5) перелік елементів до кожної схеми;
- 6) схеми у програмному забезпеченні Multisim з результатами її симуляції;
- 7) висновки про відповідність роботи схеми заданим умовам.

2.6 Контрольні запитання

1. Як виконується синтез одноканальних схем на мультиплекторах-селекторах?
2. Як складається матриця вхідних сигналів мультиплектора при синтезі одноканальних схем?
3. Як вибираються селекторні змінні мультиплектора при синтезі одноканальних схем?

4. Як складається матриця вхідних сигналів мультиплексора при синтезі багатотактних схем?
5. Як виконується синтез схем на тригерах?
6. Як складаються рівняння для вихідних змінних на основі графопереходу?
7. Як визначається кількість тригерів та мультиплексорів при синтезі багатотактних схем?

Лабораторна робота № 3

ДОСЛІДЖЕННЯ РЕЛЕЙНО-КОНТАКТНИХ СХЕМ АВТОМАТИКИ

Тривалість лабораторної роботи – 4 год.

Тривалість домашньої роботи – 4 год.

Мета роботи – практично перевірити різні методи синтезу одноконтурних та багатоактних схем, навчитися будувати релейно-контакторні схеми на основі синтезованих логічних рівнянь, набути вміння працювати із промисловими реле, налаштовувати реле часу, складати схеми з їх використанням та практично перевіряти роботу таких схем

3.1. Основні теоретичні відомості

Основні методи синтезу релейно-контактних схем описано в основних теоретичних відомостях до лабораторної роботи №1-2 “Логічний синтез та дослідження схем автоматики на дискретних логічних елементах”.

3.1.1 Синтез схем з технологічними затримками

У деяких системах промислової автоматики потрібно забезпечувати певну затримку часу між операціями, що виконуються робочою машиною. У схемах таких систем застосовуються елементи часу, які затримують сигнали елементів, що реалізують проміжні функції.

З’ясуємо особливості синтезу схем із спеціальними технологічними затримками. На першому етапі синтезу – формулюванні умов роботи необхідно зазначити, які операції супроводжуються затримками і вказати величини затримок. Ці затримки потрібно записати в таблицю переходів.

Основна особливість процесу синтезу з урахуванням технологічних затримок виявляється під час вибору кількості проміжних змінних і розміщення їх станів. Кожному переходу схеми з одного стану в інший з незалежною затримкою має відповідати змінювання значення однієї проміжної змінної з 1 на 0 або з 0 на 1.

Отже, якщо затримки часу в разі змінювання аргументу в обох напрямках можна встановлювати незалежно, то за допомогою n проміжних змінних можна забезпечити $2n$ незалежних затримок. Якщо затримка забезпечується змінюванням аргументу тільки в одному напрямку, то кількість незалежних затримок не може перевищувати n .

Отже, потрібну кількість проміжних змінних визначають умовою:

у першому випадку

$$n \geq \frac{t}{2}, \quad (3.1)$$

у другому:

$$n \geq t, \quad (3.2)$$

де t – кількість незалежних затримок часу.

Більшість реальних елементів часу, що застосовуються в схемах промислової автоматики, забезпечують регульовану затримку часу тільки в разі змінювання сигналу на їх вході з 0 на 1. Сигнал на виході елемента часу зникає одночасно зі зникненням сигналу на його вході. Такі ж самі функції виконують програмно реалізовані таймери логічних програмованих контролерів. Тому кількість проміжних змінних будемо визначати за формулою (3.2).

Друга особливість синтезу схем із затримками – у таблиці переходів з'являються рядки, що не мають стійких станів. Схема перебуває в таких нестійких станах протягом технологічної затримки.

Розглянемо приклади синтезу схем з технологічними затримками. Човниковий транспортер – це вантажна платформа I , що переміщується за допомогою тягового троса між одним пунктом завантаження A і двома пунктами розвантаження B , C (рис. 3.1). Під час прибуття в ці пункти вантажна платформа замикає кінцеві вимикачі 2 , 3 , 4 (сигнали a , b , c).

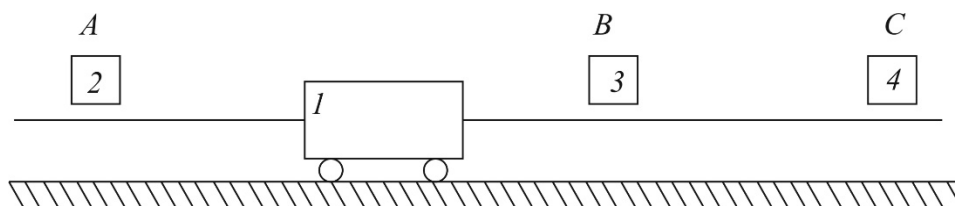


Рисунок 3.1 – Човниковий транспортер

Необхідно виконати синтез схеми керування, що забезпечує такий цикл переміщення платформи: стояння в пункті завантаження A протягом часу t_1 , переміщення у перший пункт розвантаження B , стояння в пункті B протягом часу t_2 , переміщення в другий пункт розвантаження C , стояння в пункті C протягом часу t_3 , переміщення в пункт завантаження A без проміжної зупинки в пункті B . Після цього цикл роботи повторюється.

Як виконавчі елементи вибираємо магнітні пускачі двигунів човникового транспортера «праворуч» і «ліворуч». Відповідні вихідні сигнали позначимо через X і Y ($X = 1$ і $Y = 1$, коли пускачі ввімкнено).

Виходячи з умов роботи схеми, робимо висновок, що вона може перебувати у таких станах: 1 – стояння в пункті завантаження A ; 2 – рух від пункту A до пункту B ; 3 – стояння в пункті B ; 4 – рух від пункту B до пункту C ; 5 – стояння в пункті C ; 6 – рух від пункту C до пункту A .

Первинну таблицю переходів показано на рис. 3.2, *a*. Усі комбінації, в яких більш ніж одна вхідна змінна має значення 1, неможливі через те, що кінцеві вимикачі замикаються вантажною платформою по чергово. У відповідних стовпцях таблиці ставимо риси.

Стани 1, 3, 5 не можуть бути стійкими, оскільки без змінювання будь-яких зовнішніх сигналів вони переходять у стани 2, 4 і 6 із затримками часу t_1 , t_2 , t_3 відповідно. Перехід до стану 2 відбувається, якщо $a = 1$, $b = 0$, $c = 0$. Стійкий стан 2 зберігається, якщо $a = 0$, $b = 0$, $c = 0$, коли транспортер починає рухатися і сигнал кінцевого вимикача стає рівним нулю. Перехід від стану 2 до стану 3 відбувається, якщо $a = 0$, $b = 1$, $c = 0$. Проте в третьому рядку записується не стійкий стан <3>, а нестійкий стан 4 через те, що за цього набору вхідних сигналів після закінчення затримки часу t_2 без будь-якої зміни зовнішніх сигналів схема переходить у стійкий стан <4>, записаний у четвертому рядку. Цей стан зберігається, якщо $a = 0$, $b = 0$, $c = 0$, і переходить у стан 5 у разі досягнення платформою пункту розвантаження C (сигнал $c = 1$). Нестійкий стан 5 через час t_3 переходить у стійкий стан <6> (повернення платформи в пункт A), який зберігається, якщо $a = 0$, $b = 0$, $c = 0$ і $a = 0$, $b = 1$, c

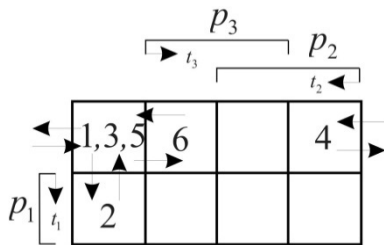
= 0. Якщо $a = 1, b = 0, c = 0$, цей стан переходить в стан 1 і цикл роботи повторюється.

Номер вихідного стану	Затримки	Наступні стани								Вихідні змінні	
										X	Y
1	t_1	-	2_{t_1}	-	-	-	-	-	-	0	0
2	-	<2>	<2>	-	3	-	-	-	-	1	0
3	t_2	-	-	-	4_{t_2}	-	-	-	-	0	0
4	-	<4>	-	-	<4>	-	-	-	5	1	0
5	t_3	-	-	-	-	-	-	-	6_{t_3}	0	0
6	-	<6>	1	-	<6>	-	-	-	<6>	0	1

a

Номер вихідного стану	Наступні стани							
1, 3, 5	-	2_{t_1}	-	4_{t_2}	-	-	-	6_{t_3}
2	<2>	<2>	-	3	-	-	-	-
4	<4>	-	-	<4>	-	-	-	5
6	<6>	1	-	<6>	-	-	-	<6>

б



в

	P_1	P_2	P_3	X	Y
1, 3, 5	0	0	0	0	0
2	1	0	0	1	0
4	0	1	0	1	0
6	0	0	1	0	1

Рисунок 3.2 – Синтез схеми керування човниковим транспортером: *a* – первинна таблиця переходів; *б* – стиснена таблиця переходів; *в* – карта відповідності та таблиця відповідності

Кількість рядків первинної таблиці переходів (рис. 3.2, а) можна зменшити, об'єднавши перший, третій і п'ятий рядки. Стиснену таблицю переходів показано на рис. 3.2, б.

Розглянемо питання про вибір кількості і розміщення станів проміжних змінних. Будемо вважати, що елемент часу зумовлює затримку тільки в разі змінювання вхідного сигналу від 0 до 1. Тому згідно з виразом (3.2) кількість проміжних змінних $n = 3 (P_1, P_2, P_3)$.

Карту відповідності показано на рис. 3.2, в. Стрілками позначено напрямки змінювання проміжних змінних, що супроводжуються затримками t_1, t_2, t_3 .

Таблицю відповідності, складену за картою відповідності, показано на рис. 3.2, г. У ній, крім проміжних змінних, указано також значення вихідних змінних X і Y .

За картами визначено такі алгебричні вирази

$$P_1 = \bar{p}_2 \bar{p}_3 \bar{b} \bar{c}; \quad (3.3)$$

$$P_2 = \bar{p}_1 \bar{p}_3 \bar{a} \bar{c}; \quad (3.4)$$

$$P_3 = p_3 \bar{a} + \bar{p}_2 \bar{c}; \quad (3.5)$$

$$X = p_1 + p_2; \quad (3.6)$$

$$Y = p_3. \quad (3.7)$$

Вирази (3.3), (3.4), (3.5) заздалегідь перетворено до вигляду

$$P_1 = \overline{p_2 + p_3 + b + c};$$

$$P_2 = \overline{p_1 + p_3 + a + c};$$

$$P_3 = \overline{\overline{p_3 \cdot a \cdot p_2 c}}.$$

Крім того, враховано, що таймери затримують сигнали проміжних змінних P_1, P_2, P_3 , а сигнали p_1, p_2, p_3 – це вихідні сигнали таймерів.

У розглянутому прикладі вдалось обійтись мінімально необхідною кількістю проміжних змінних, що дорівнює кількості затримок часу. В таблиці переходів з'явилися додаткові рядки, що відповідають нестійким станам, зумовленим затримками. Крім того, вихідні змінні було визначено як

комбінаційні функції проміжних змінних. Це можливо тільки в тому разі, коли під час синтезу первинна таблиця переходів не стискалася або під час стиснення об'єднувалися рядки з однаковими значеннями вихідних змінних.

3.1.2 Синтез схем з технологічними затримками методом циклограм

Під час синтезу схем з технологічними затримками на циклограмі позначаються величини затримок часу і лінії дії елементів часу (таймерів). Нехай, наприклад, елемент X треба ввімкнути із затримкою часу t після надходження сигналу a і вимкнути без затримки часу в разі його зникнення. Час дії сигналу a перевищує величину затримки t .

Циклограму, що відповідає цим умовам роботи, зображено на рис. 3.3.

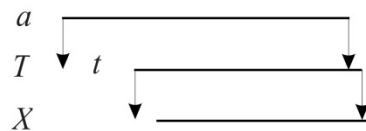


Рисунок 3.3 – Циклограма, що відображає вмикання елемента X із затримкою часу t

Виконаємо синтез схеми.

Умови спрацьовування елемента X

$$f'_X = t,$$

умови неспрацьовування

$$\bar{f}''_X = \bar{t} = t.$$

Перша та друга перевірки задовольняються, тому

$$X = tt = t.$$

Третя перевірка також задовольняється і вираз для X залишається незмінним.

Для елемента часу T умови спрацьовування та неспрацьовування такі:

$$f'_T = a; \quad \bar{f}''_T = a$$

і

$$T = a.$$

Отже, в результаті синтезу отримали найпростіші вирази для елементів X і T . Це стало можливим тому, що ці елементи вмикаються тими самими сигналами, що й вимикаються.

Якщо тривалість сигналу, що дає команду починати відлік затримання часу, менша за величину затримки часу, то цей сигнал не можна безпосередньо застосовувати для керування таймером. У цьому разі необхідно ввести проміжний елемент, змінювання стану якого дає команду починати відлік затримання часу. Надалі стан цього елемента не повинен змінюватися принаймні до кінця затримки часу.

Логічні формули для таймерів визначають за тими ж самими правилами, що й для вихідних та проміжних елементів. Розглянемо приклад. Керування трьома двигунами $M1$, $M2$, $M3$ виконують за допомогою кнопок «Пуск» і «Стоп». Натисненням кнопки «Пуск» вмикають двигун $M1$ без затримки часу, потім через час t_1 – двигун $M2$ і після цього через час t_2 – двигун $M3$. Натисненням кнопки «Стоп» двигун $M3$ вимикають без затримки часу, потім через час t_3 – двигун $M2$ і після цього через час t_4 – двигун $M1$. Виконати синтез схеми керування двигунами $M1$, $M2$, $M3$.

Уведемо такі позначення: a – сигнал кнопки «Пуск»; b – сигнал кнопки «Стоп»; X_1 , X_2 , X_3 – сигнали керування двигунами $M1$, $M2$, $M3$ відповідно; T_1 , T_2 , T_3 , T_4 – таймери, що забезпечують затримки часу t_1 , $t_1 + t_2$, t_3 , $t_3 + t_4$.

Циклограму, що відповідає заданим умовам роботи схеми, зображено на рис. 3.4.

Час натиснення кнопки «Пуск» менший за затримку часу t_1 , але вводити проміжну змінну немає потреби. Замість неї можна використати сигнал X_1 , який набуває значення 1 без затримки після натиснення кнопки «Пуск» і не змінюється протягом часу, що перевищує t_1 . Сигнал X_1 подає команду на початок відліку часу таймером T_1 . Цей самий сигнал подає команду на початок відліку часу також таймером T_2 , тому затримка часу цього таймера становить не t_2 , а $t_1 + t_2$.

Час натиснення кнопки «Стоп» (сигнал b) менший за затримку часу t_3 , тому вводимо проміжну змінну P , яка набуває значення 1 без затримки і подає команду на початок відліку часу t_3 таймером T_3 та часу $t_3 + t_4$ таймером T_4 .

Сигнал b подає команду на вимикання X_3 , сигнал таймера $T_3 - X_4$ і таймера $T_4 - X_1$. Таймери T_1 і T_2 вмикаються сигналом X_1 , а таймери T_3 і T_4 – сигналом P , тобто такими самими сигналами, якими вони вмикалися. Проміжна змінна P вмикається сигналом X_1 .

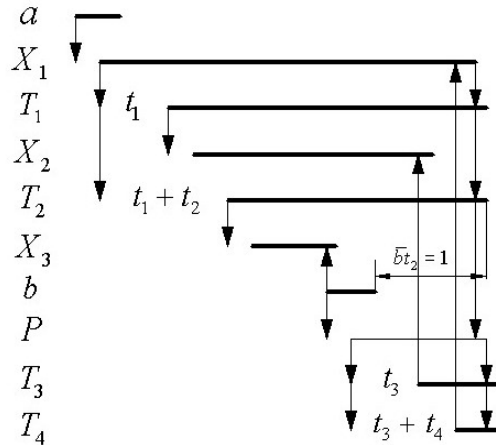


Рисунок 3.4 – Циклограма до прикладу

За циклограмою визначаємо такі логічні формули:

$$\begin{aligned} X_1 &= (a + x_1)\bar{t}_4; & X_2 &= t_1\bar{t}_3; & X_3 &= t_2\bar{b}\bar{p}; & T_1 &= x_1; & T_2 &= x_1; \\ T_3 &= p; & T_4 &= p; & P &= (b + p)x_1. \end{aligned}$$

Під час визначення виразу для X_3 враховано, що не виконується третя перевірка. Дійсно, умови спрацьовування $f'_{x_3} = t_2$, умови неспрацьовування $\bar{f}''_{x_1} = \bar{b}$. Перша та друга перевірки виконуються. Проте вираз $t_2\bar{b}$ набуває значення 1 у частині вимикального періоду елемента X_3 , показаний на рис. 3.4. Уводити нову проміжну змінну немає потреби. Можна використати змінну P , яка дорівнює нулю у вмикальному періоді X_3 і дорівнює одиниці, якщо $t_2\bar{b} = 1$ у вимикальному періоді. Тому остаточно вираз X_3 набуває вигляду $t_2\bar{b}\bar{p}$.

За логічними виразами, що описують роботу схеми, на рис. 3.5 зображено її релейно-контактний варіант.

Отримані логічні вирази дають змогу побудувати схему на будь-якій елементній базі або скласти програму для логічного програмованого контролера. Водночас циклограму дуже зручно використовувати для аналізу роботи релейно-контактної схеми, тому що на циклограмі чітко визначається

послідовність спрацьовування елементів схеми та вказується, під дією яких сигналів ці спрацьовування відбуваються.

У схемі на рис. 3.5 після натиснення кнопки «Пуск» (сигнал а) без затримки часу вмикається контактор X_1 і залишається ввімкненим після відпускання цієї кнопки завдяки самоблокуванню. Контакт x_1 вмикає таймери T_1 і T_2 . Із затримкою часу замикаються контакти t_1 і t_2 та вмикаються відповідно контактори X_2 і X_3 .

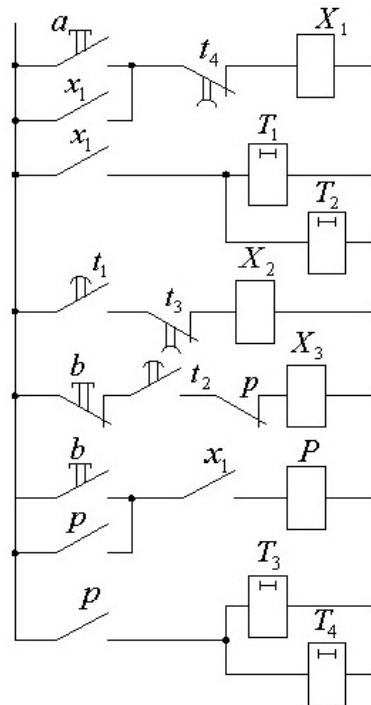


Рисунок 3.5 – Релейно-контактна схема до прикладу

Після натиснення кнопки «Стоп» (сигнал b) без затримки часу вимикається контактор X_3 та вмикається реле P . Після відпускання кнопки «Стоп» контактор X_3 залишається вимкнутим, оскільки контакт p у колі його котушки буде розімкненим (у цьому сенс виконання третьої перевірки реалізованості циклограми). Контакт p проміжного реле P вмикає таймери T_3 і T_4 . Із затримкою t_3 вимикається контактор X_2 , а потім із затримкою t_4 – контактор X_1 . Контакт x_1 контактора X_1 вмикає таймери T_1 і T_2 , а також проміжне реле P , яке вмикає таймери T_2 і T_4 .

3.2 Програма роботи

За варіантом, наведеним у таблиці 3.1, виконати наступні завдання:

1. Виконати логічний синтез схеми керування та отримати логічні рівняння (**виконується вдома під час СРС**).
2. Мінімізувати кількість змінних у рівняннях (**виконується вдома під час СРС**).
3. За мінімізованими алгебричними рівняннями у пункті 2 спроектувати релейно-контактну схему (**виконується вдома під час СРС**).
4. Зібрати схему на стенді та перевірити її роботу*.
5. Показати зібрану схему викладачу та продемонструвати її роботу (**викладач підтверджує своїм підписом у протоколі працездатність схеми**)*.
6. Зібрати схему у програмному середовищі ECTS та виконати симуляцію її роботи (**виконується безпосередньо під час лабораторних робіт**);
7. Оформити звіт по лабораторній роботі та зробити висновки.

* - виконується тільки під час очного навчання

Таблиця 3.1 – Варіанти завдань

Бригада	Номери завдань
1	1, 7, 9, 16
2	2, 8, 10, 15
3	3, 5, 11, 14
4	4, 6, 12, 13
5	1, 8, 11, 14
6	2, 5, 9, 15

Умови роботи схем

1. Умови роботи схеми задано наступними функціями:

$$f_1 = \bar{a}\bar{b}c + d\bar{b}$$

$$f_2 = d\bar{c} + \bar{a}\bar{b}c$$

2. Умови роботи схеми задано наступними функціями:

$$f_1 = cd + da\bar{b}$$

$$f_2 = cba + \bar{a}\bar{b}c$$

3. Умови роботи схеми задано наступними функціями:

$$f_1 = \bar{a}\bar{c}d + cb$$

$$f_2 = ac\bar{d} + d\bar{a}$$

4. Умови роботи схеми задано наступними функціями:

$$f_1 = \bar{a}d\bar{b} + c\bar{a}$$

$$f_2 = c\bar{d}b + ba$$

5. Схема має три вхідні a , b , c і один вихідний X сигнали. Сигнал X набуває значення 1 після надходження сигналів a або b і зберігає це значення після зняття цих сигналів. Сигнал X вимикається ($X = 0$) після надходження сигналу c незалежно від стану сигналів a і b .

6. Схема має чотири вхідні a , b , c , d і один вихідний X сигнали. Сигнал X набуває значення 1 після надходження сигналів a і c або b і зберігає це значення після зняття цих сигналів. Сигнал X вимикається ($X = 0$) після надходження сигналу d незалежно від стану сигналів a, b, c .

7. Умови роботи схеми подано циклограмою на рис. 3.6.

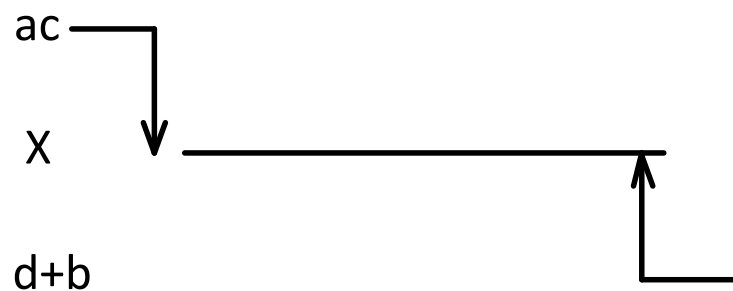


Рисунок 3.6 – Задана циклограма

8. Умови роботи схеми подано циклограмою на рис. 3.7.

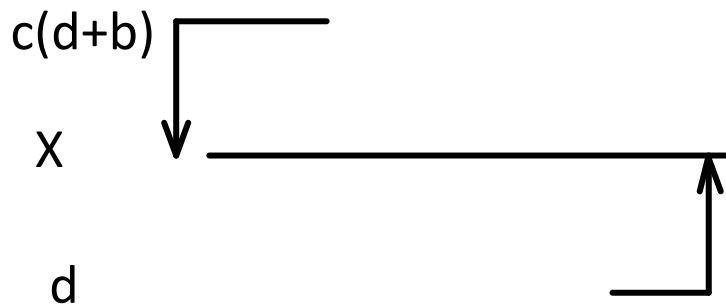


Рисунок 3.7 – Задана циклограма

9. Схема має два вхідні сигнали a , b і два вихідні сигнали X та Y . Стан входів – двійкове число. Якщо це число при зміні входів збільшувалося, то $X=1$, $Y=0$; якщо зменшувалося, то $X=0$, $Y=1$.

10. Схема має два вхідні сигнали a , b і два вихідні X , Y . Сигнал $X=1$, якщо обидва сигнали a і b дорівнюють 1, але сигнал a надійшов раніше, ніж сигнал b . Сигнал $Y=1$, якщо $a=1$, $b=1$, але сигнал b надійшов раніше, ніж сигнал a . У будь-який момент часу може змінюватися стан не більше ніж одного виходу.

11. Схема має два вхідні сигнали a , b і один вихідний X . Якщо $b=1$, то значення X співпадає зі значенням a , якщо ж $b=0$, то X повинен зберігати своє останнє значення, яке було до того моменту, коли сигнал b став рівним нулю. У будь-який момент часу може змінюватися стан не більше ніж одного входу.

12. Схема має два вхідні сигнали a , b і два вихідні X , Y . Якщо сигнал a набуває значення 1 першим, то $X=1$, а якщо сигнал $b=1$ перший, то $Y=1$.

13. Виконати синтез схеми керування за допомогою кнопок «Пуск» і «Стоп». Світлодіод вмикається із затримкою 3τ після натискання кнопки «Пуск» і вимикається із затримкою 8τ після натиснення кнопки «Стоп». Час натиснення кнопок менший за затримку часу.

14. Виконати синтез схеми керування за допомогою кнопок «Пуск» і «Стоп». Світлодіод вмикається із затримкою 3τ після відпускання кнопки «Пуск» і вимикається після натиснення кнопки «Стоп». Час натиснення кнопок менший за затримку часу.

15. Виконати синтез схеми керування за допомогою кнопок «Пуск» і «Стоп». Світлодіод вмикається після натискання кнопки «Пуск» і вимикається із затримкою 8с після відпускання кнопки «Стоп». Час натиснення кнопок менший за затримку часу.

16. Виконати синтез схеми керування за допомогою кнопок «Пуск» і «Стоп». Світлодіод вмикається із затримкою 7с після натиснення кнопки «Пуск» і вимикається після відпускання кнопки «Стоп». Час натиснення кнопок менший за затримку часу.

3.3 Опис лабораторної установки

Схема передньої панелі лабораторного стенду для дослідження релейно-контактних схем автоматики представлено на рисунку 3.8.

На функціональній схемі представлено такі елементи :

- проміжні реле X1-X5 (мають 4 пари розмикаючих/замикаючих контактів), що використовуються у якості вихідних та проміжних змінних;
- реле часу T1, T2;
- клеми для складання схеми;
- кнопки з самоблокуванням SB1-SB5, які використовуються у якості вхідних змінних;
- світлодіодні лампочки HL1-HL7, що фіксують наявність вихідного сигналу на кожному реле та HL11, що сигналізує наявність напруги живлення стенду;
- автоматичний вимикач QF1, за допомогою якого подається та знімається живлення стенда.

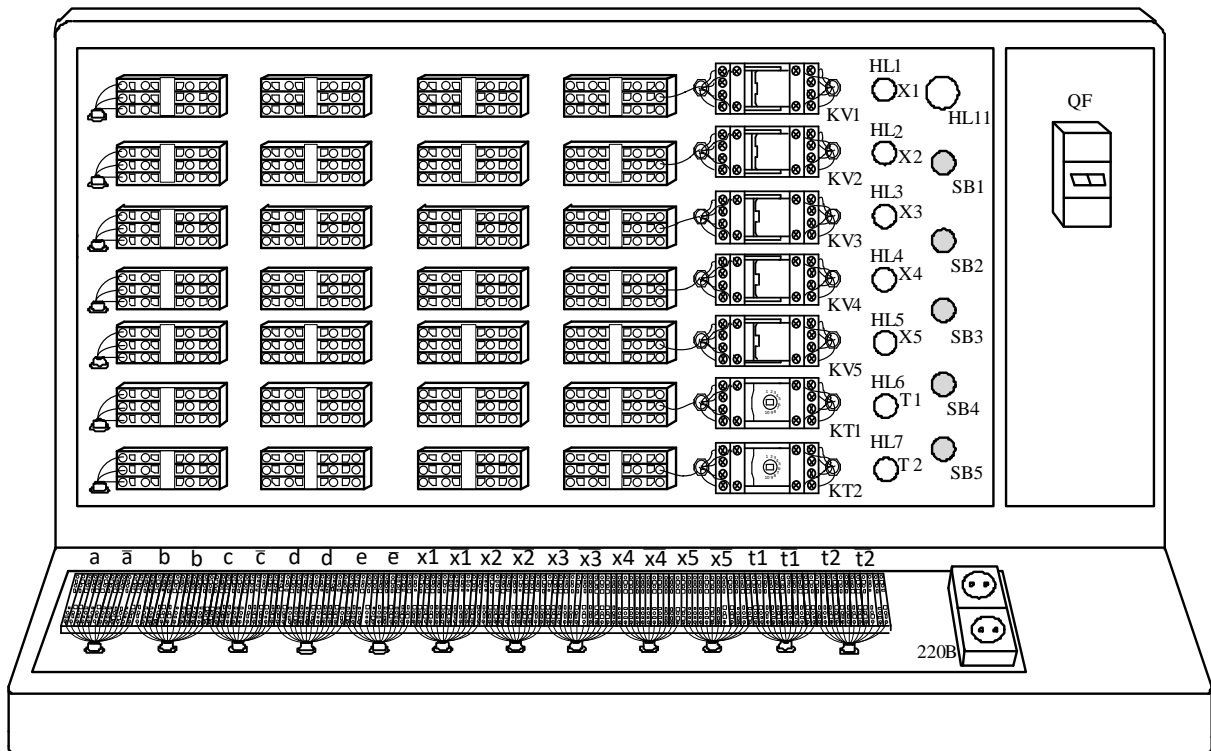


Рисунок 3.8 – Зовнішній вигляд передньої панелі лабораторного стенда

3.4 Методичні вказівки до виконання лабораторної роботи

Принцип роботи лабораторного стенда полягає у наступному:

Живлення +24 В подається на кожну ліву клему передньої панелі. За допомогою провідників крайня ліва клемка з'єднується з крайньою правою, через середні клеми або напряму, формуючи потрібну релейно-контактну схему. Оскільки середні клеми не підключені до живлення та до реле, то їх можна використовувати у довільному порядку та конфігурації. Останній контакт у кожному колі необхідно підключати до клеми, що з'єднана із необхідним реле. У нижній частині стенду виведені за допомогою клем контакти кнопок та реле. Кожна кнопка та кожне реле мають по три замикаючих або розмикаючих контактів, які нумеруються 1-3 відповідно. Якщо завдання потребує використання реле часу, то потрібно налаштувати реле на необхідну затримку по часу. Для налаштування реле часу треба вибрати потрібну затримку на його корпусі (0,1 с. – 100 год.). Після всіх налаштувань та збирання релейно-контактної схеми на стенд подається живлення та перевіряються умови, які задані в завданнях.

При виконанні лабораторної роботи важливо пам'ятати, що кожне реле містить 4 пари розмикаючих/замикаючих контактів, причому одна з них застосовуються для керування світлодіодом сигналізації спрацювання реле. Отже, релейно-контактну схему, яка містить більше трьох однакових прямих/інверсних контактів неможливо зібрати, тому її необхідно перетворити та спростити.

Основи роботи в середовищі EKTS

EKTS розроблено, щоб допомогти створювати віртуальні електромеханічні системи, що дозволяє візуалізувати спосіб функціонування цих систем та ідентифікувати можливі помилки.

Загальний вигляд робочого вікна середовища EKTS наведено на рис. 3.9.

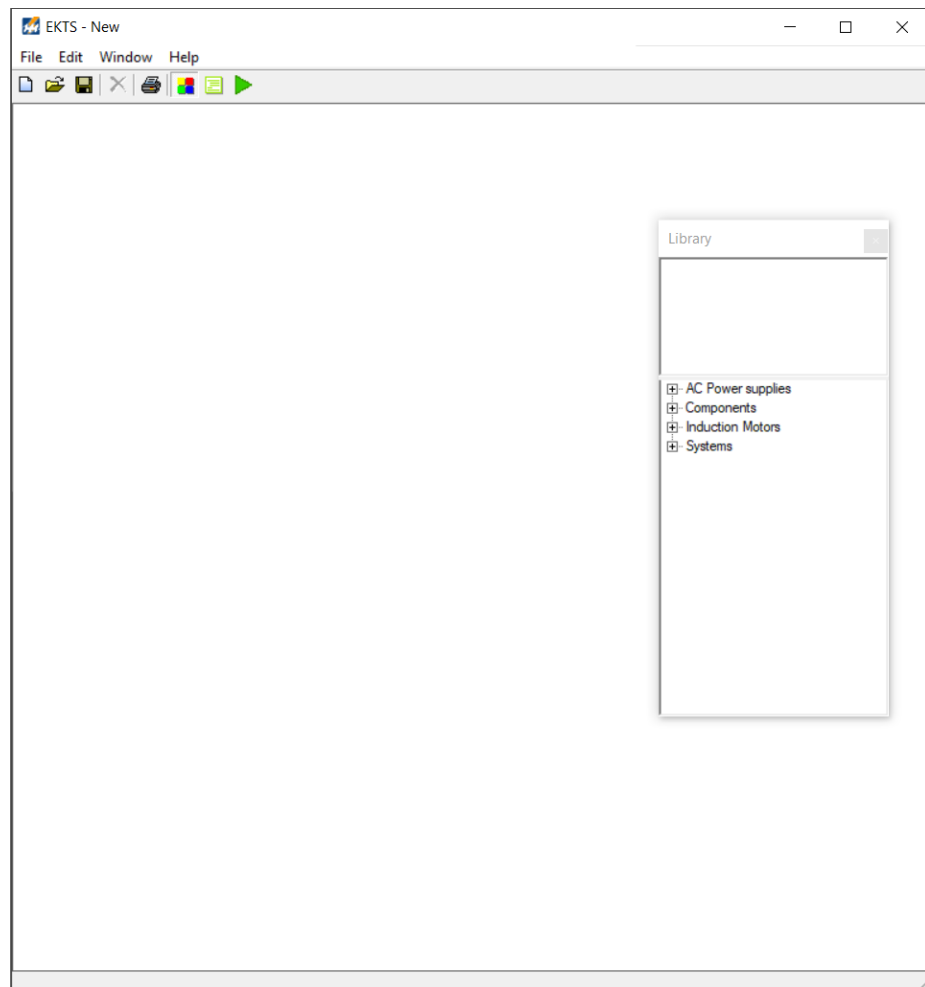




Рисунок 3.9 – Робоче вікно середовища EKTS

Програма пропонує набір заздалегідь визначених елементів, які можна використовувати при побудові електромеханічних систем. Щоб створити нову систему, необхідно вибрати потрібні елементи з бібліотек у розділі Library,

розташувати їх на робочому просторі та повернути при необхідності, щоб їх можна було з'єднати. Надаються різні моделі реле, двигунів, реле часу, перемикачів і кнопок, а також набір простих механічних систем, які можуть бути використані як відправна точка для менш досвідчених учнів.

Після створення нової системи можна писати коментарі про свій проєкт, а функція «Список використаних елементів» в розділі «Вікно» дозволяє відстежувати перелік елементів, які використані у своєму проєкті.

Працездатність розробленої схеми можливо перевірити в режимі симуляції. Для початку симуляції необхідно натиснути піктограму  «Run». Для зупинки симуляції треба натиснути кнопку  «Stop».

Приклад проєкту релейно-контактної схеми наведений на рис. 3.10.

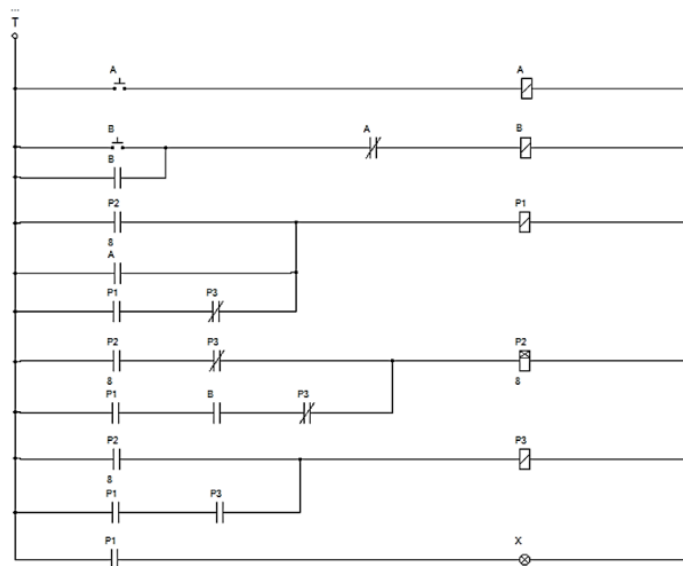


Рисунок 3.10 – Приклад релейно-контактної схеми в середовищі EKS

3.5 Зміст звіту

Звіт має містити такі матеріали дослідження кожної схеми:

- 1) титульний аркуш із зазначенням назви дисципліни та лабораторної роботи (див. Додаток А);
- 2) мету роботи, програму роботи та умови роботи схеми відповідно до свого варіанту;
- 3) логічний синтез схеми;

- 4) логічні вирази за якими безпосередньо можна побудувати принципові схеми;
- 5) релейно-контактні схеми, дослідження яких виконано в лабораторії, з підписами викладача;
- 6) схеми у програмному забезпеченні ECTS;
- 7) висновки про відповідність роботи схеми заданим умовам.

3.6 Контрольні запитання

1. Яким чином вибирається кількість проміжних змінних та розміщуються їх стани під час синтезу з урахуванням технологічних затримок?
2. У якому випадку необхідно ввести проміжний елемент, змінювання стану якого дає команду починати відлік затримання часу?
3. Яким чином відбувається з'єднання схем на лабораторному стенді?
4. Яке обмеження вхідних/вихідних і проміжних сигналів має стенд при збиранні релейно-контактних схем?
5. Яким чином відбувається налаштування таймерів?
6. За допомогою яких елементів реалізуються проміжні змінні?

Лабораторна робота № 4

ПРИСТРОЇ ВВОДУ ТА ВИВОДУ ПЛАТИ DE10-Lite

Тривалість лабораторної роботи – 4 год.

Тривалість домашньої роботи – 4 год.

Мета роботи – навчитися програмувати на мові FBD у середовищі Quartus Prime та використовувати периферичні пристрої вводу/виводу інформації плати розробника DE10-Lite з ПЛІС Altera сімейства MAX 10 для реалізації одноктактних та багатотактних схем систем автоматизації промислових механізмів.

4.1 Основні теоретичні відомості

4.1 Загальний огляд плати DE10-Lite. Пристрої вводу/виводу

Плата укомплектована широким набором апаратних засобів, які дозволяють реалізувати найрізноманітніші проекти. Загальний вигляд плати та її основні елементи вказані на рис. 4.1.

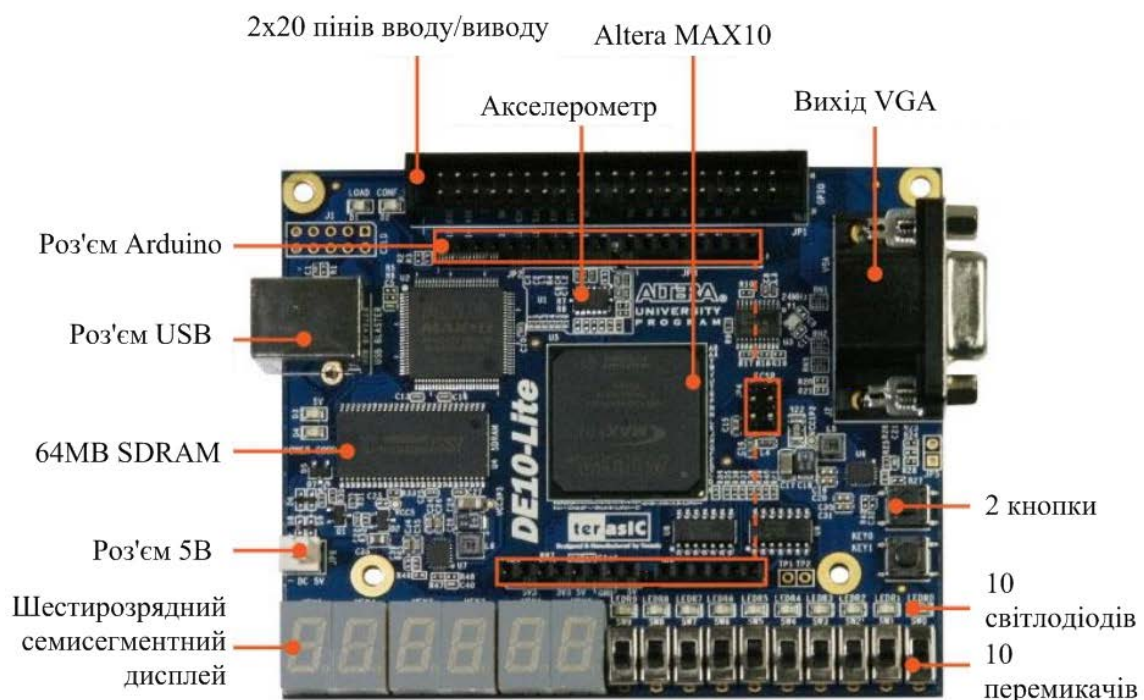


Рисунок 4.1 – Загальний вигляд та основні компоненти плати DE10-Lite

Апаратні ресурси плати включають:

- вбудований програматор USB Blaster;
- пам'ять SDRAM на 64МБ, з 16-бітною шиною зв'язку;
- роз'єм на 2x20 контактів вводу/виводу;
- роз'єм стандарту Arduino Uno R3 з шістьма каналами АЦП;
- чотирьохбітний резисторний ЦАП для відеосигналу VGA;
- 10 світлодіодів;
- 10 перемикачів;
- 2 кнопки зі схемою анти-дріб'язку контактів
- шестирозрядний дисплей на семисегментних індикаторах.

На рис. 4.2 наведено функціональну схему плати

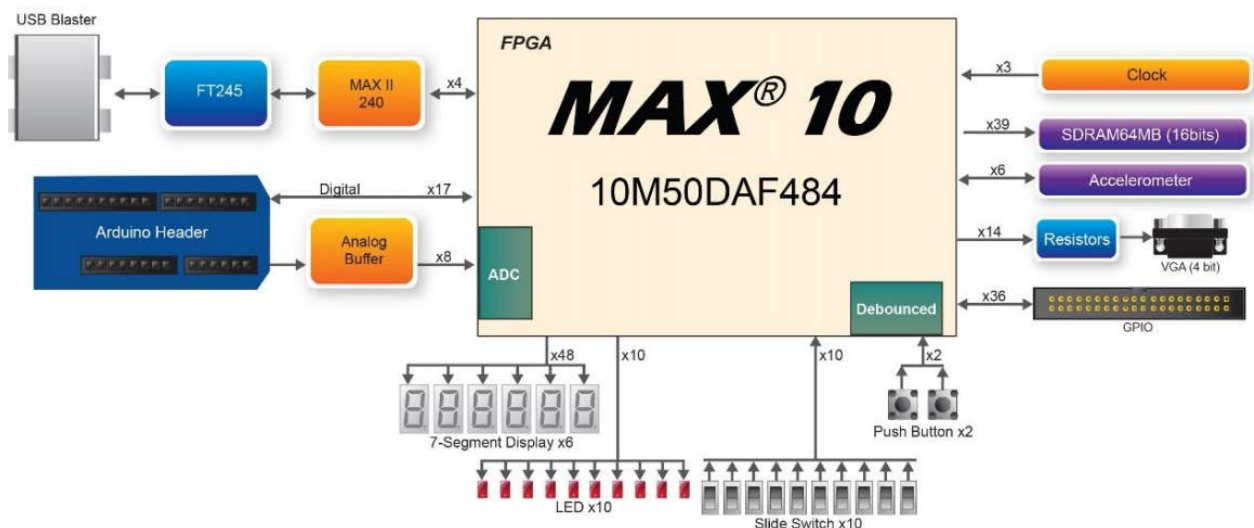


Рисунок 4.2 – Функціональна схема

В даній роботі будуть розглянуті методика роботи з найпростішими пристроями вводу та індикації двійкового коду і числової інформації – перемикачами, кнопками, світлодіодними індикаторами та семисегментним дисплеєм.

4.2. Пристрої вводу інформації

4.2.1. Кнопки зі схемою анти-дріб'язку

На платі доступні дві кнопки, підключені через тригер Шмітта як вказано на рис. 4.3. Це дозволяє використовувати в програмах сигнали від кнопок

напряму, без введення програмної схеми на RS-тригері, що потребує двох кнопок для одного сигналу. Отримана часова діаграма роботи кнопок вказана на рис. 4.4.

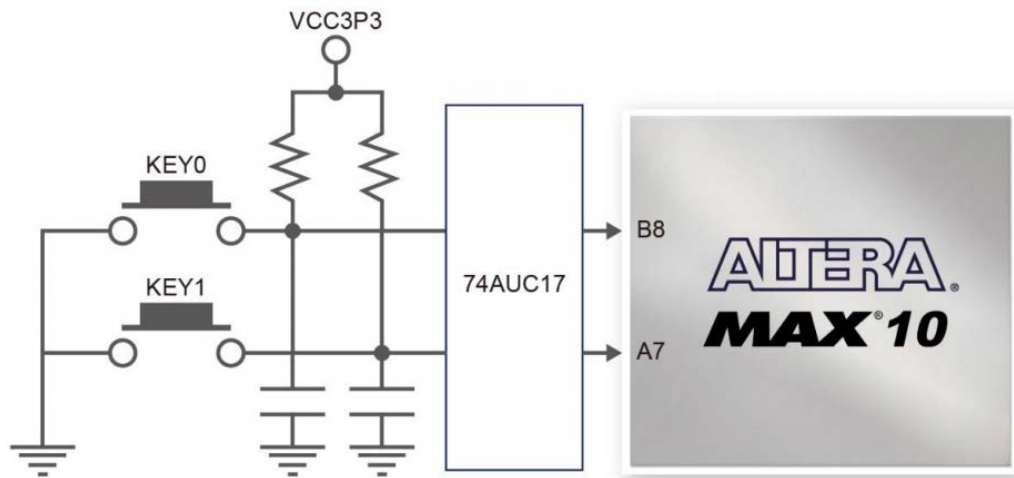


Рисунок 4.3 – Схема підключення кнопок

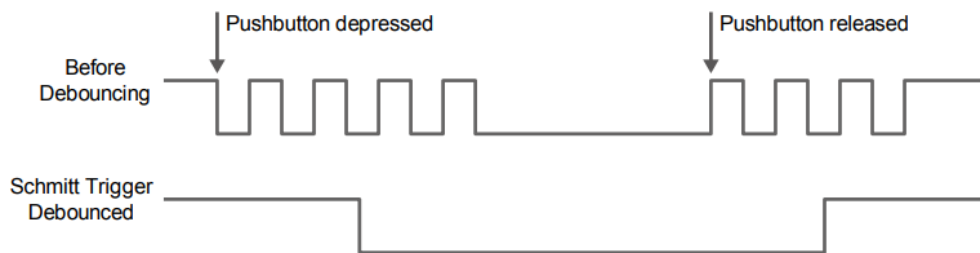


Рисунок 4.4 – Часова діаграма роботи кнопок

Всі периферичні пристрої плати, такі як кнопки, індикатори, роз'єми мають жорстко призначену адресацію підключень, оскільки контакти периферичних пристроїв електрично з'єднані з певними контактами ПЛІС через провідники друкованої плати. Тому при створенні програм необхідно користуватися таблицею підключень, в якій вказано адресацію сигналів плати. Адресація кнопок вказана в таблиці 4.1.

Таблиця 4.1 – Адресація кнопок

Назва сигналу	Адреса контакту ПЛІС	Опис сигналу	Тип контакту
KEY0	PIN_B8	Push-button[0]	3.3 V SCHMITT TRIGGER
KEY1	PIN_A7	Push-button[1]	3.3 V SCHMITT TRIGGER

4.2.2. Перемикачі

На платі доступні десять перемикачів, кожен з яких під'єднаний до контакту ПЛІС. Верхнє положення перемикача відповідає сигналу логічної одиниці, нижнє – логічного нуля. Адресація перемикачів вказана в таблиці 4.2.

Таблиця 4.2 – Адресація перемикачів

Назва сигналу	Адреса контакту ПЛІС	Опис сигналу	Тип контакту
SW0	PIN_C10	Slide Switch[0]	3.3-V LVTTL
SW1	PIN_C11	Slide Switch[1]	3.3-V LVTTL
SW2	PIN_D12	Slide Switch[2]	3.3-V LVTTL
SW3	PIN_C12	Slide Switch[3]	3.3-V LVTTL
SW4	PIN_A12	Slide Switch[4]	3.3-V LVTTL
SW5	PIN_B12	Slide Switch[5]	3.3-V LVTTL
SW6	PIN_A13	Slide Switch[6]	3.3-V LVTTL
SW7	PIN_A14	Slide Switch[7]	3.3-V LVTTL
SW8	PIN_B14	Slide Switch[8]	3.3-V LVTTL
SW9	PIN_F15	Slide Switch[9]	3.3-V LVTTL

4.2.3. Світлодіоди

На платі доступний двійковий десятирозрядний дисплей, що складаються зі світлодіодів, підключених до контактів ПЛІС через окремі струмообмежувальні резистори. Присвоєння відповідному контакту ПЛІС високого логічного рівня вмикає світлодіод, низького рівня – вимикає. Якщо контакту ПЛІС, до якого підключений світлодіод, не присвоєно логічного рівня, то світлодіод буде світитися в половину потужності, оскільки контакти ПЛІС мають внутрішню «підтяжку» до живлення. Тому всім світлодіодним виводам, що не використовуються, слід програмно призначати значення логічного нуля, щоб уникнути подальших похибок при перевірці роботи програми на платі. Адресація світлодіодів вказана в таблиці 4.3.

Таблиця 4.3 – Адресація світлодіодів

Назва сигналу	Адреса контакту ПЛІС	Опис сигналу	Тип контакту
LEDR0	PIN_A8	LED[0]	3.3-V LVTTL
LEDR1	PIN_A9	LED[1]	3.3-V LVTTL
LEDR2	PIN_A10	LED[2]	3.3-V LVTTL
LEDR3	PIN_B10	LED[3]	3.3-V LVTTL
LEDR4	PIN_D13	LED[4]	3.3-V LVTTL
LEDR5	PIN_C13	LED[5]	3.3-V LVTTL
LEDR6	PIN_E14	LED[6]	3.3-V LVTTL
LEDR7	PIN_D14	LED[7]	3.3-V LVTTL
LEDR8	PIN_A11	LED[8]	3.3-V LVTTL
LEDR9	PIN_B11	LED[9]	3.3-V LVTTL

4.2.4. Семисегментні індикатори

На платі доступні шість семисегментних світлодіодних індикаторів зі спільним анодом. Це означає, що для увімкнення необхідного сегменту індикатора, необхідно призначити відповідному виводу ПЛІС низький логічний рівень. Схема підключення одного розряду індикатора та нумерація його сегментів вказана на рис. 4.5, адресація вказана в таблиці 4.4.

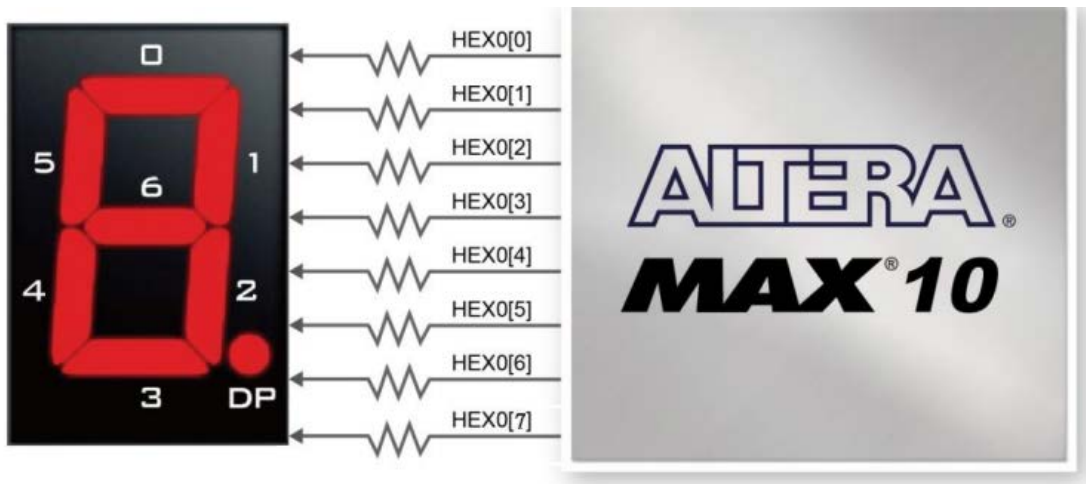


Рисунок 4.5 – Схема підключення індикатора

Таблиця 4.4 – Адресація семисегментних індикаторів

Назва сигналу	Адреса контакту ПЛІС	Опис сигналу	Тип контакту
HEX00	PIN_C14	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX01	PIN_E15	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX02	PIN_C15	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX03	PIN_C16	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX04	PIN_E16	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX05	PIN_D17	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX06	PIN_C17	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX07	PIN_D15	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX10	PIN_C18	Seven Segment Digit 0[0]	3.3-V LVTTL

HEX11	PIN_D18	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX12	PIN_E18	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX13	PIN_B16	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX14	PIN_A17	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX15	PIN_A18	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX16	PIN_B17	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX17	PIN_A16	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX20	PIN_B20	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX21	PIN_A20	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX22	PIN_B19	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX23	PIN_A21	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX24	PIN_B21	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX25	PIN_C22	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX26	PIN_B22	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX27	PIN_A19	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX30	PIN_F21	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX31	PIN_E22	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX32	PIN_E21	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX33	PIN_C19	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX34	PIN_C20	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX35	PIN_D19	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX36	PIN_E17	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX37	PIN_D22	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX40	PIN_F18	Seven Segment Digit 0[0]	3.3-V LVTTL

HEX41	PIN_E20	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX42	PIN_E19	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX43	PIN_J18	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX44	PIN_H19	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX45	PIN_F19	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX46	PIN_F20	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX47	PIN_F17	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX50	PIN_J20	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX51	PIN_K20	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX52	PIN_L18	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX53	PIN_N18	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX54	PIN_M20	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX55	PIN_N19	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX56	PIN_N20	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX57	PIN_L19	Seven Segment Digit 0[0]	3.3-V LVTTL

4.2 Програма роботи

1. Ознайомитися з функціональною схемою плати та доступними на ній периферичними пристроями вводу/виводу.
2. Виконати синтез схем за завданням згідно з варіантом.
3. Засобами програмного середовища Quartus Prime скласти FBD-програму за синтезованою схемою.
4. Перевірити правильність виконання завдання на реальній платі.
5. Оформити протокол по лабораторній роботі та зробити висновки.

4.3 Завдання до лабораторної роботи

Завдання 1. Синтезувати схему індикації десятирозрядного двійкового числа на світлодіодному індикаторі. Вивід числа на індикатор та його скидання виконати однією кнопкою. Число – сума номеру варіанту за списком та дня народження.

Завдання 2. Синтезувати схему напівавтоматичного керування восьмидвигунним конвеєром відповідно до номеру варіанту за списком. Передбачити індикацію стану вихідних змінних схеми на світлодіодному дисплеї. Синтез схеми виконати методом графопереходів. Тактовий сигнал перемикання двигунів та сигнал скидання тригерів подавати кнопками.

Таблиця 4.5 – Завдання 2 відповідно до номеру варіанту

Номер варіанту	Циклограма вмикання двигунів
Парний	Вихідний стан → M1 → M2 → M3 → M4 → M5 → M6 → M7 → M8 → Вихідний стан → M1 → ...
Непарний	Вихідний стан → M1 → M2 → M3 → M4 → M3 → M2 → M1 → Вихідний стан → M1 → ...
Кратний трьом*	Вихідний стан → M1, M8 → M2, M7 → M3, M6 → M4, M5 → Вихідний стан → → M1, M8 → ...

*Дана умова має першочергову важливість (наприклад, студент з номером варіанту 6 виконує завдання для варіантів кратних трьом, а не для парних).

Завдання 3. Творче завдання. Синтезувати схему запам'ятовуючого пристрою 1x8 біт на RS-тригерах, інформацію з якого можна вивести у однібітну шину даних за допомогою восьмирозрядного мультиплексора. Для варіантів №1-15 за списком групи ввести в запам'ятовуючий пристрій номер варіанту, для інших варіантів – номер місяцю народження. Передбачити індикацію стану комірок запам'ятовуючого пристрою та шини даних за допомогою світлодіодного дисплею та скидання комірок запам'ятовуючого

пристрою кнопкою. Стан входів адресації мультиплексора та комірок запам'ятовуючого пристрою задавати перемикачами.

4.4 Методичні вказівки до виконання лабораторної роботи

Приклад 1. Завдання 1: для синтезу схеми доцільно використати RS-тригер у якості перемикаючого елементу. Перевести з десяткового коду в двійковий задане згідно з варіантом число слід заздалегідь. Виводити число слід у наступному вигляді: LEDR0 – молодший розряд, всі невикористані світлодіоди – вимкнені (подати на відповідні входи логічний нуль). Приклад синтезованої схеми виводу числа 64 наведено на рис. 4.6.

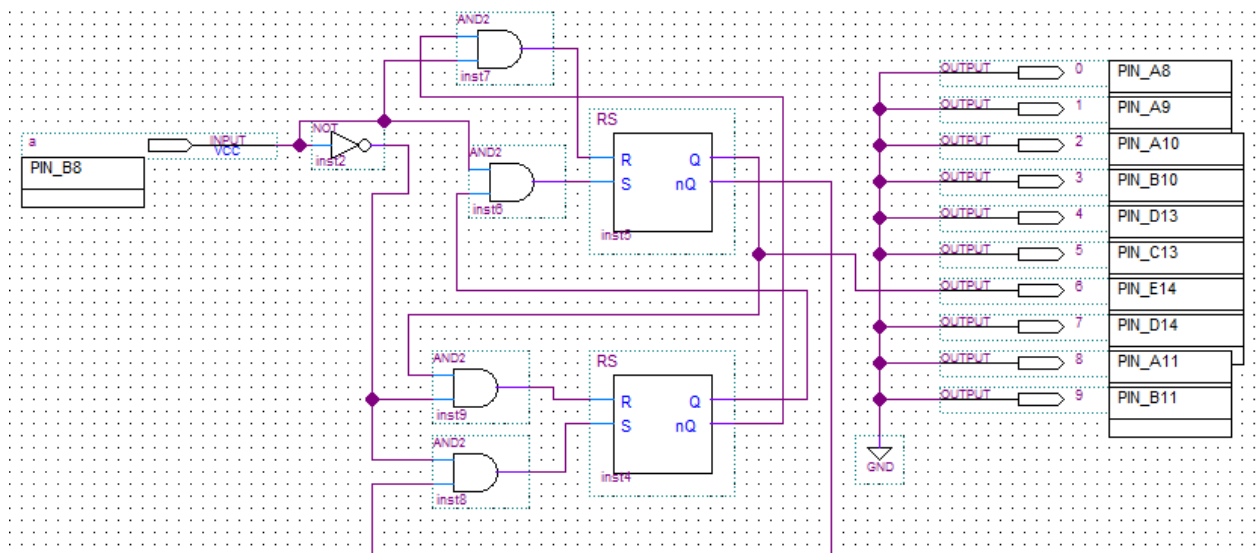


Рисунок 4.6 – Схема до прикладу 1

Приклад 2. Завдання 2: при синтезі схеми особливу увагу приділити правильності складу вихідних рівнянь графу. Індикацію стану двигунів конвеєра виконати у вигляді: LEDR0 – M1, LEDR1 – M2 і т.д., всі невикористані світлодіоди – вимкнені (подати на відповідні входи логічний нуль). На рис. 4.7 наведено приклад схеми до спрощеної задачі, що реалізує керування 3 двигунами у порядку вмикання $M1 \rightarrow M2 \rightarrow M3$, синтез схеми виконано методом графопереходів.

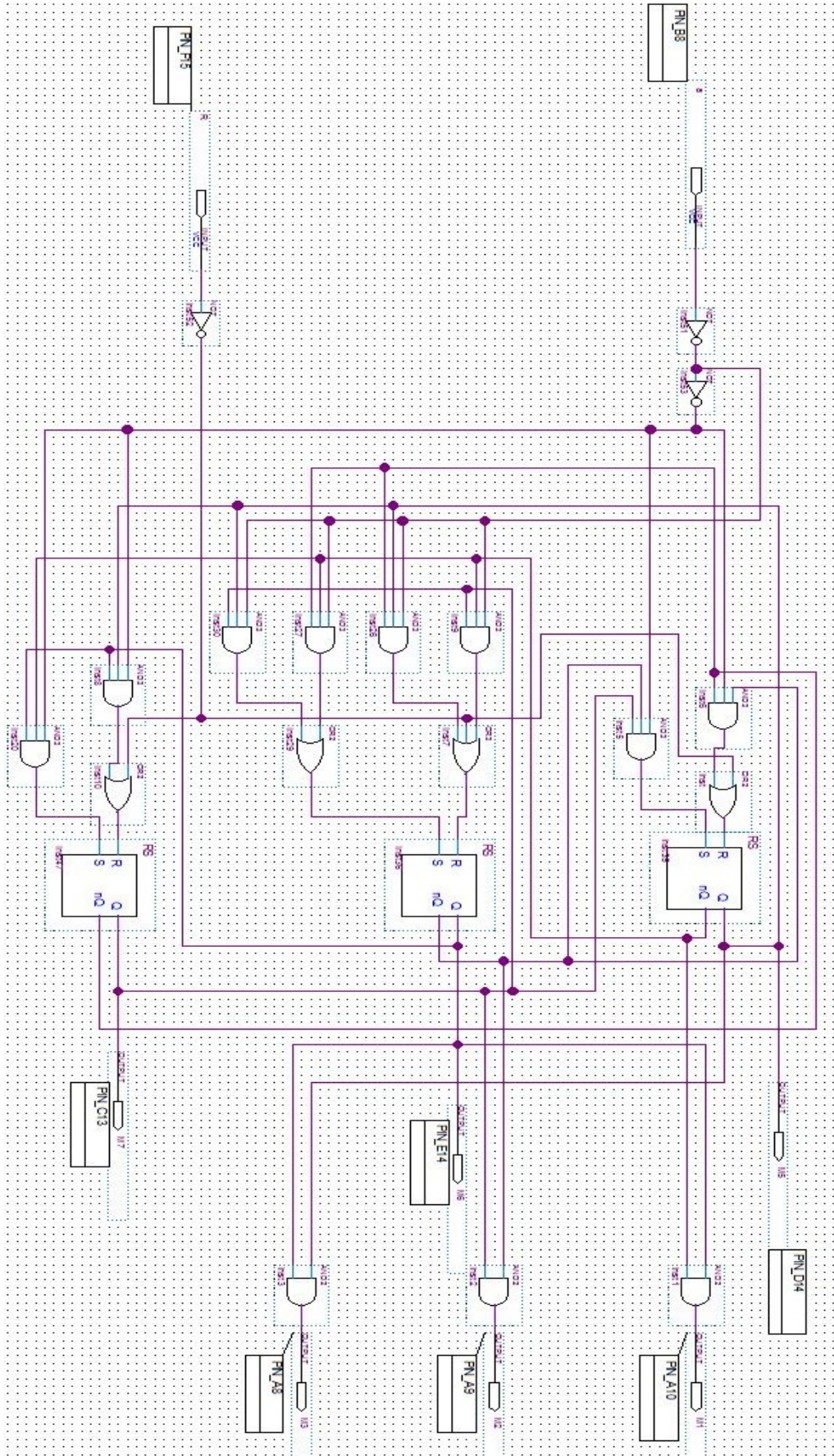


Рисунок 4.7 – Схема до прикладу 2

Приклад 3. Завдання 3: загальний структуру побудови схеми показано на рисунку 4.8. В зв'язку з недостатньою кількістю перемикачів на платі, стан молодшого розряду адреси мультиплектора (вхід С на рисунку) задавати за допомогою кнопки. Схему восьмирозрядного мультиплектора наведено на рис. 4.9.

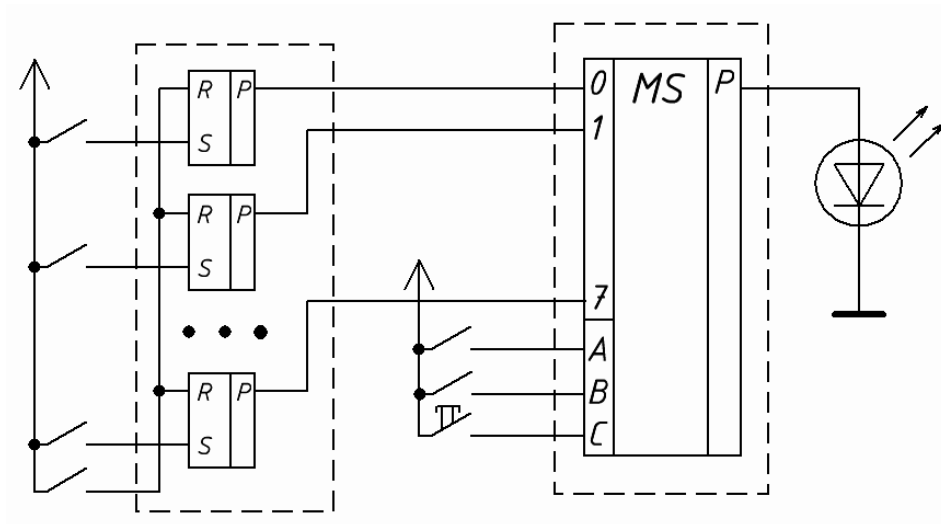


Рисунок 4.8 – Загальна структура побудови схеми

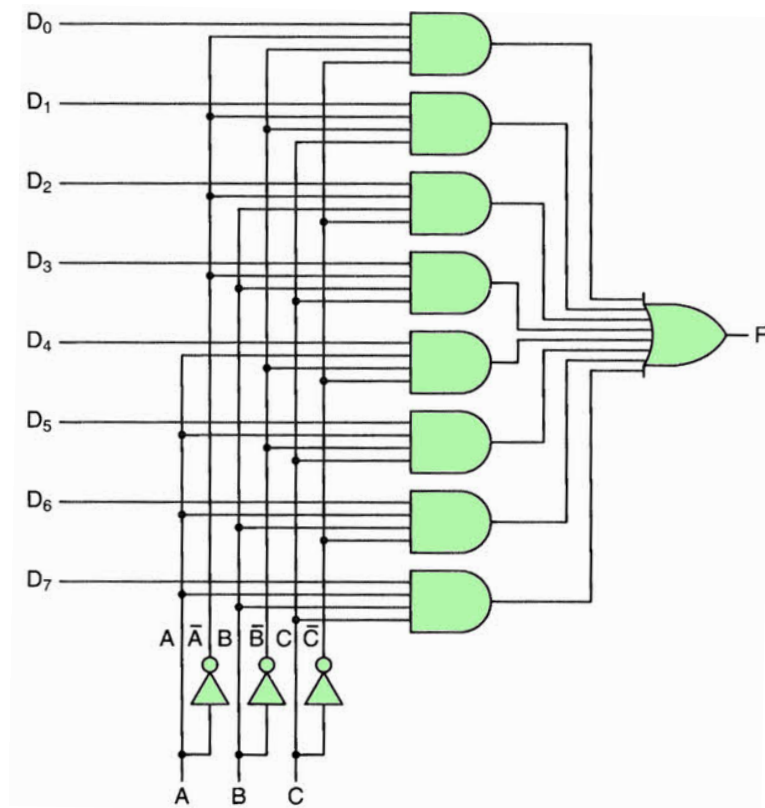


Рисунок 4.9 – Схема восьмирозрядного мультиплектора

Функціонально, схема запам'ятовуючого пристрою складається з 4 основних блоків: масив зберігання інформації, блок мультиплектору, блок

засобів вводу інформації та керування, блок індикації стану комірок і лінії даних. Для перевірки правильності вибраного підходу розробки схеми, слід спочатку зібрати масив тригерів із блоком індикації стану тригерів і перевірити правильність його роботи. Після цього, схема доповнюється мультиплексором, і перевіряється правильність виконання завдання згідно з варіантом.

Опис програмного пакета QUARTUS Prime 15.1

Створення нового проекту

Відкривши пакет Quartus Prim, вибираємо з меню File пункт New Project Wizard... – майстер створення нових проектів. Потрапляємо у вікно для завдання поточної директорії проекту. Заповнюємо три рядки як показано на рис. 4.10 і натискаємо кнопку Next.

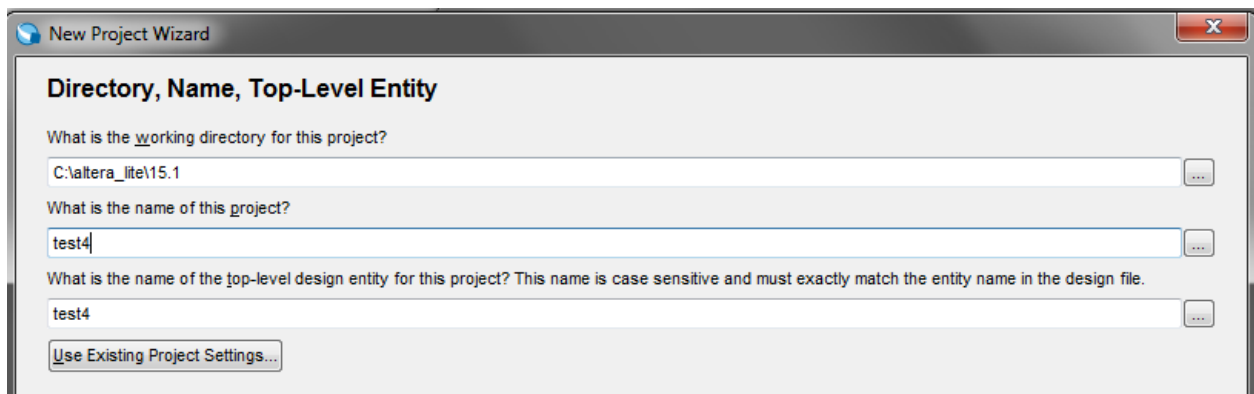


Рисунок 4.10 – Меню завдання поточної директорії проекту

В даному випадку проект буде названо test4 і буде створена директорія за адресою c:\altera_lite\15.1\test4.

Після натиснення кнопки Next з'являється вікно, в якому можна вибрати тип проекту – пустий, чи за шаблоном. Вибираємо Empty project.

З'являється вікно, в якому можна вибрати вже існуючий файл з програмою і додати до проекту. Натискаємо кнопку Next, пропускаючи це вікно.

Далі потрапляємо у вікно де потрібно вибрати сімейство та модель мікросхеми. Модель мікросхеми (10M50DAF484C7G) можна подивитись безпосередньо на платі, вибрати відповідний пункт у вікні, як це і показано на рис. 4.11 та натиснути кнопку Finish. Після цього новий проект створено.

Для того, щоб додати робочий файл до проекту вибираємо з меню File пункт New. У вікні, що з'явилося обираємо мову, на якій буде написана програма. Вибираємо BlockDiagram/SchematicFile та натискаємо ОК, рис. 4.12. Новий файл створено.

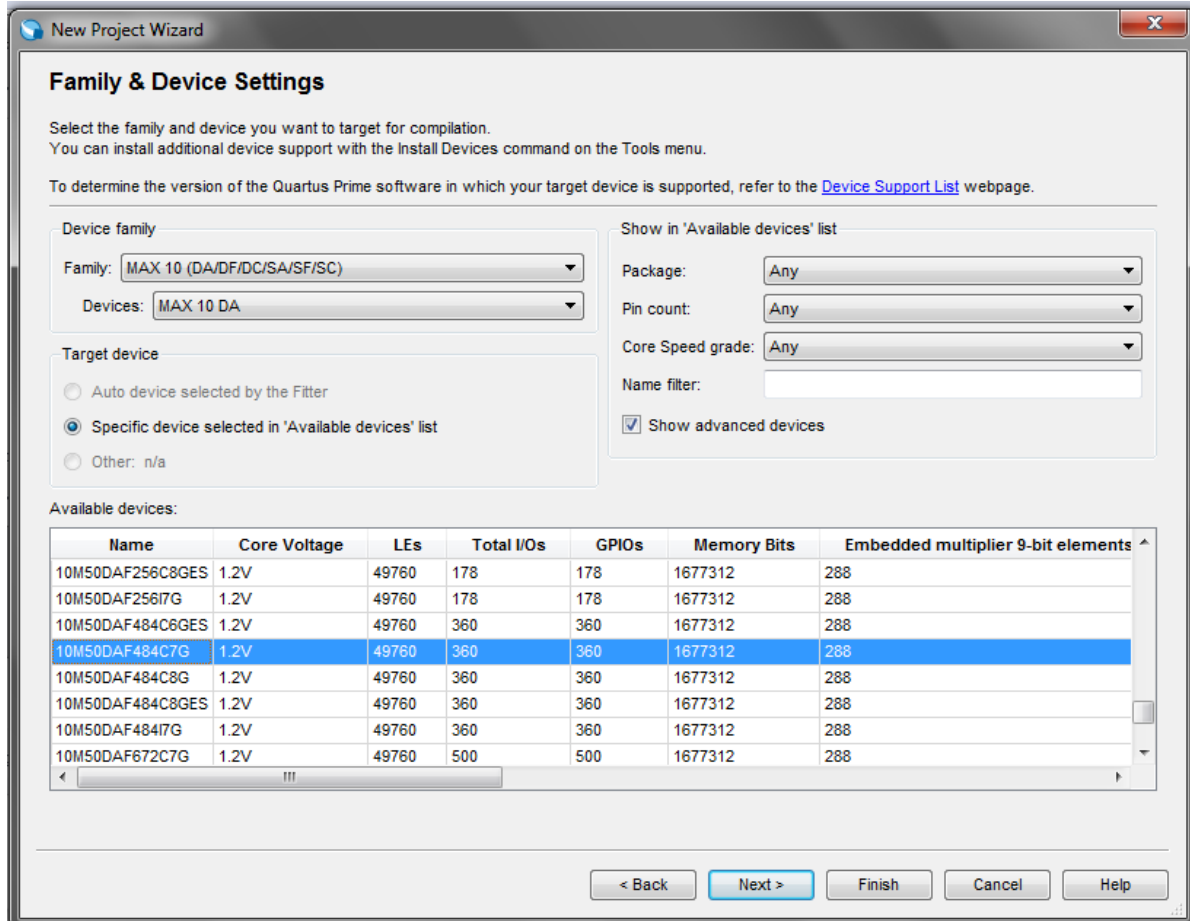


Рисунок 4.11 – Вікно вибору сімейства та моделі мікросхеми

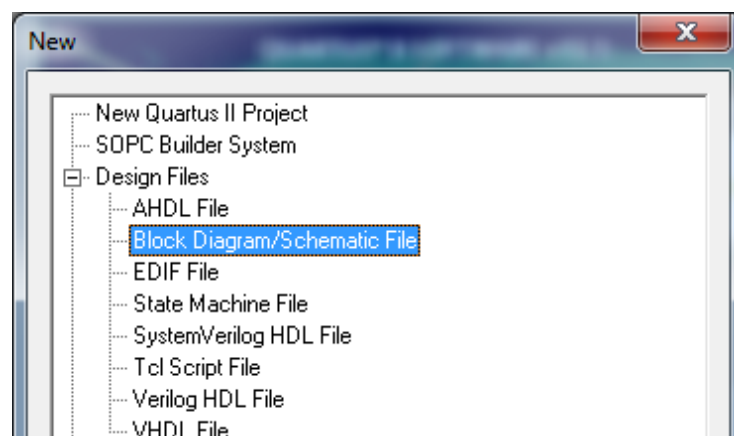


Рисунок 4.12 – Вікно вибору мови програмування та типу файлу

Створення схеми проекту

Після створення файлу проекту стає активною панель інструментів розташована ліворуч від робочої області файлу і представлена на рис. 4.13. В панелі інструментів представлені засоби для створення проекту. Інструмент SymbolTool зображений на рис. 4.13 у верхньому правому куті, натиснемо на його іконці лівою кнопкою миші, в результаті нам відкриється вікно Symbol рис. 4.14 У лівому верхньому куті представлений список стандартних бібліотек Quartus II, з основними типами елементів, що застосовуються на практиці.

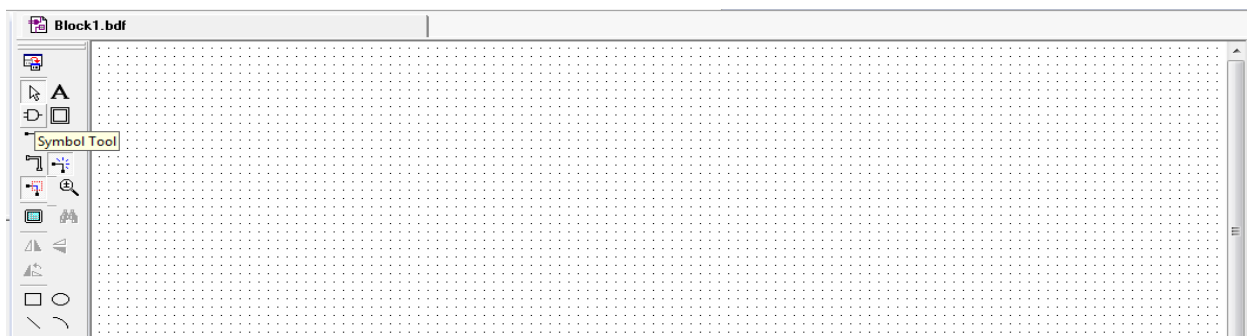


Рисунок 4.13 – Робоча область та панель інструментів

Як показано на рис. 4.14 з бібліотеки виберемо primitives / pin / input для завдання вхідних пінів. Зображення компоненти з'являється в правому вікні. Натиснемо кнопку ОК, і приступимо до безпосереднього розташуванню обраного компонента.

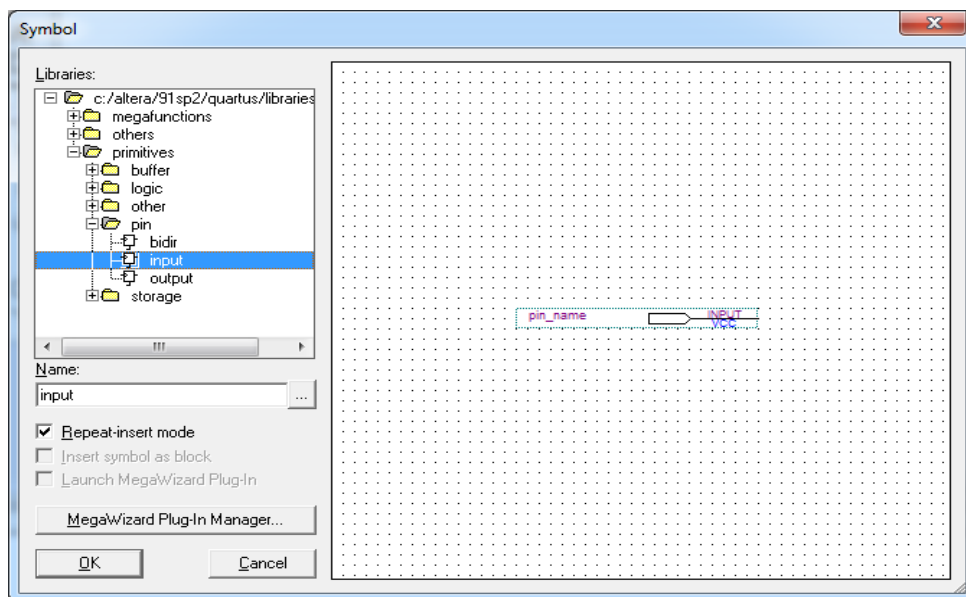


Рисунок 4.14 – Вікно Symbol

Після розташування достатньої кількості обраних компонентів в робочій області файлу Block1.bdf натиснемо праву кнопку миші і з контекстного меню

виберемо пункт Cancel, скасувавши подальше введення компоненти. Повторно скориставшись інструментом SymbolTool встановимо крім компоненти входу input, компоненту виходу output.

В бібліотеці primitives/logic можна знайти елементи, які будуть реалізувати такі логічні функції:

and – логічне «І»

or – логічне «АБО»

not – інверсія

xor – виключаюче «АБО»

Цифра після назви логічної функції означає кількість входів, яку буде мати елемент, наприклад and3 буде мати 3 входи, рис. 4.15(а). Наявність букви b перед назвою логічної функції буде означати, що елемент буде мати інверсні входи, рис. 4.15(б), а букви n – інверсний вихід, рис. 4.15(в). Наявність обох букв означатиме, що елемент матиме інверсні як входи, так і вихід, рис. 4.15(г).

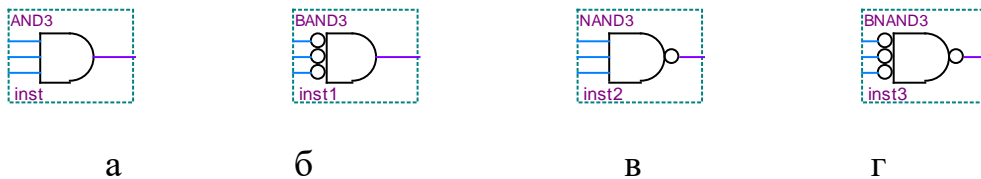


Рисунок 4.15 – Приклади логічних елементів

Побудову схеми розглянемо на прикладі створення асинхронного RS-тригера з двох елементів NOR2. Зробивши установку необхідного числа елементів, зробимо з'єднання входів і виходів компонент, помістивши курсор миші на один з входів елемента NOR2, затиснемо ліву кнопку миші і з'єднаємо і вихідним кінцем елемента входу. Проведемо аналогічну операцію для всієї схеми і отримаємо схему наведену на рис. 4.16.

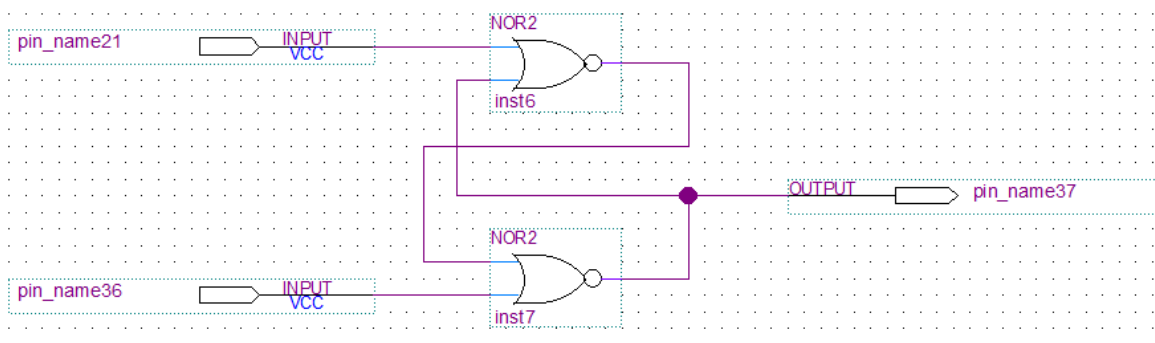


Рисунок 4.16 – Схема поточного проекту

Перейменуємо елементи. Для цього зробимо спочатку подвійне натискання лівою кнопкою миші по одному з елементів input. У результаті відкриється вікно зображене на рис. 4.17, що представляє властивості обраного Піна. У закладці General вміст рядка Pinname (s) змінимо на S, таким чином присвоївши ім'я S першому входу. Рядок Defaultvalue являє собою значення логічного сигналу на pins за замовчуванням, в даному випадку VCC тобто пін «підтягнутий» на плюс живлення. Можливо змінити значення за замовчуванням на GND або нуль. Це визначається побудовою схеми, і в нашому випадку необхідно встановити GND. Подібним чином змінимо ім'я другого входного піна на R, а вихідного на Q. Зміни імен відразу ж будуть відображатись на схемі.

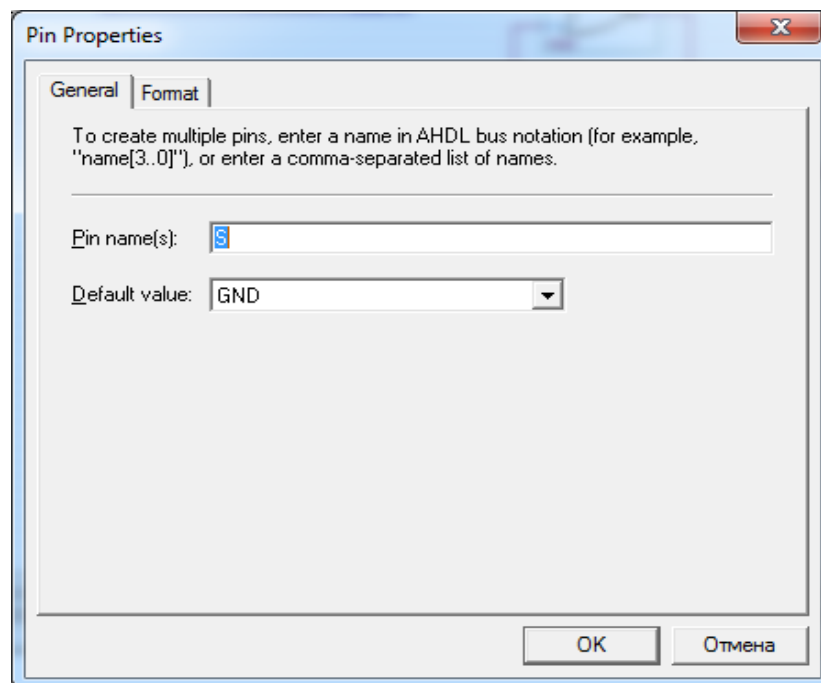




Рисунок 4.17 – Властивості піна

Далі необхідно виконати розпіновку. Перед цим необхідно виконати компіляцію проекту. Для цього необхідно натиснути кнопку StartCompilation .

По закінченні компіляції з'являється вікно з повідомленням про результати компіляції, кількості помилок і попереджень. Далі натискаємо кнопку Pin Planer . Відкриється вікно зображене на рис. 4.18, де необхідно відповідно до таблиць адресації встановити входні змінні на ті виводи, до яких підключено кнопки чи перемикачі, а вихідні змінні – на ті, до яких підключено

світлодіоди чи семисегментні індикатори. Для цього необхідно два рази клацнути на поле Location необхідного виводу та ввести адресу контакту ПЛІС.

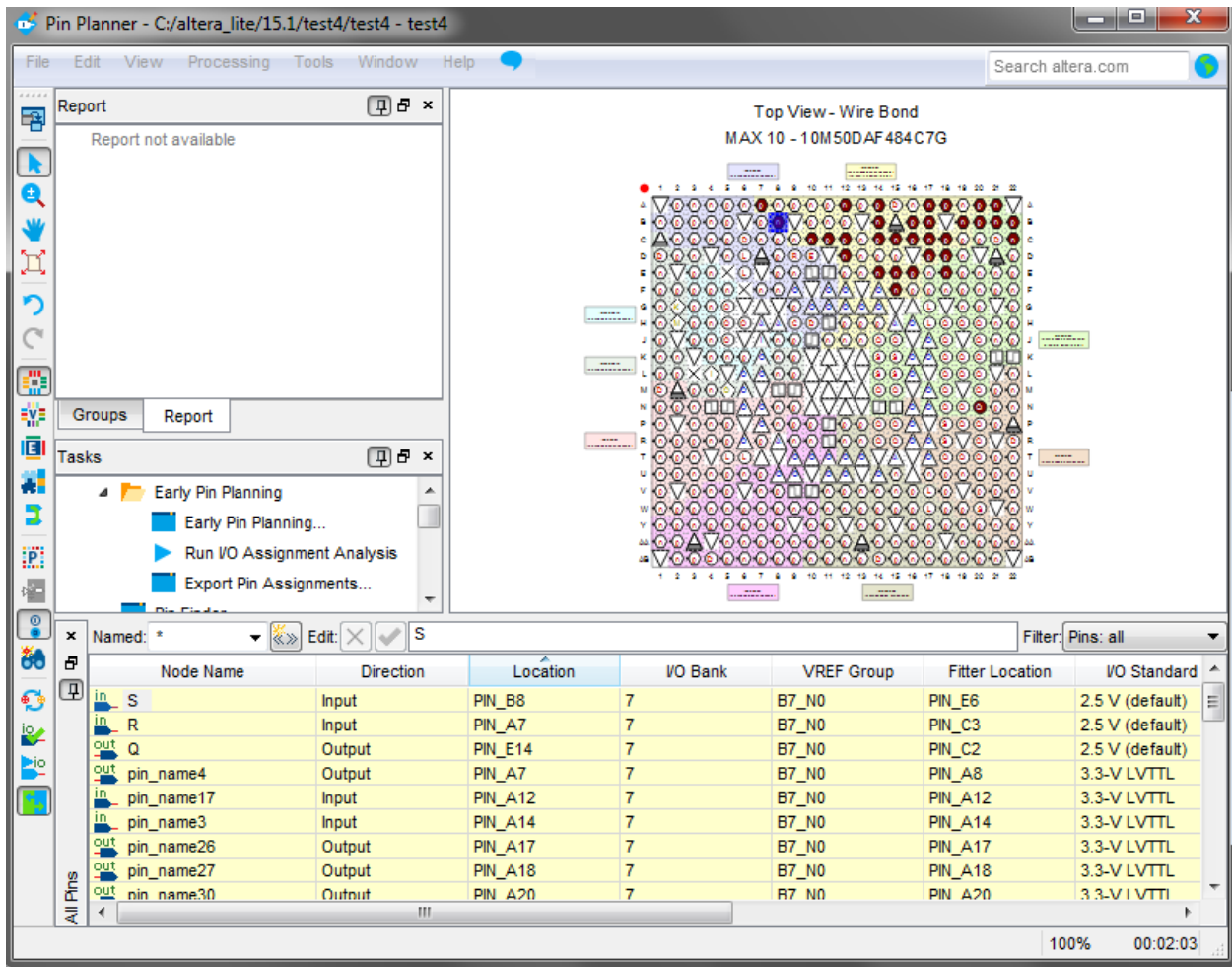


Рисунок 4.18 – Вікно Pin planner

Після цього панель розпіновки можна закрити. Результати виконаних операцій відразу будуть відображені на схемі, рис. 4.19.

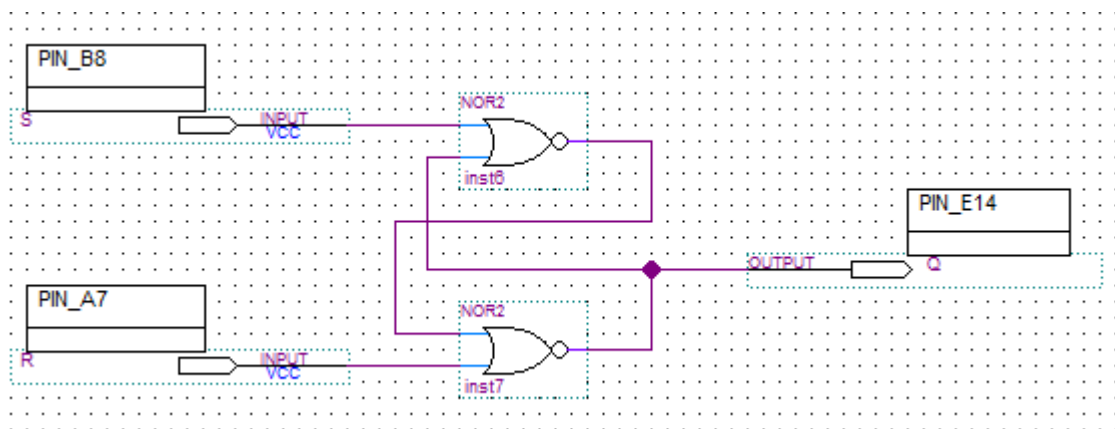


Рисунок 4.19 – Схема поточного проекту

Після цього потрібно виконати компіляцію ще раз.

Програмування плати

Після компіляції проекту на панелі Tasks зліва від робочого поля вибираємо вкладку Program Device (рис. 4.20). Плата має бути підключена до комп'ютера.

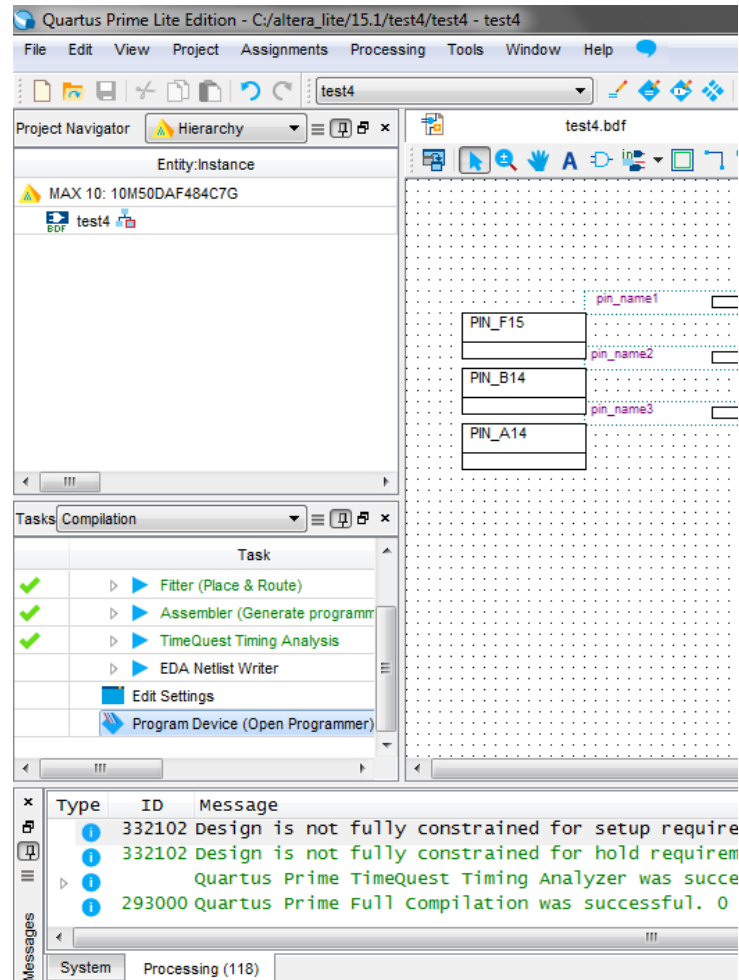


Рисунок 4.20 – Розташування вкладки Program Device

У вікні, що відкрилося, у вкладці Hardware вибираємо наш програматор (рис. 4.21)

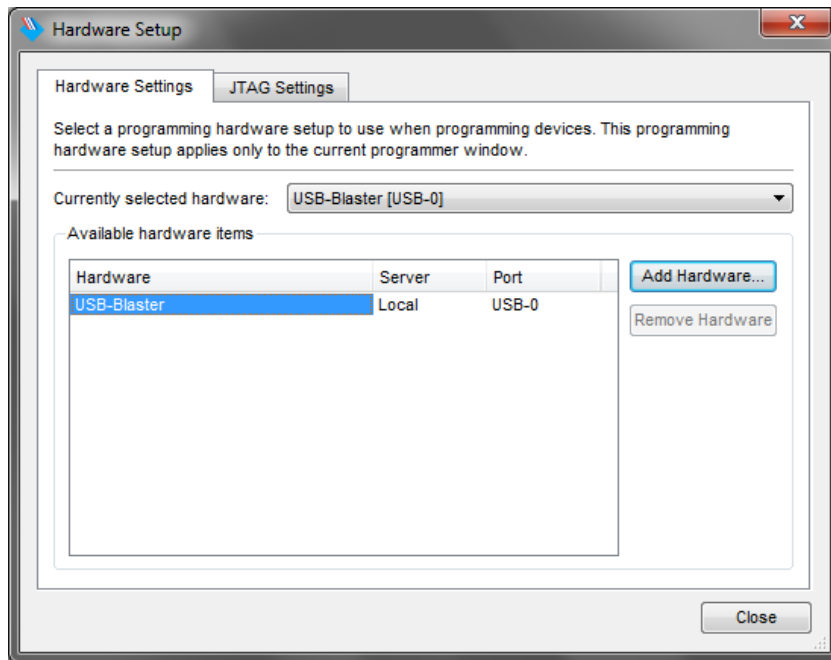


Рисунок 4.21 – Вікно вибору пристрою для програмування

У вкладці Autodetect вибираємо нашу мікросхему (рис. 4.22)

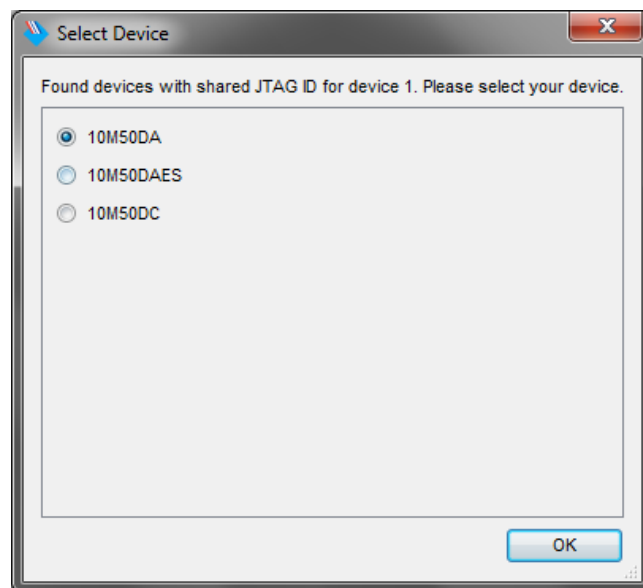


Рисунок 4.22 – Вікно вибору пристрою

Її схематичне зображення з'явиться у вікні (рис. 4.23). Натискаємо на нього, та вибираємо на панелі зліва пункт Change file, вибираємо файл прошивки з розширенням .sof

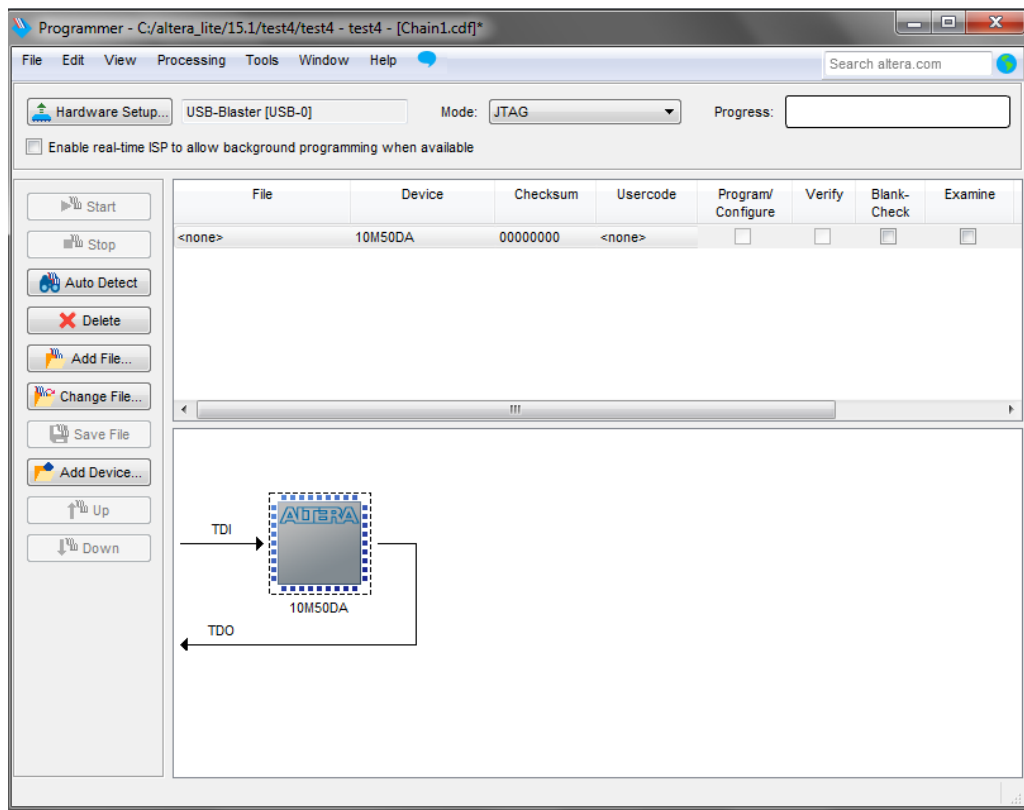


Рисунок 4.23 – Вікно параметрів програмування

Натискаємо чекбокс Program/Configure та Verify, після цього натискаємо Start. Прогрес-бар покаже процес програмування плати.

4.5 Зміст звіту

- 1) Титульний аркуш.
- 2) Мета роботи, програма роботи.
- 3) Умова роботи схем згідно варіанту.
- 4) Логічний синтез схеми керування.
- 5) Програма на мові FBD.
- 6) Таблиці для вхідних/вихідних сигналів.
- 7) Висновок про відповідність роботи заданим умовам.

4.6. Контрольні питання

1. Які основні апаратні ресурси плати?
2. Поясніть призначення тригерів Шмідта при підключенні кнопок.
3. Поясніть особливості роботи зі світлодіодами
4. Поясніть особливості роботи зі семисегментними індикаторами.
5. З яких основних блоків складається запам'ятовуючий пристрій?
6. Наведіть процедуру програмування плати.

Лабораторна робота №5

РОБОТА З СЕМИСЕГМЕНТНИМИ ІНДИКАТОРАМИ І ТАЙМЕРАМИ

Тривалість лабораторної роботи – 4 год.

Тривалість домашньої роботи – 4 год.

Мета роботи – набути вмінь синтезувати логічні схеми систем автоматизації з використанням семисегментних індикаторів, навчитися розробляти програми на мові FBD у середовищі Quartus Prime з використанням таймерів та генераторів тактових імпульсів і практично перевіряти їх за допомогою плати розробника DE10-Lite.

5.1 Основні теоретичні відомості

Функціональна схема плати показана на рис. 5.1:

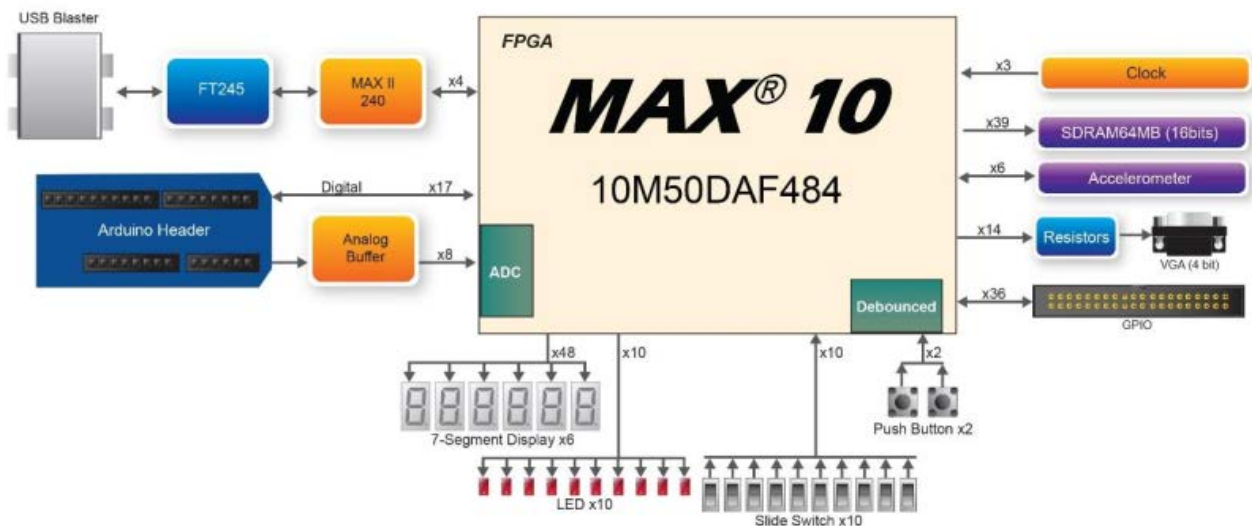


Рисунок 5.1 – Функціональна схема плати DE10-Lite

Дана плата виконана на базі ПЛІС MAX 10 10M50DAF484C7G, яка відноситься до класу програмованих користувачем вентиляльних матриць або FPGA. В даній ПЛІС міститься 50 000 програмованих логічних елементів.

До складу плати входять USB Blaster, 6 семисегментних індикаторів, 10 світлодіодів, 10 перемикачів, 2 кнопки, 3 годинника, акселерометр, SDRAM64MB, 36 входів/виходів загального призначення, Arduino Header, VGA. В даному циклі лабораторних робіт будуть використовуватися світлодіоди, семисегментні індикатори, перемикачі та годинник.

Опис кнопок

На платі DE10-Lite є дві вмонтовані кнопки, підтягнуті до плюса джерела живлення. Ці кнопки підключені до пінів B8 та A7 мікросхеми 10M50DAF484C7G через тригери Шмідта для усунення дріб'язку контактів. Схема підключення кнопок наведена на рис. 5.2

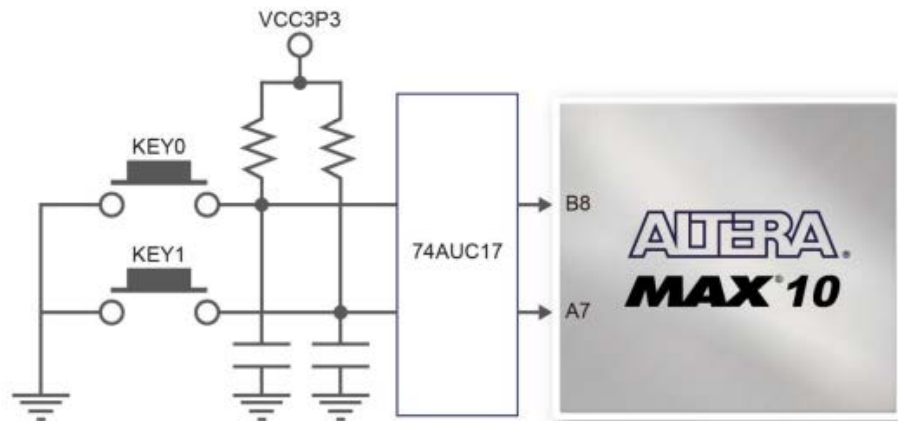


Рисунок 5.2 – Кнопки

Таблиця 5.1 – Розпіновка кнопок

Назва сигналу	№ піну FPGA	I/O стандарт
KEY0	PIN_B8	3.3 V SCHMITT TRIGGER
KEY1	PIN_A7	3.3 V SCHMITT TRIGGER

Опис перемикачів

На даній платі є 10 вмонтованих двопозиційних перемикачів. Назви сигналів та відповідна розпіновка показана у табл. 5.2, рисунок 5.3. містить функціональну схему підключення перемикачів. Важливим є той факт, що

розімкнений стан перемикача відповідає його положенню ближче до краю плати.



Рисунок 5.3 – Перемикачі

Таблиця 5.2 – Розпіновка перемикачів

Назва сигналу	№ піну FPGA	I/O стандарт
SW0	PIN_C10	3.3-V LVTTL
SW1	PIN_C11	3.3-V LVTTL
SW2	PIN_D12	3.3-V LVTTL
SW3	PIN_C12	3.3-V LVTTL
SW4	PIN_A12	3.3-V LVTTL
SW5	PIN_B12	3.3-V LVTTL
SW6	PIN_A13	3.3-V LVTTL
SW7	PIN_A14	3.3-V LVTTL
SW8	PIN_B14	3.3-V LVTTL
SW9	PIN_F15	3.3-V LVTTL

Опис світлодіодів

На платі є 10 вмонтованих світлодіодів, які підтягнуті до плюса джерела живлення. Ці світлодіоди керуються одиницями і у випадку коли вони не використовуються на відповідні піни рекомендується подавати логічні нулі, оскільки у протилежному випадку світлодіоди будуть світитися у пів-накалу.

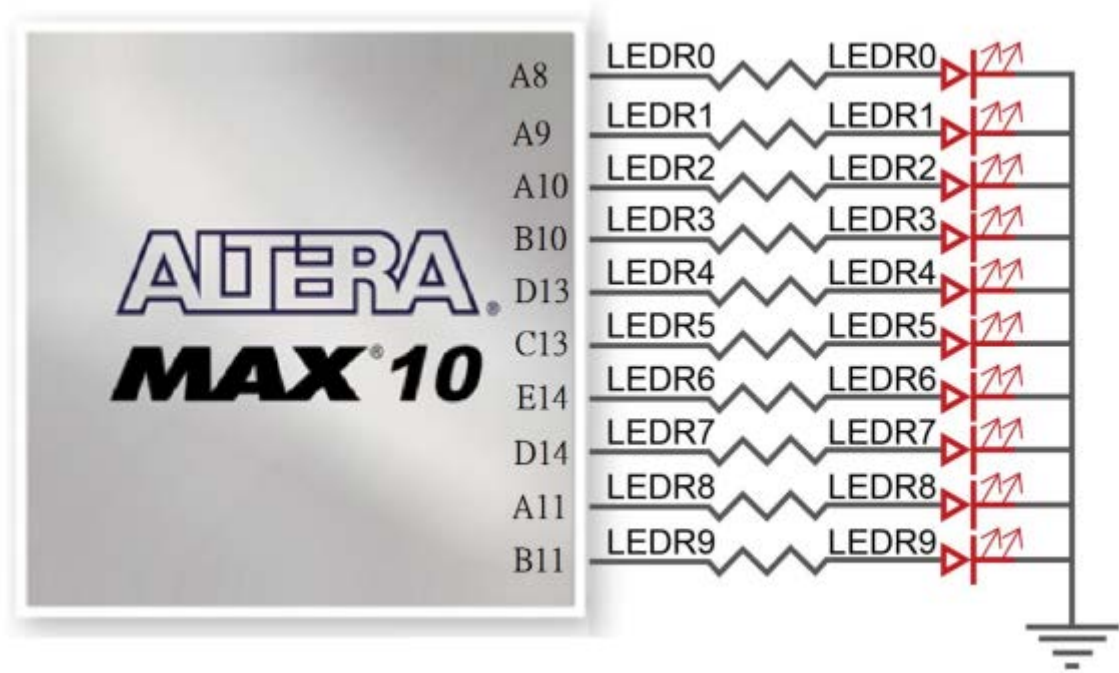


Рисунок 5.4 – Світлодіоди

Таблиця 5.3 – Розпіновка світлодіодів

Назва сигналу	№ піну FPGA	I/O стандарт
LEDR0	PIN_A8	3.3-V LVTTL
LEDR1	PIN_A9	3.3-V LVTTL
LEDR2	PIN_A10	3.3-V LVTTL
LEDR3	PIN_B10	3.3-V LVTTL
LEDR4	PIN_D13	3.3-V LVTTL
LEDR5	PIN_C13	3.3-V LVTTL
LEDR6	PIN_E14	3.3-V LVTTL
LEDR7	PIN_D14	3.3-V LVTTL
LEDR8	PIN_A11	3.3-V LVTTL
LEDR9	PIN_B11	3.3-V LVTTL

Опис семисегментних індикаторів

На платі є 6 вмонтованих семисегментних індикаторів, які керуються нулями.

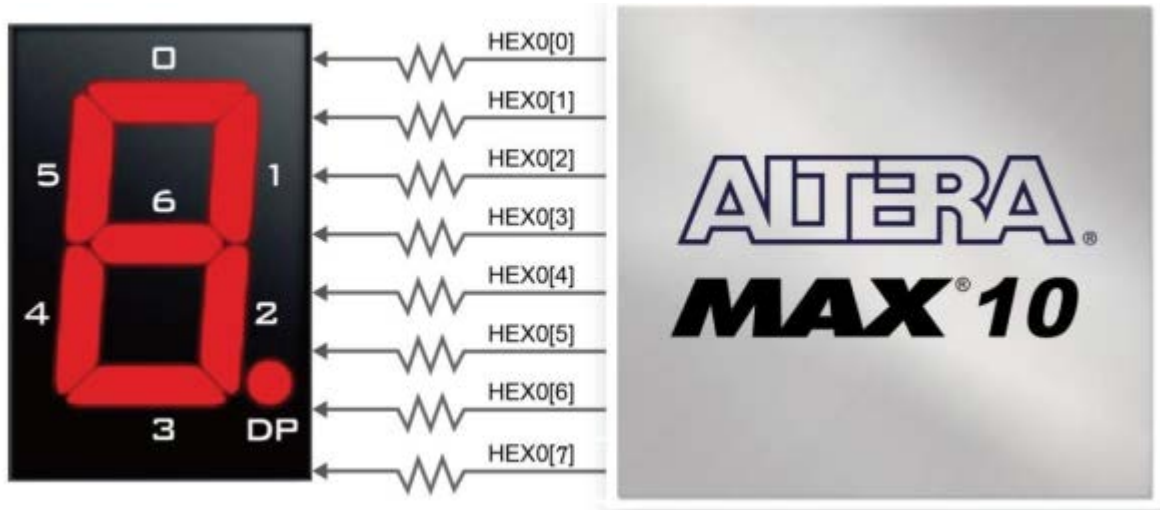


Рисунок 5.5 – Семисегментний індикатор

Таблиця 5.4 – Розпіновка групи семисегментних індикаторів

Назва сигналу	№ піну FPGA	I/O стандарт
HEX00	PIN_C14	3.3-V LVTTL
HEX01	PIN_E15	3.3-V LVTTL
HEX02	PIN_C15	3.3-V LVTTL
HEX03	PIN_C16	3.3-V LVTTL
HEX04	PIN_E16	3.3-V LVTTL
HEX05	PIN_D17	3.3-V LVTTL
HEX06	PIN_C17	3.3-V LVTTL
HEX07	PIN_D15	3.3-V LVTTL
HEX10	PIN_C18	3.3-V LVTTL
HEX11	PIN_D18	3.3-V LVTTL
HEX12	PIN_E18	3.3-V LVTTL
HEX13	PIN_B16	3.3-V LVTTL
HEX14	PIN_A17	3.3-V LVTTL
HEX15	PIN_A18	3.3-V LVTTL
HEX16	PIN_B17	3.3-V LVTTL
HEX17	PIN_A16	3.3-V LVTTL
HEX20	PIN_B20	3.3-V LVTTL
HEX21	PIN_A20	3.3-V LVTTL
HEX22	PIN_B19	3.3-V LVTTL
HEX23	PIN_A21	3.3-V LVTTL
HEX24	PIN_B21	3.3-V LVTTL

HEX25	PIN_C22	3.3-V LVTTL
HEX26	PIN_B22	3.3-V LVTTL
HEX27	PIN_A19	3.3-V LVTTL
HEX30	PIN_F21	3.3-V LVTTL
HEX31	PIN_E22	3.3-V LVTTL
HEX32	PIN_E21	3.3-V LVTTL
HEX33	PIN_C19	3.3-V LVTTL
HEX34	PIN_C20	3.3-V LVTTL
HEX35	PIN_D19	3.3-V LVTTL
HEX36	PIN_E17	3.3-V LVTTL
HEX37	PIN_D22	3.3-V LVTTL
HEX40	PIN_F18	3.3-V LVTTL
HEX41	PIN_E20	3.3-V LVTTL
HEX42	PIN_E19	3.3-V LVTTL
HEX43	PIN_J18	3.3-V LVTTL
HEX44	PIN_H19	3.3-V LVTTL
HEX45	PIN_F19	3.3-V LVTTL
HEX46	PIN_F20	3.3-V LVTTL
HEX47	PIN_F17	3.3-V LVTTL
HEX50	PIN_J20	3.3-V LVTTL
HEX51	PIN_K20	3.3-V LVTTL
HEX52	PIN_L18	3.3-V LVTTL
HEX53	PIN_N18	3.3-V LVTTL
HEX54	PIN_M20	3.3-V LVTTL
HEX55	PIN_N19	3.3-V LVTTL
HEX56	PIN_N20	3.3-V LVTTL
HEX57	PIN_L19	3.3-V LVTTL

Опис годинника

У плату вбудовано годинник (генератор імпульсів фіксованої частоти).

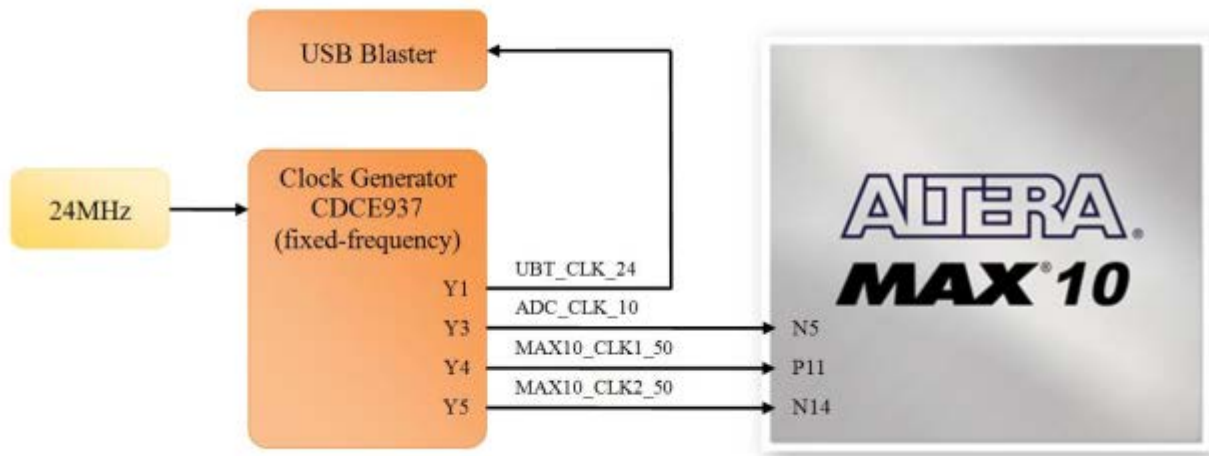


Рисунок 5.6 – Схема підключення годинника

Увага: не змінюйте налаштувань годинника – це може призвести до некоректної роботи плати!

Таблиця 5.5 – Розпіновка годинника

Назва сигналу	№ піну FPGA	Опис	I/O стандарт
ADC_CLK_10	PIN_N5	10 МГц годинник для ADC	3.3-V LVTTL
MAX10_CLK1_50	PIN_P11	50 МГц годинник	3.3-V LVTTL
MAX10_CLK2_50	PIN_N14	50 МГц годинник	3.3-V LVTTL

5.2 Програма роботи

1. Ознайомитися з функціональною схемою плати та чітко розуміти призначення кожного блоку системи.
2. За таблицею 5.6 обрати свій варіант.
3. Виконати логічний синтез завдань за вказаним методом.
4. У програмному середовищі Quartus Prime скласти на мові FBD програму, яка вирішує поставлене завдання.
5. Перевірити правильність виконання завдання.
6. Оформити протокол по лабораторній роботі та зробити висновки.

Номер бригади	Номер завдання
1	1, 7, 13
2	2, 8, 14
3	3, 9, 15
4	4, 10, 16
5	5, 11, 17
6	6, 12, 18

5.3 Завдання до лабораторної роботи

Виконати синтез на логічних елементах:

1. Схема додає 2 3-бітні числа і виводить суму у вигляді десяткового числа на 2 семисегментні індикатори. Результат виводиться по натисненню кнопки. Числа задаються у вигляді двійкового коду перемикачами.

2. Схема віднімає 2 3-бітні числа і виводить різницю у вигляді десяткового числа на 2 семисегментні індикатори. Результат виводиться по натисненню кнопки. Числа задаються у вигляді двійкового коду перемикачами.

3. Схема додає 2 4-бітні числа і виводить суму у вигляді десяткового числа на 2 семисегментні індикатори (сума не більше 15). Результат виводиться по натисненню кнопки. Числа задаються у вигляді двійкового коду перемикачами.

4. Схема віднімає 2 4-бітні числа і виводить різницю у вигляді десяткового числа на 2 семисегментні індикатори. Результат виводиться по натисненню кнопки. Числа задаються у вигляді двійкового коду перемикачами.

5. Схема додає 2 5-бітні числа і виводить 1 на семисегментний індикатор, якщо результат знаходиться в проміжку між 25 і 30 включно, в іншому випадку виводиться 0. Результат виводиться по натисненню кнопки. Числа задаються у вигляді двійкового коду перемикачами.

6. Схема віднімає 2 5-бітні числа і виводить 1 на семисегментний індикатор, якщо результат знаходиться в проміжку між 10 і 15 включно, в

іншому випадку виводиться 0. Результат виводиться по натисненню кнопки. Числа задаються у вигляді двійкового коду перемикачами.

Виконати синтез на мультиплексорах-селекторах:

7. Схема віднімає 2 4-бітні числа і виводить 1 на семисегментний індикатор, якщо різниця більша 10 або менша 3, в інших випадках виводиться 0. Числа задаються у вигляді двійкового коду перемикачами. Передбачити виведення результату обчислень на два інші семисегментні індикатори у вигляді десяткового числа і відділити його від результату операції порівняння точкою.

8. Схема додає 2 4-бітні числа і виводить 1 на семисегментний індикатор, якщо сума належить проміжку $[3;10] \cup [17;25]$, в інших випадках виводиться 0. Числа задаються у вигляді двійкового коду перемикачами. Передбачити виведення результату обчислень на два інші семисегментні індикатори у вигляді десяткового числа і відділити його від результату операції порівняння точкою.

9. Схема віднімає 2 5-бітні числа і виводить 1 на семисегментний індикатор, якщо різниця ділиться на 3, в інших випадках виводиться 0. Числа задаються у вигляді двійкового коду перемикачами. Передбачити виведення результату обчислень на два інші семисегментні індикатори у вигляді десяткового числа і відділити його від результату операції порівняння точкою.

10. Схема віднімає 2 5-бітні числа і виводить 1 на семисегментний індикатор, якщо різниця ділиться на 4, в інших випадках виводиться 0. Числа задаються у вигляді двійкового коду перемикачами. Передбачити виведення результату обчислень на два інші семисегментні індикатори у вигляді десяткового числа і відділити його від результату операції порівняння точкою.

11. Схема додає 2 4-бітні числа і виводить 1 на семисегментний індикатор, якщо сума ділиться на 6 або 7, в інших випадках виводиться 0. Числа задаються у вигляді двійкового коду перемикачами. Передбачити виведення результату обчислень на два інші семисегментні індикатори у вигляді десяткового числа і відділити його від результату операції порівняння точкою.

12. Схема додає 3 3-бітні числа і виводить 1 на семисегментний індикатор, якщо сума ділиться на 3 або 7, в інших випадках виводиться 0. Числа задаються у вигляді двійкового коду перемикачами. Передбачити виведення результату обчислень на два інші семисегментні індикатори у вигляді десяткового числа і відділити його від результату операції порівняння точкою.

Виконати синтез на тригерах:

13. Схема працює наступним чином: при натисненні кнопки Пуск на семисегментному індикаторі засвічується 1, яка через 1 секунду перемикається на 2, потім - на 3, після чого знову на 1. Якщо кнопку Пуск відтиснути схема не переходить у наступний стан. При натисненні кнопки Початок роботи схема переходить у відповідно 1, 2 або третій режим роботи – в залежності від того, яка цифра була засвічена на семисегментному індикаторі.

Режими:

- 1) Мигання світлодіодом.
- 2) Засвічення літери E на іншому семисегментному індикаторі.
- 3) Засвічення літери H на іншому семисегментному індикаторі.

При натисненні кнопки Назад схема повертається до вибору режиму в вершину, яка відповідає режиму, в якому вона працювала. Схема повністю припиняє роботу при натисненні кнопки Стоп.

14. Схема працює наступним чином: при натисненні кнопки Пуск на семисегментному індикаторі засвічується А, яка через 1.5 секунди перемикається на В, потім - на С, після чого знову на А. Якщо кнопку Пуск відтиснути схема не переходить у наступний стан. При натисненні кнопки Початок роботи схема переходить у відповідно у режим А,В або С – в залежності від того, яка літера була засвічена на семисегментному індикаторі.

Режими:

- 1) Мигання цифра 1 на іншому семисегментному індикаторі.
- 2) Мигання цифра 0 на іншому семисегментному індикаторі.
- 3) Засвічення 6 сегменту на іншому семисегментному індикаторі.

При натисненні кнопки Назад схема повертається до вибору режиму в вершину, яка відповідає режиму, в якому вона працювала. Схема повністю припиняє роботу при натисненні кнопки Стоп.

15. Схема працює наступним чином: при натисненні кнопки Пуск на семисегментному індикаторі засвічується **t**, яка через 1.5 секунди перемикається на **t.**, потім - на **Е**, після чого знову на **t**. Якщо кнопку Пуск відтиснути схема не переходить у наступний стан. При натисненні кнопки Початок роботи схема переходить у відповідно у режим **t,t.** або **Е** – в залежності від того, яка літера була засвічена на семисегментному індикаторі.

Режими:

- 1) Засвічення сегментів 0-1-6-5-0-... із затримкою 0.5 с. в безкінечному циклі на іншому семисегментному індикаторі.
- 2) Засвічення сегментів 0-5-6-1-0-... із затримкою 0.5 с. в безкінечному циклі на іншому семисегментному індикаторі.
- 3) Засвічення 1 світлодіоду на 2 с., потім 2 світлодіоду на 1 с. і повернення до вибору режиму.

При натисненні кнопки Назад схема повертається до вибору режиму в вершину, яка відповідає режиму, в якому вона працювала. Схема повністю припиняє роботу при натисненні кнопки Стоп.

16. Схема працює наступним чином: при натисненні кнопки Пуск на семисегментному індикаторі засвічується **t**, яка через 2 секунди перемикається на **t.**, потім - на **Е**, після чого знову на **t**. Якщо кнопку Пуск відтиснути схема не переходить у наступний стан. При натисненні кнопки Початок роботи схема переходить у відповідно у режим **t,t.** або **Е** – в залежності від того, яка літера була засвічена на семисегментному індикаторі.

Режими:

- 1) Засвічення сегментів 6-2-3-4-6-... із затримкою 0.7 с. в безкінечному циклі на іншому семисегментному індикаторі.
- 2) Засвічення сегментів 6-4-3-2-6-... із затримкою 0.7 с. в безкінечному циклі на іншому семисегментному індикаторі.
- 3) Засвічення літери Е на іншому семисегментному індикаторі.

При натисненні кнопки Назад схема повертається до вибору режиму в вершину, яка відповідає режиму, в якому вона працювала. Схема повністю припиняє роботу при натисненні кнопки Стоп.

17. Схема працює наступним чином: при натисненні кнопки Пуск на семисегментному індикаторі засвічується А, яка через 2.5 секунди перемикається на D, потім - на E, після чого знову на А. Якщо кнопку Пуск відтиснути схема не переходить у наступний стан. При натисненні кнопки Початок роботи схема переходить у відповідно у режим А, D або E – в залежності від того, яка літера була засвічена на семисегментному індикаторі.

Режими:

- 1) Засвічення першого світлодіоду на 2 с. і повернення до вибору режиму.
- 2) Засвічення другого світлодіоду на 1.5 с. і повернення до вибору режиму.
- 3) Мигання літерою E на іншому семисегментному індикаторі.

При натисненні кнопки Назад схема повертається до вибору режиму в вершину, яка відповідає режиму, в якому вона працювала. Схема повністю припиняє роботу при натисненні кнопки Стоп.

18. Схема працює наступним чином: при натисненні кнопки Пуск (під час роботи кнопка не натиснута) на семисегментному індикаторі засвічується цифра 0, яка через 0.5 секунди перемикається на 1, потім - на 2 і так до 5 після чого знову на 0. Якщо під час роботи схеми натиснути кнопку Старт, то засвічуються світлодіоди за наступним правилом: 0 – LED1; 1,2 – LED2; 3,4 – LED3; 5 – LED4.

При натисненні кнопки Стоп система повертається в початкове положення.

5.4 Методичні вказівки до виконання лабораторної роботи

1) Для виконання першого завдання (1-6) доцільно скористатися схемою повного суматора і заміни віднімання на додавання. В таблиці 5.7 наведена таблиця істинності для однобітового суматора.

Таблиця 5.7 – Таблица істинності однобітного суматора

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Для побудови суматорів на більшу кількість біт використовують схему з'єднання однобітових суматорів, наведену на рис. 5.7:

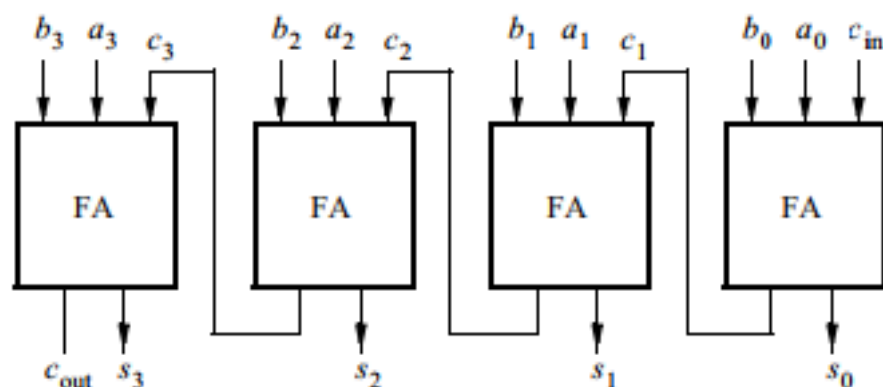


Рисунок 5.7 – Чотирибітний суматор

2) Також слід зважати на те, що логічна одиниця подана на семисегментний індикатор гасить відповідний сегмент, а нуль запалює.

3) Для заміни віднімання на додавання від'ємник перетворюють у додатковий код наступним чином: беруть інверсію від числа і до даної інверсії додають одиницю. Наприклад: додатковий код для числа 1011 – $0100+0001=0101$.

При відніманні з використанням додаткового коду слід відкидати старший розряд різниці. Наприклад: $14-11=1110-1011=1110+0101=10011$. Як бачимо, якщо відкинути старший розряд то відповідь правильна.

4) Для виведення цифр на семисегментні індикатори в 2 завданні (7-12) доцільно скористатися перетворенням двійкового коду в двійково-десятковий. У двійково-десятковому коді кожному розряду десяткового числа відповідає 4-бітне число у двійковому коді. Наприклад: 71-0111 0001. Це досить сильно спрощує виведення цифр на семисегментний індикатор. Схема для перетворення коду і таблиця істинності представлені на рис. 5.8 та 5.9.

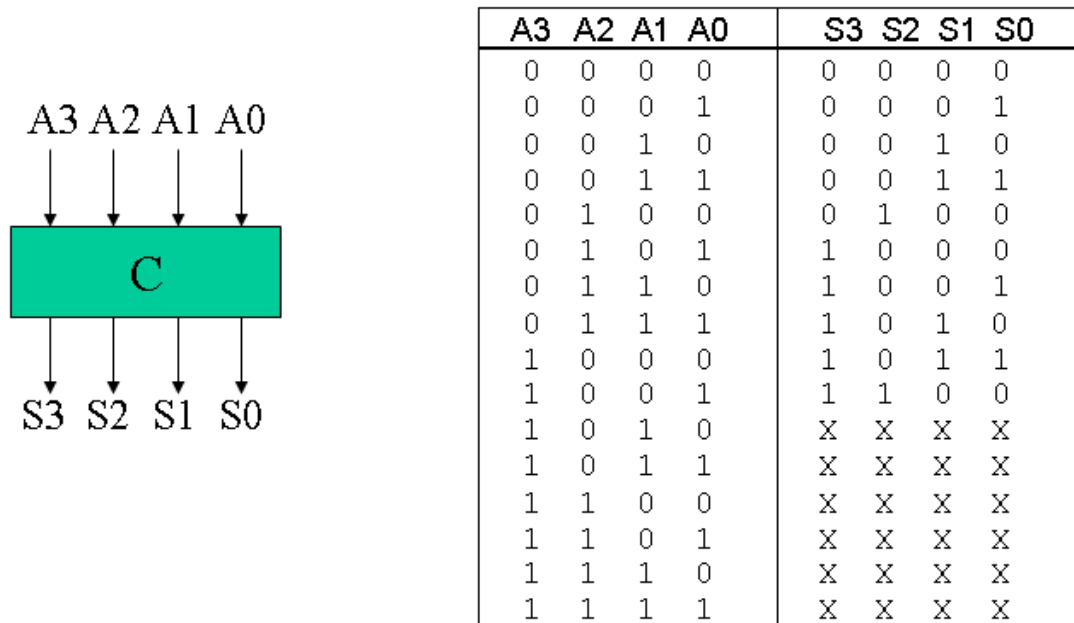


Рисунок 5.8 – Ланка для перетворення коду

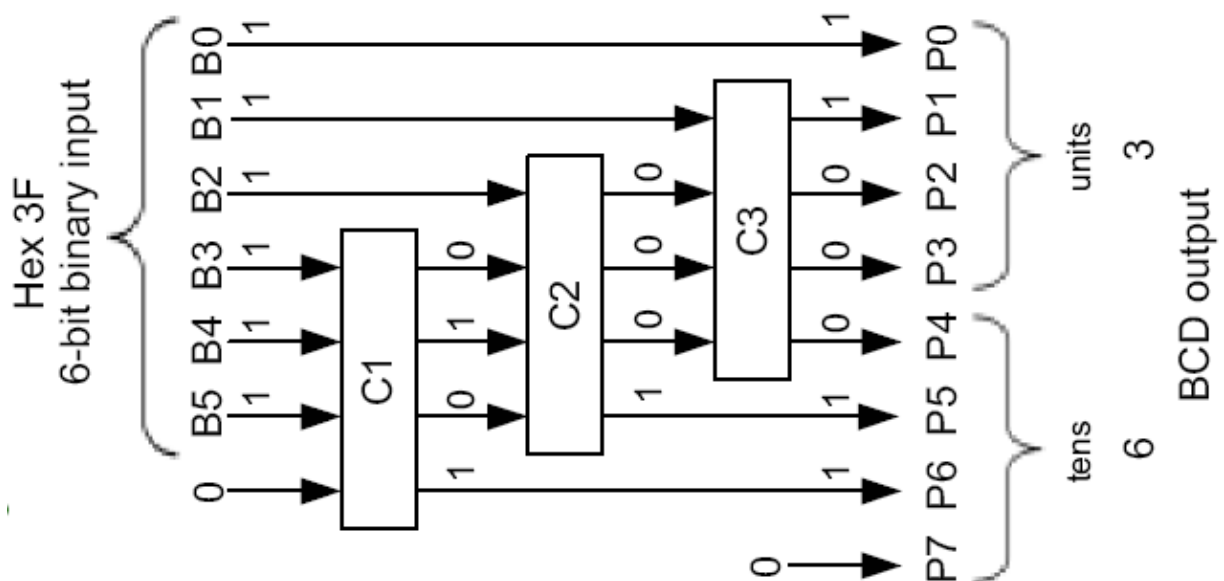


Рисунок 5.9 – Схема для перетворення

В схемі ланки деякі вихідні сигнали позначені хрестиками – це означає, що не важливо, який там буде сигнал схема все одно працюватиме правильно. На картах Карно ці хрестики позначають як тире і або включають, або не включають в контури.

5) Для виконання 3 завдання (13-18) необхідно створити таймер. Таймер створюється за допомогою перетворювача частоти, лічильника імпульсів, годинника і RS-тригера.

Перетворювач частоти – перетворює частоту 50 МГц в частоту 10кГц.

Щоб створити даний перетворювач обираємо функцію ATPLL:

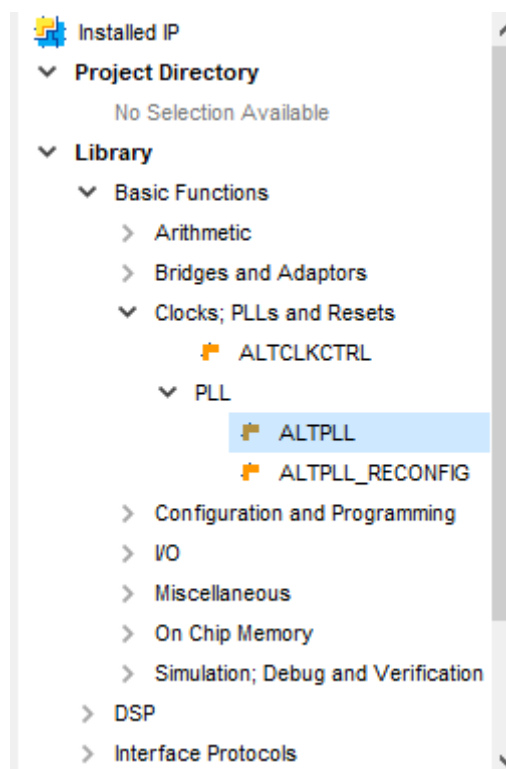


Рисунок 5.10 – Перетворювач частоти

Обираємо довільну назву і переходимо в наступне вікно:

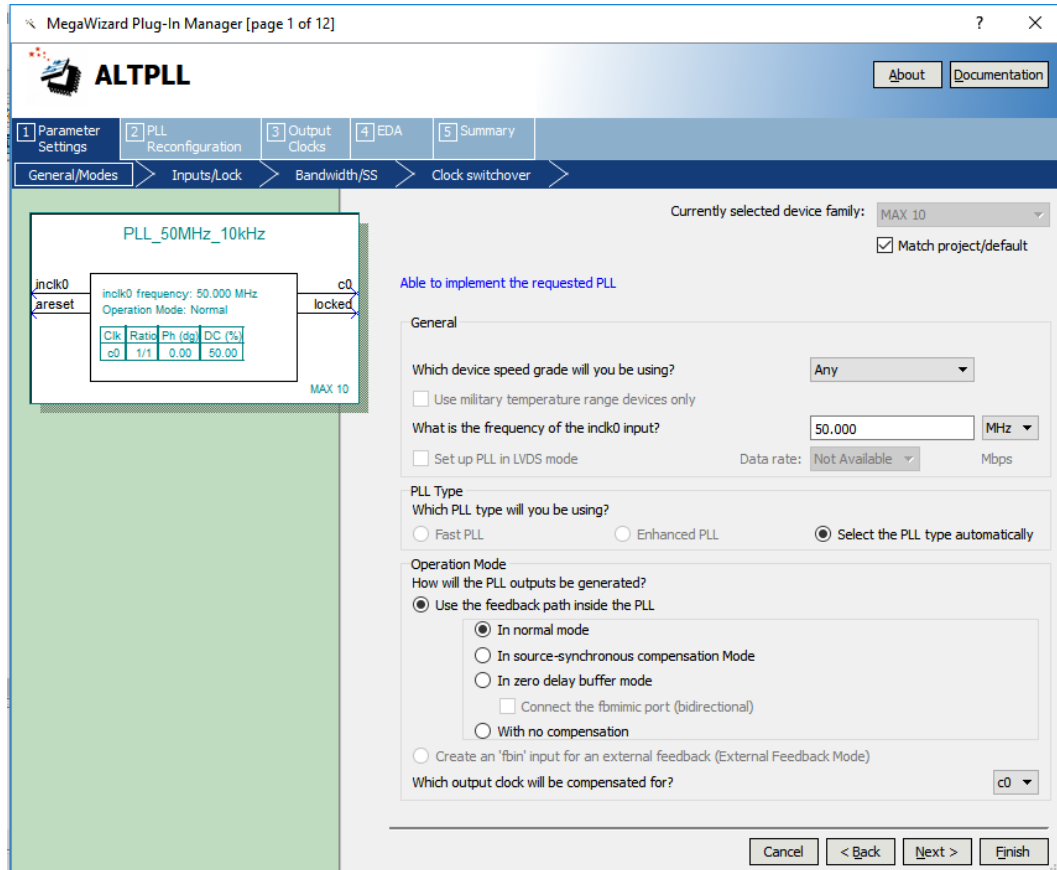


Рисунок 5.11 – Налаштування модуля ALTPLL (початок)

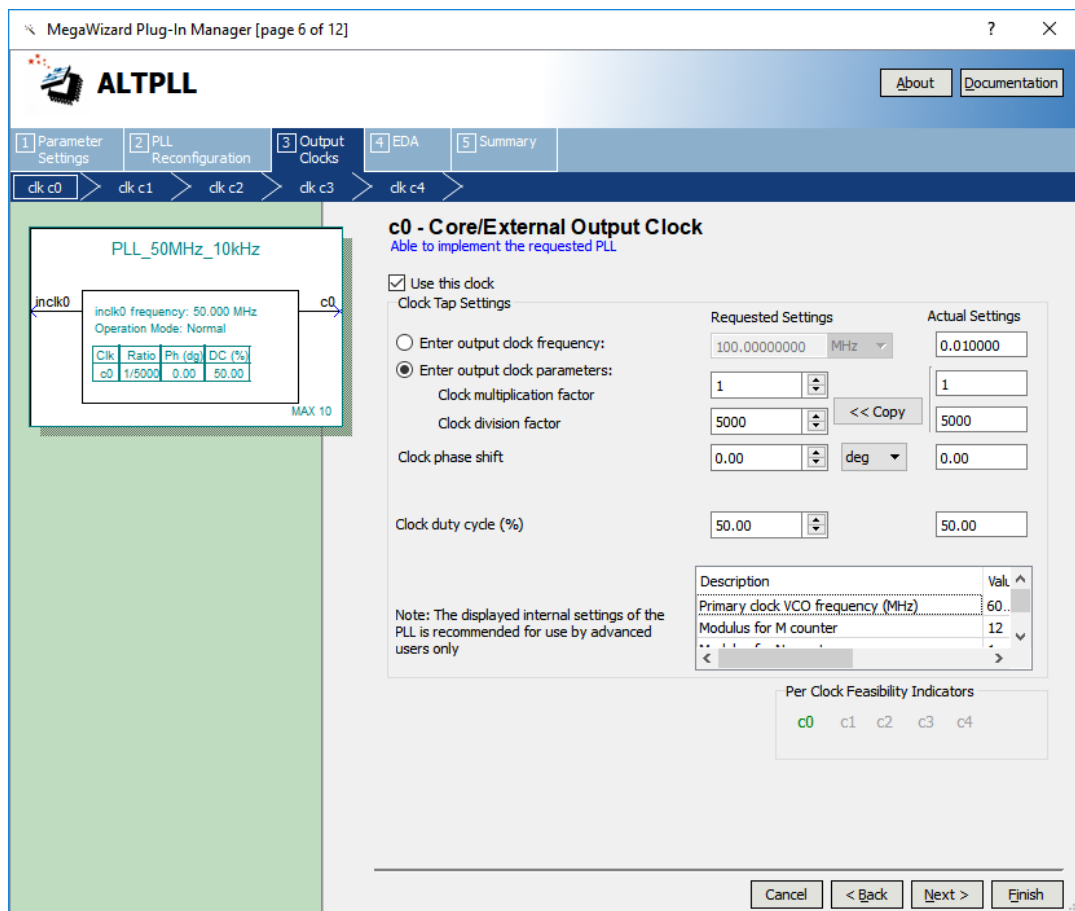


Рисунок 5.12 – Налаштування тактового сигналу модуля ALTPLL

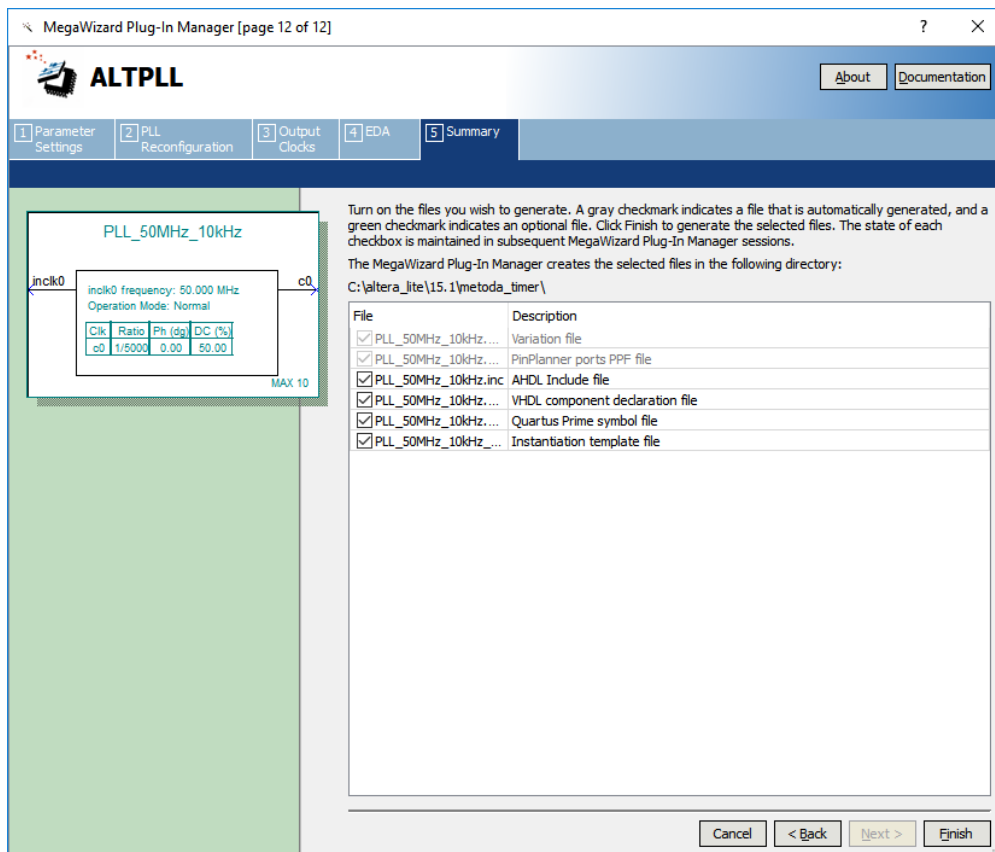


Рисунок 5.13 – Завершення налаштування модуля ALTPLL

Файл перетворювача буде в бібліотеці Symbol/Project.

Далі створюємо RS-тригер і зберігаємо його як Symbol File.

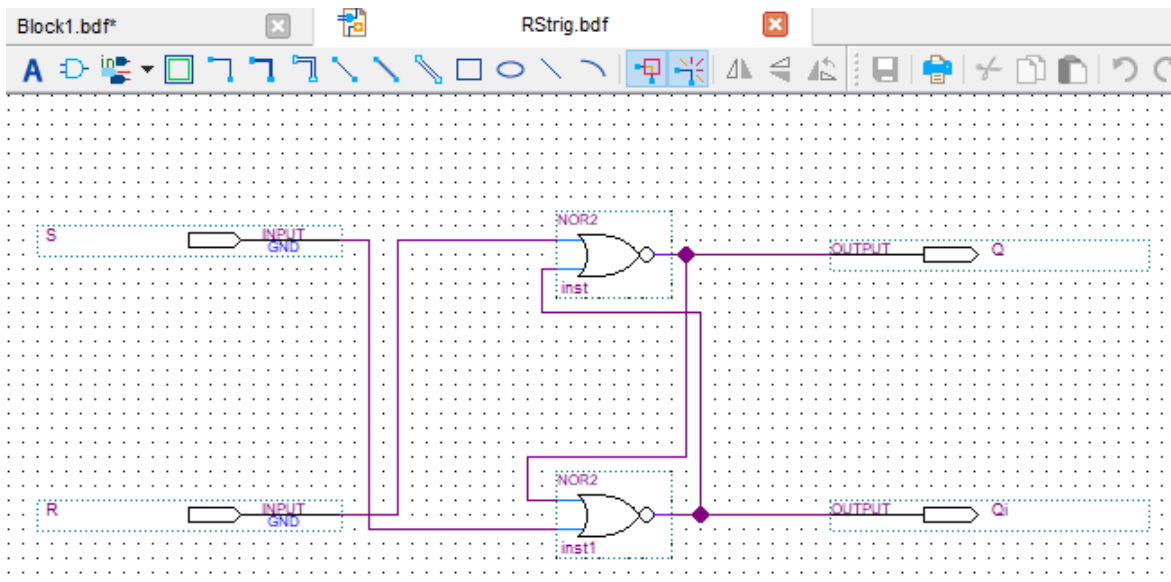


Рисунок 5.14 – RS-тригер

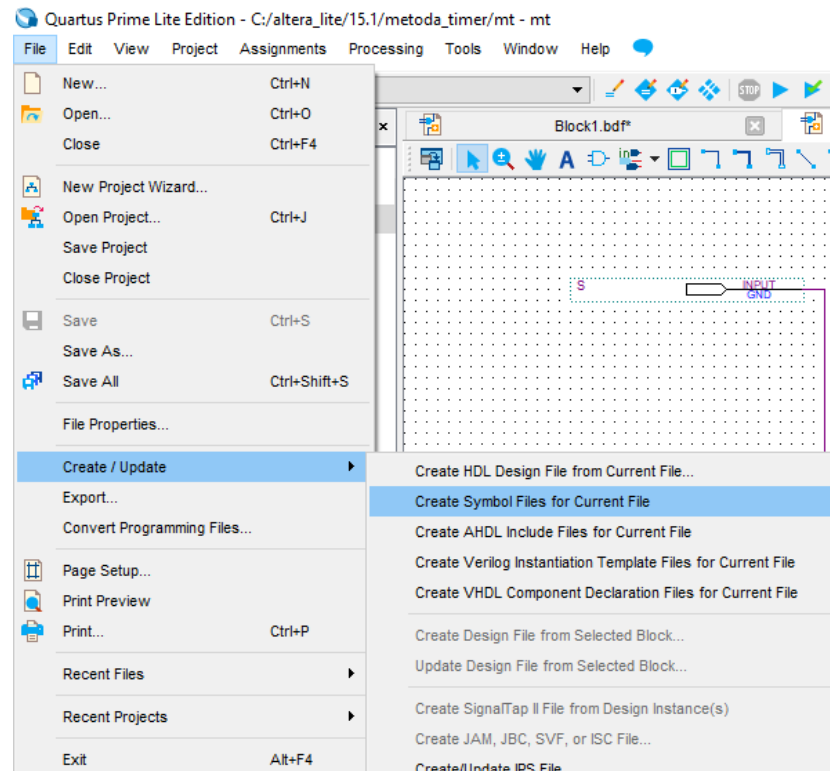


Рисунок 5.15 – Зберігання файлу

Потім створюємо лічильник імпульсів. Обираємо:

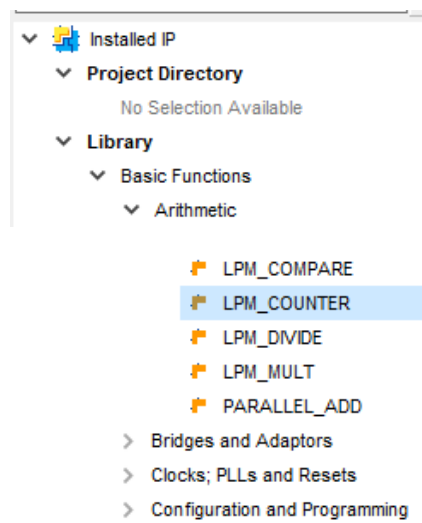


Рисунок 5.16 – Лічильник

Обираємо назву:

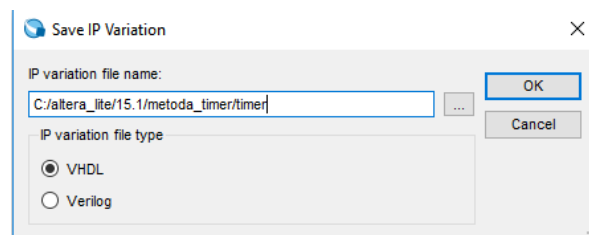


Рисунок 5.17 – Назва файлу лічильника

Переводимо необхідну кількість імпульсів у двійковий код і визначаємо кількість розрядів. В нашому випадку число 14-бітне (визначається по старшому розряду):

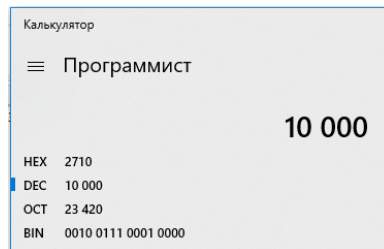


Рисунок 5.18 – Перевод числа у двійкову систему

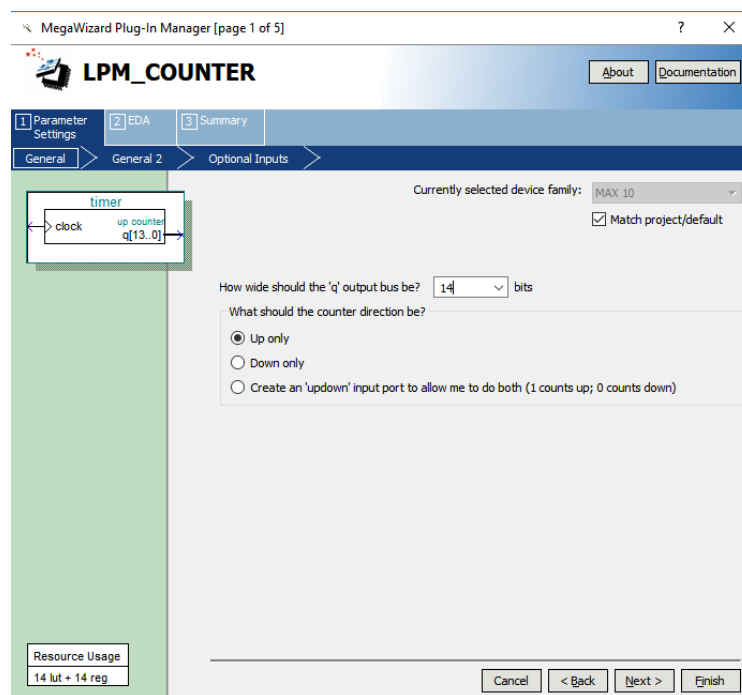


Рисунок 5.19 – Вибір кількості розрядів

Потім задаємо модуль, до якого буде вестися рахунок. Після досягнення даного числа лічильник обнуляється. Також задаємо вхід дозволу відліку (cnt_en) і сигнал переповнення (cout). Якщо на вході дозволу відліку 1, то відлік відбувається, якщо нуль – припиняється, але лічильник не скидається. Сигнал переповнення рівний одиниці, коли лічильник досяг числа на 1 меншого за модуль.

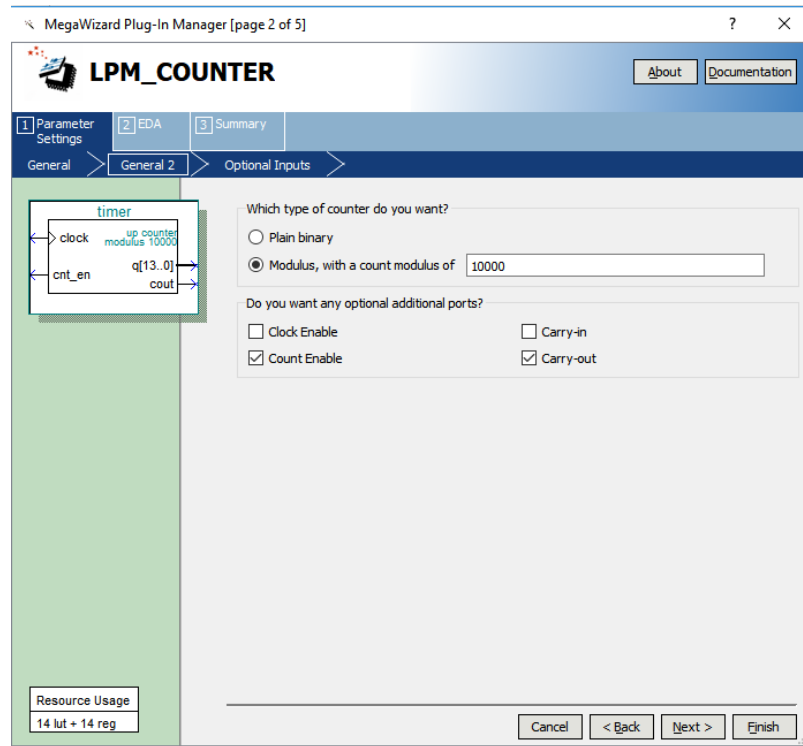


Рисунок 5.20 – Модуль

Створюємо асинхронний вхід скидання:

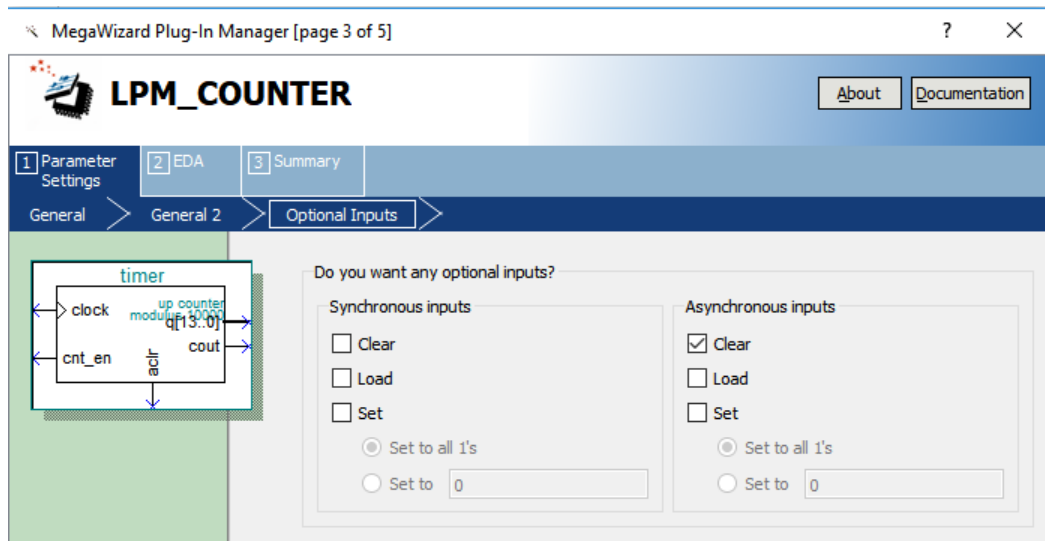


Рисунок 5.21 – Асинхронний вхід скидання

Завершуємо створення:

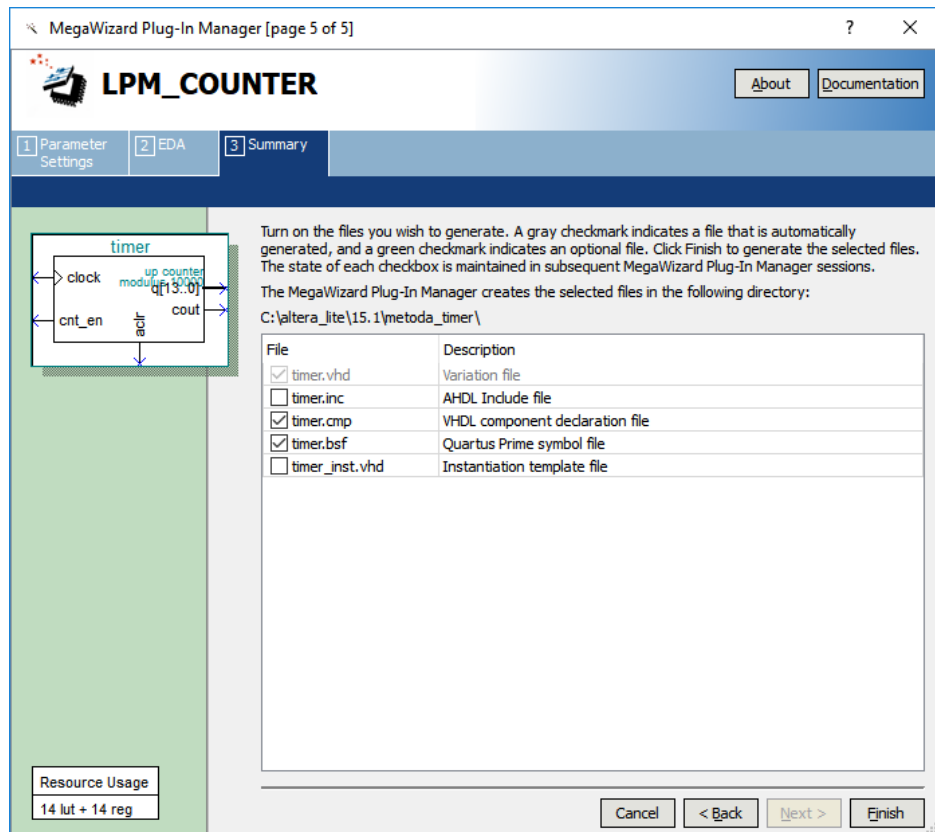


Рисунок 5.22 – Результати створення блоку

В результаті отримуємо такий блок:

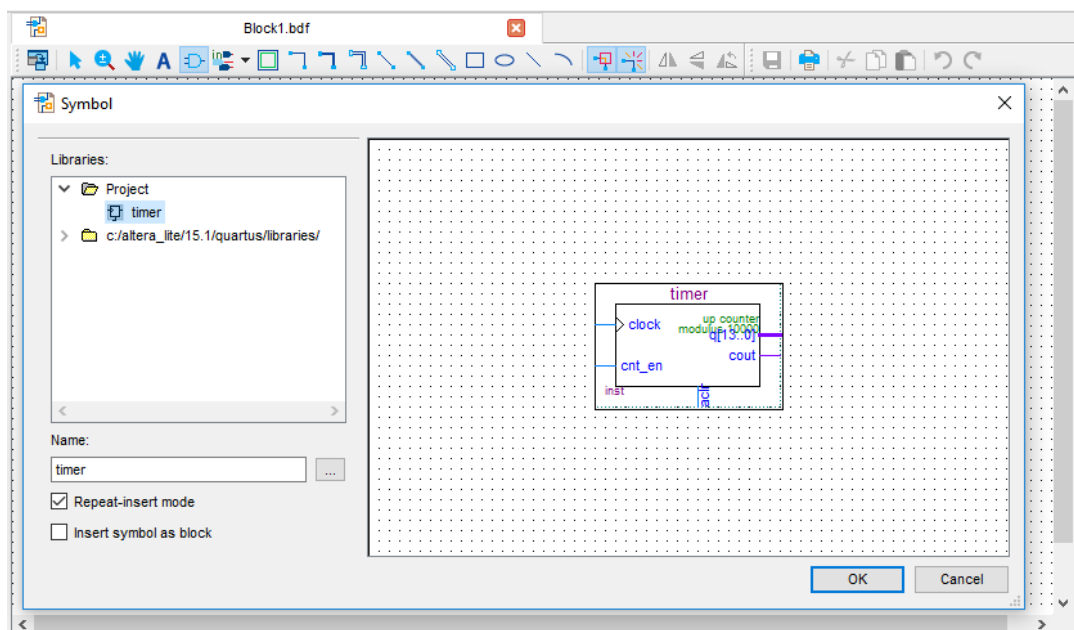


Рисунок 5.23 – Створений FBD блок

Створюємо таймер TON:

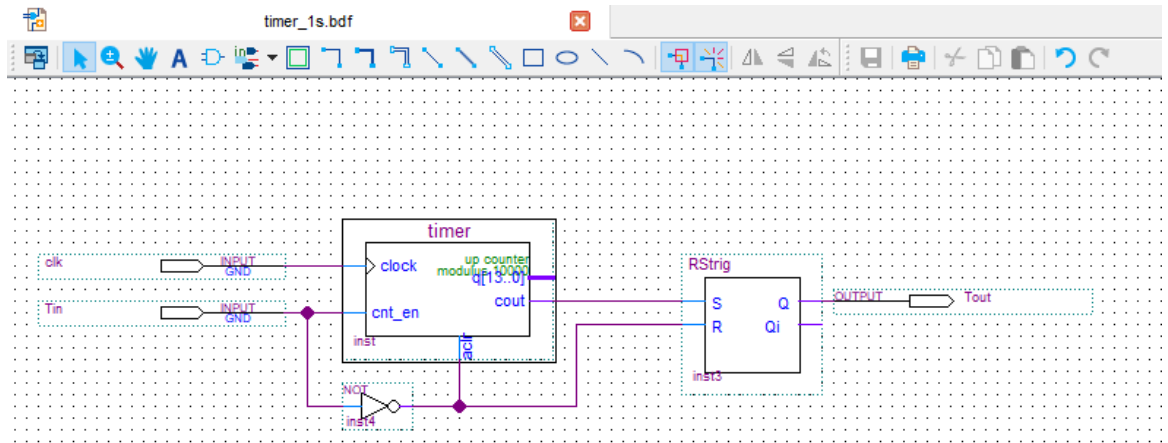


Рисунок 5.24 – Таймер

Зберігаємо як Symbol File.

Кінцева схема підключення таймера:

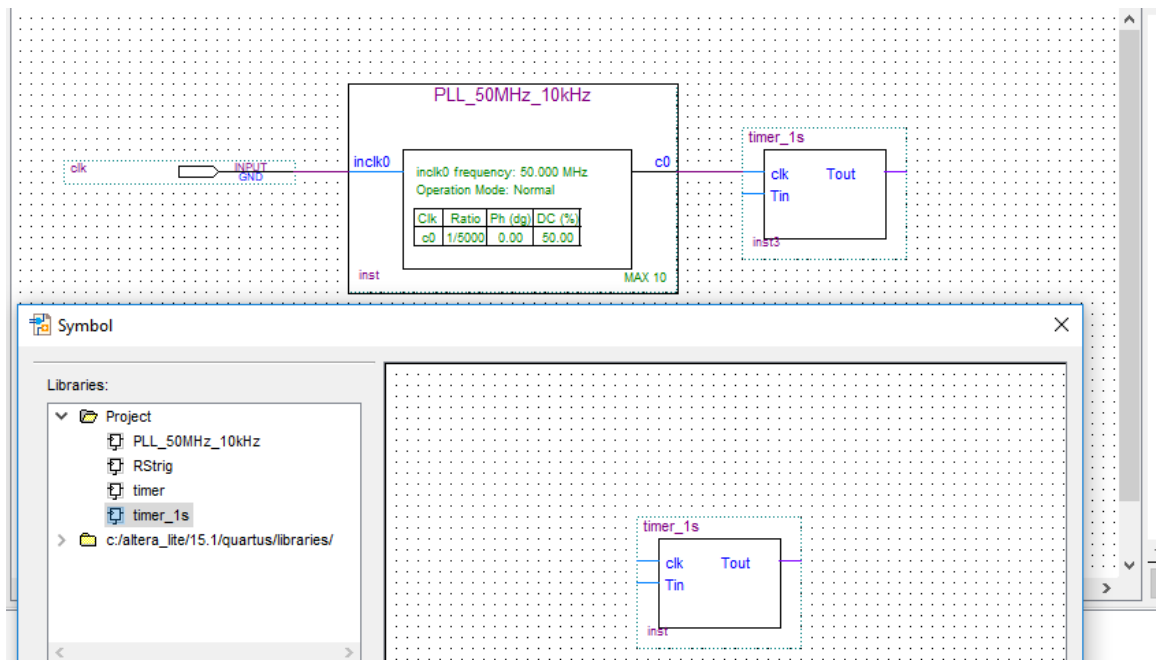


Рисунок 5.25 – Підключений таймер

Сік під'єднуємо до піна з годинником на 50 МГц, а на вхід Tin подаємо сигнал увімкнення таймера.

Для прикладу розглянемо наступну задачу

Приклад 1. Створити програму, яка віднімає два 4-бітних числа і видає 1 на семисегментний індикатор, якщо різниця знаходиться у проміжку від 2 до 10 включно.

За таблицею істинності складаємо одинітний суматор:

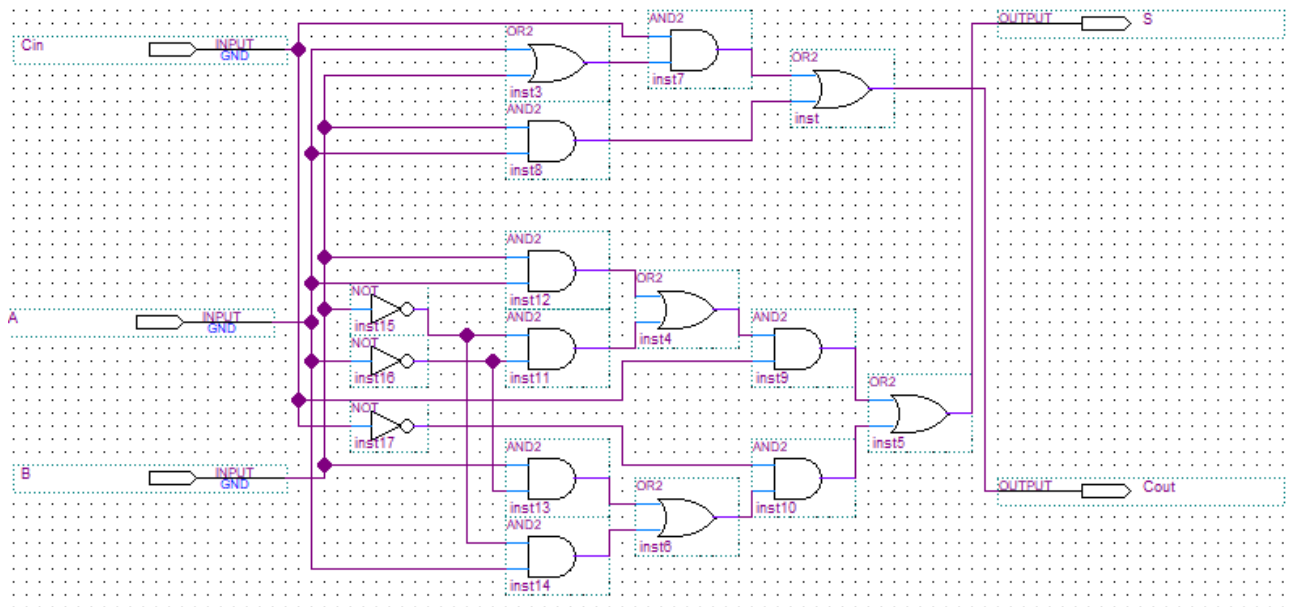


Рисунок 5.26 – Принципова схема однобітного суматора

Зберігаємо як Symbol File:

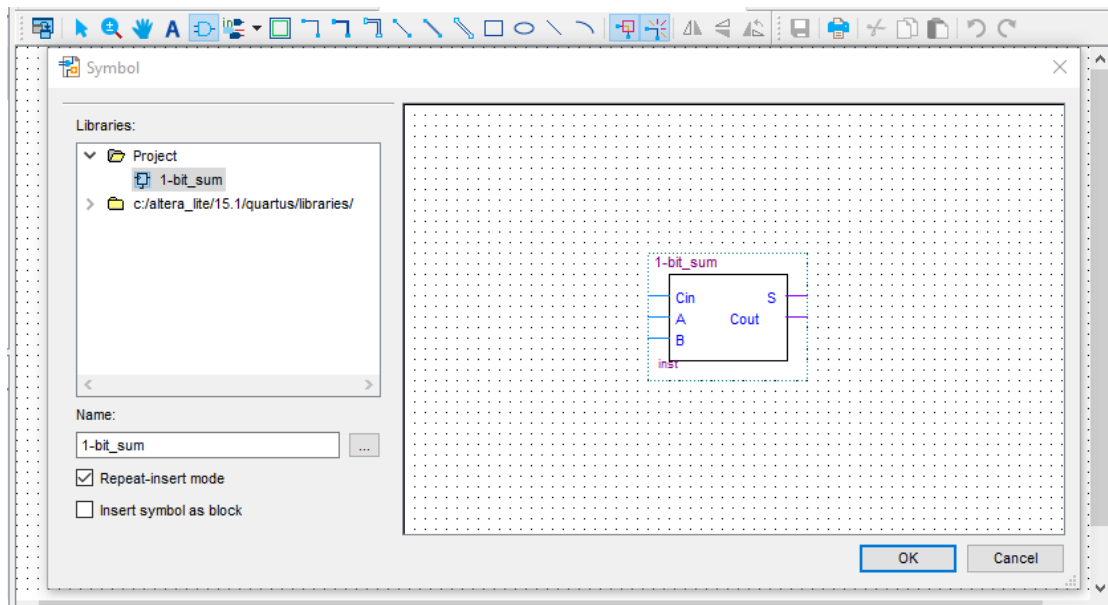


Рисунок 5.27 – FBD-блок “Однобітний суматор”

З’єднуємо однобітні суматори в 4-бітний:

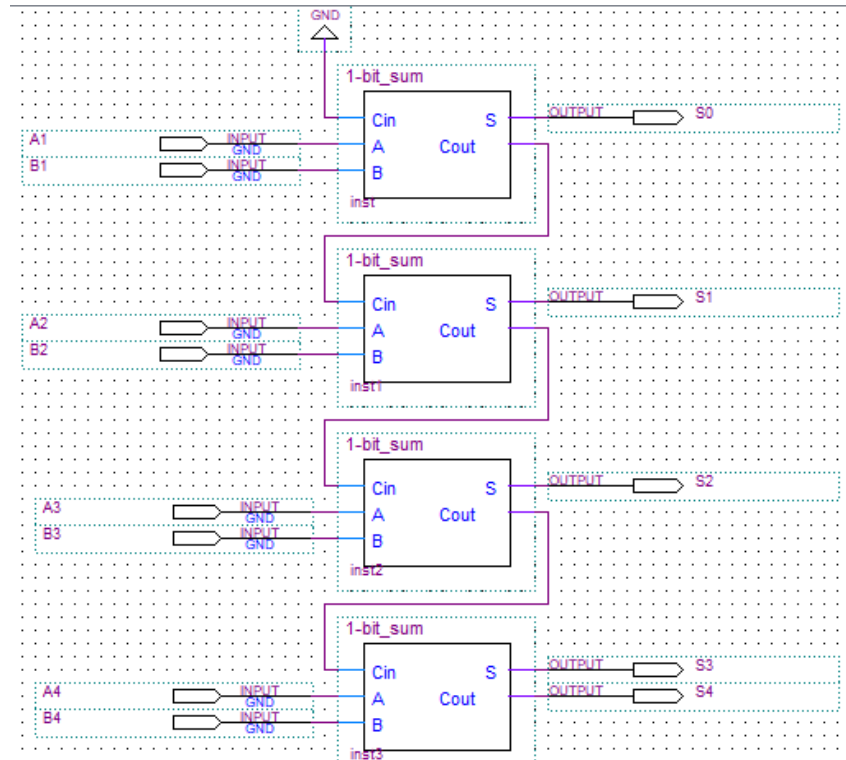


Рисунок 5.28 – Чотирибітний суматор

Зберігаємо як Symbol File.

Перетворюємо від'ємник у додатковий код:

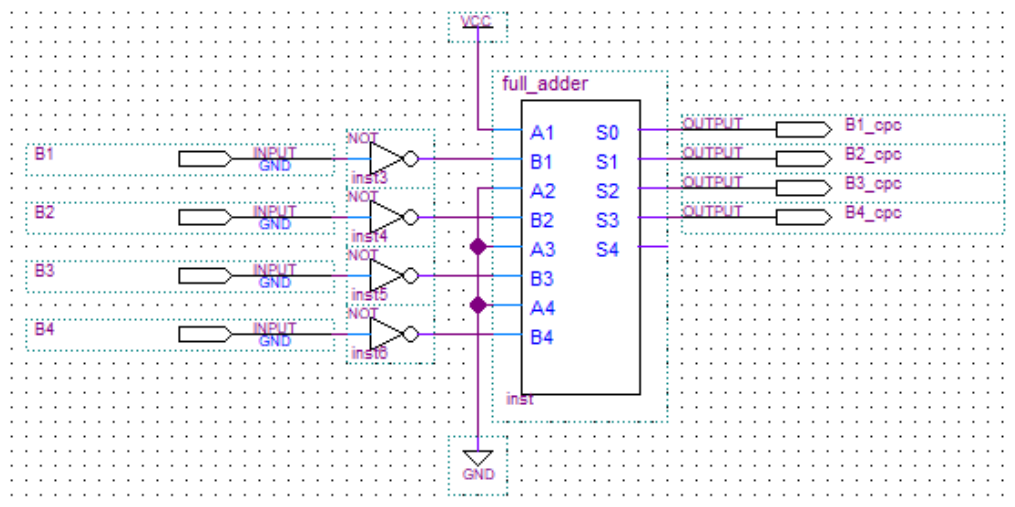


Рисунок 5.29 – Від'ємник перетворений у додатковий код

Створюємо схему, яка виконує задане завдання:

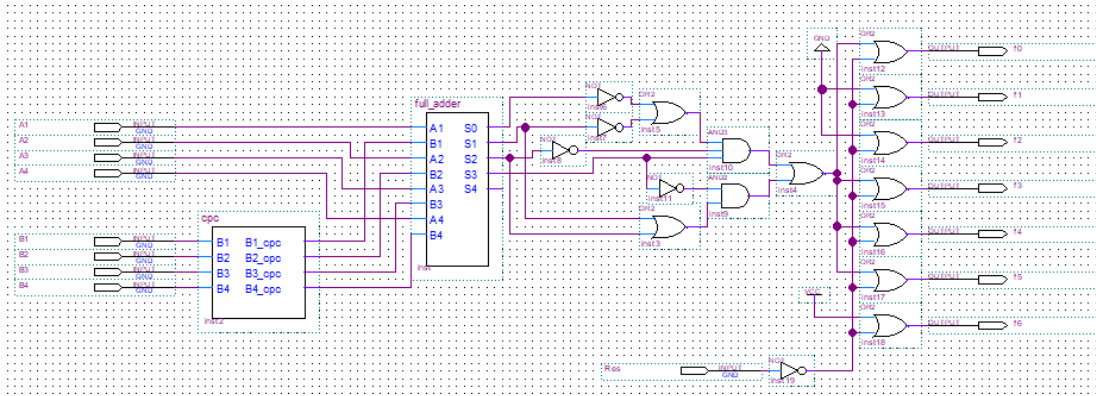


Рисунок 5.30 – Кінцева схема

Після цього компілюємо і, якщо немає помилок, робимо розпіновку та записуємо програму в ПЛІС.

Приклад 2. Створити програму, яка додає два 3-бітних числа і видає 1 на семисегментний індикатор, якщо сума більша або рівна 10. Реалізувати на мультиплексорі-селекторі:

Складемо таблицю істинності (S3-S0 – розряди суми):

Таблиця 5.8 – Таблиця істинності для розробляемого пристрою

S3	S2	S1	S0	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Складемо карту Карно за таблицею істинності:

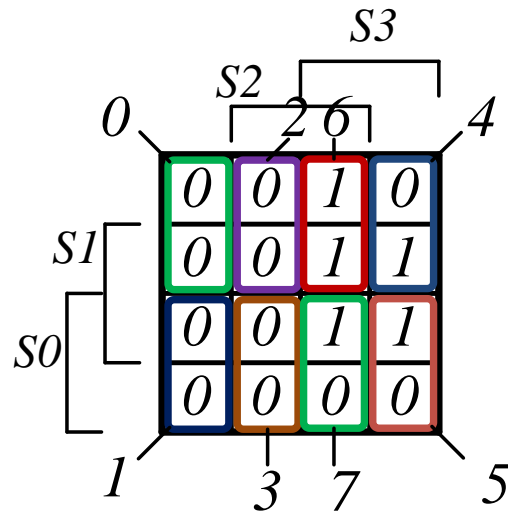


Рисунок 5.31 – Карта Карно

Складемо матрицю селектування:

$$[Z] = \begin{bmatrix} S3 \\ S2 \\ S0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix};$$

Обираємо контури на карті Карно, номер контуру відповідає коду з матриці селектування. Складаємо вектор вхідних сигналів розглядаючи кожен з контурів як карту Карно від змінної S1 і записуючи у вектор рівняння вихідного сигналу:

$$[A] = [0 \ 0 \ 0 \ 0 \ S1 \ S1 \ 1 \ S1];$$

Тоді програма матиме вигляд:

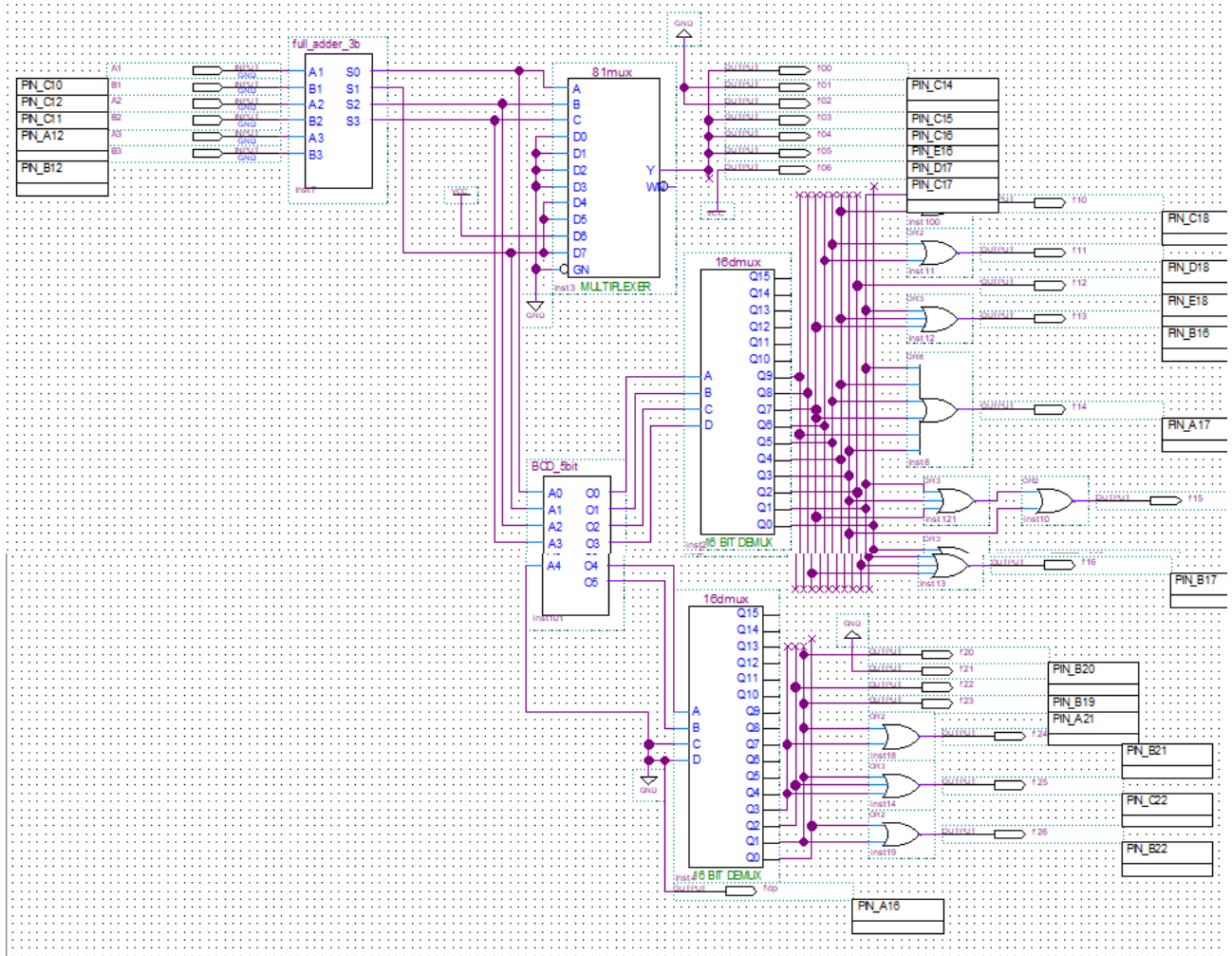


Рисунок 5.32 – Кінцева програма

Блок 81mux – це мультиплексор на 3 селекторні лінії, де А – молодший розряд, а С – старший, GN – інверсний дозвіл роботи, D0-D7 – інформаційні входи.

Блок 16dmux – демультимплексор, перетворює 4-бітний двійковий код у десятковий.

Блок BCD_5bit – це блок, що перетворює двійковий код у двійково-десятковий. Виходи O0-O3 – одиниці, O4-O5 – десятки (п'ятибітне двійкове число не може бути більшим за 37 тому у десятків всього два виходи).

Складемо графи переходів за умовами завдання:

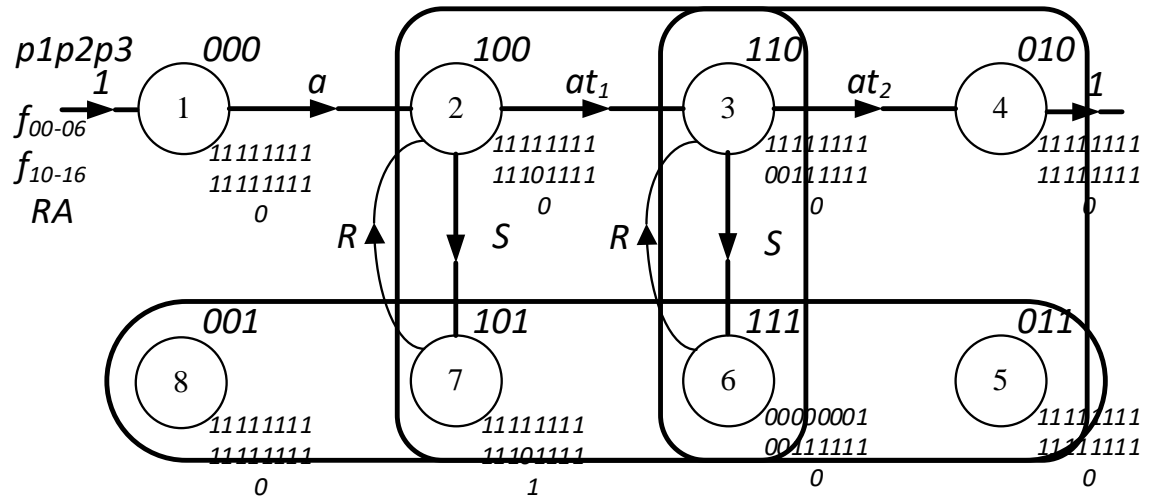


Рисунок 5.34 – Граф переходу для вибору режимів

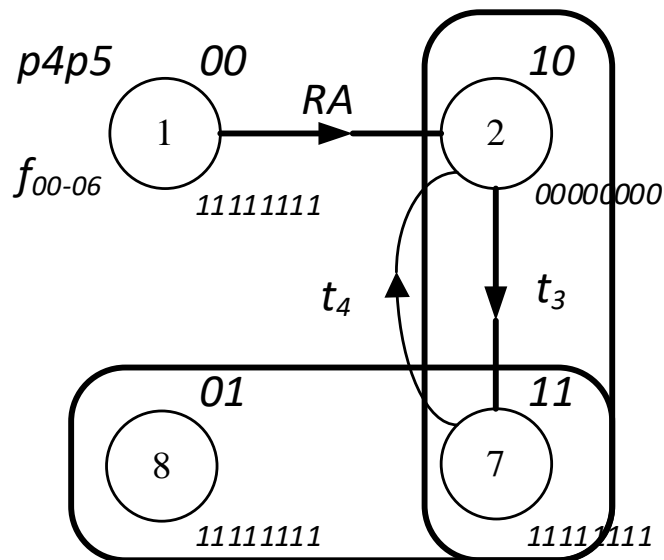


Рисунок 5.35 – Граф переходу для мигання цифрою 8

Запишемо рівняння за графами переходів:

$$\begin{aligned}
 S_{p_1} &= \overline{ap_2p_3}; & S_{p_2} &= t_1ap_1\overline{p_3}; & S_{p_3} &= Sp_1; & S_{p_4} &= RA\overline{p_5}; & S_{p_5} &= t_3p_4; \\
 R_{p_1} &= t_2ap_2\overline{p_3} + c; & R_{p_2} &= p_1\overline{p_3} + c; & R_{p_3} &= Rp_1 + c; & R_{p_4} &= R + c; & R_{p_5} &= t_4p_4 + R + c; \\
 T_1 &= \overline{p_1p_2p_3}; & T_2 &= \overline{p_1p_2p_3}; \\
 T_3 &= \overline{p_4p_5}; & T_4 &= \overline{p_4p_5}; \\
 RA &= \overline{p_1p_2p_3};
 \end{aligned}$$

$$\begin{aligned}
 f_{10} = f_{11} &= \overline{p_1} + p_2; \\
 f_{12} &= \overline{p_1}; & f'_{00-05} &= \overline{p_1} + \overline{p_2} + \overline{p_3}; & f''_{00-06} &= \overline{p_4} + p_5; & f &= f' \cdot f''; \\
 f_{13} &= \overline{p_1} + \overline{p_2}; & f'_{06} &= 1; \\
 f_{14-16} &= \overline{p_1};
 \end{aligned}$$

В рівняннях для скидання тригерів додавалися сигнали s і R для спрощення графопереходів. На функціональність це не впливає.

Запишемо програму:

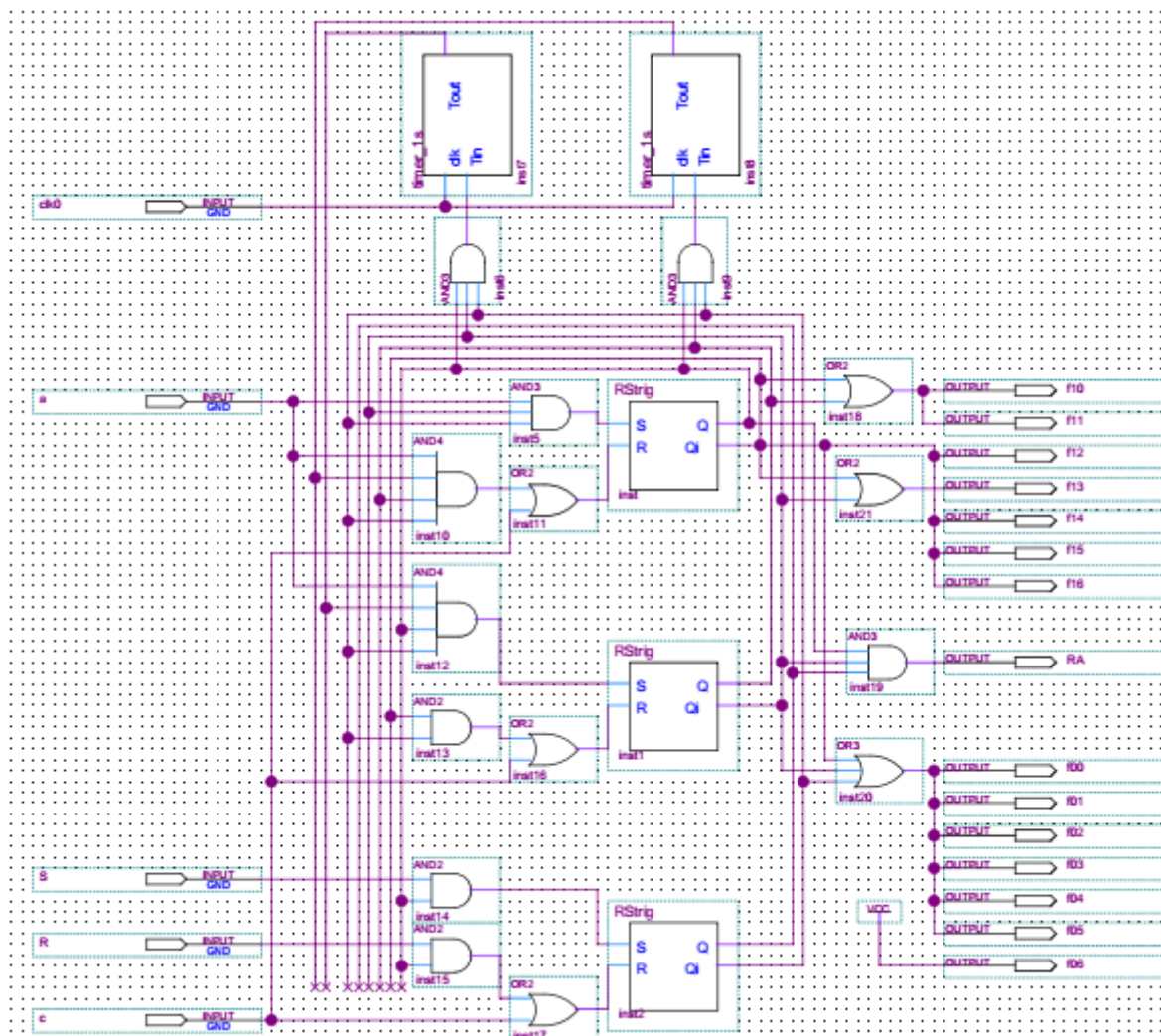


Рисунок 5.36 – Схема складена за першим графом переходу

Зберігаємо дану схему як блок:

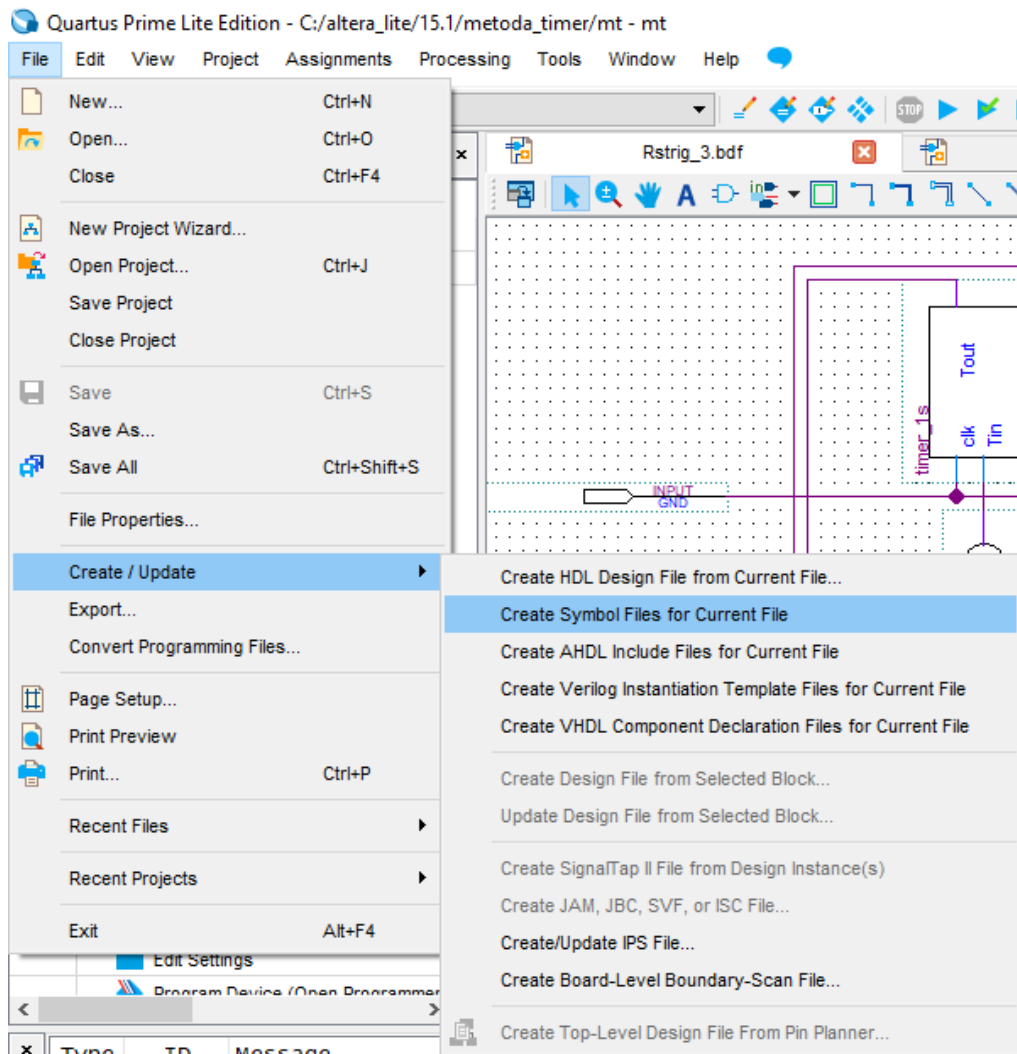


Рисунок 5.37 – Збереження блоку (1)

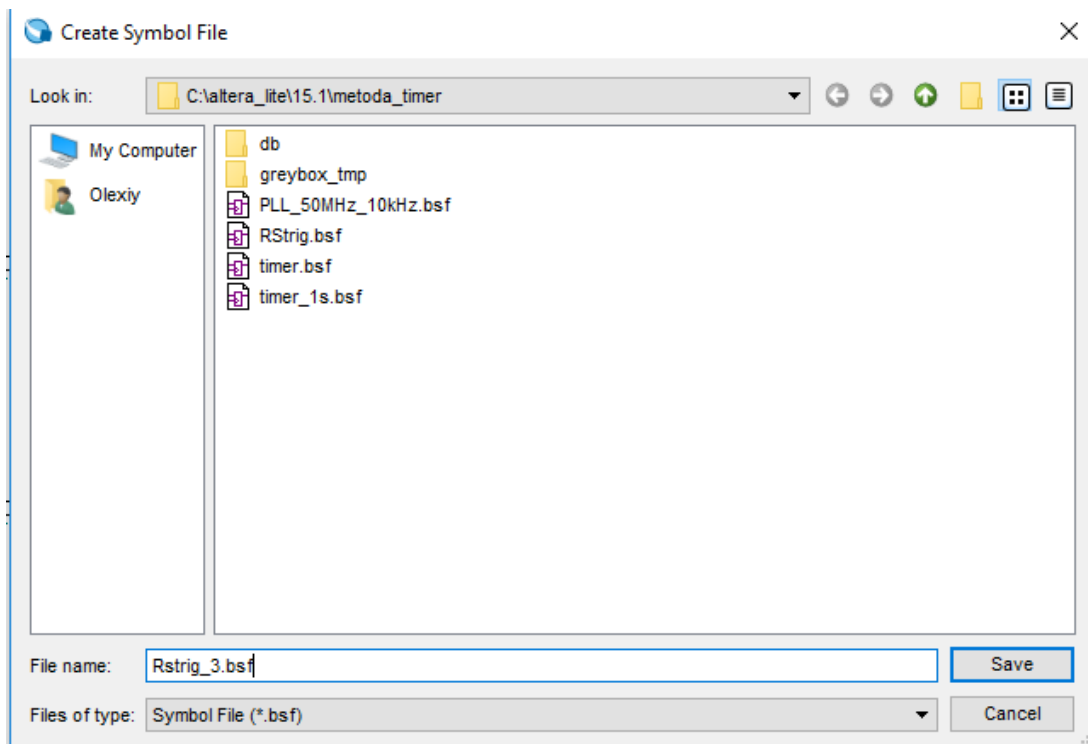


Рисунок 5.38 – Збереження блоку (2)

Аналогічні дії виконаємо над схемою складеною за графом переходу на 2 тригерах:

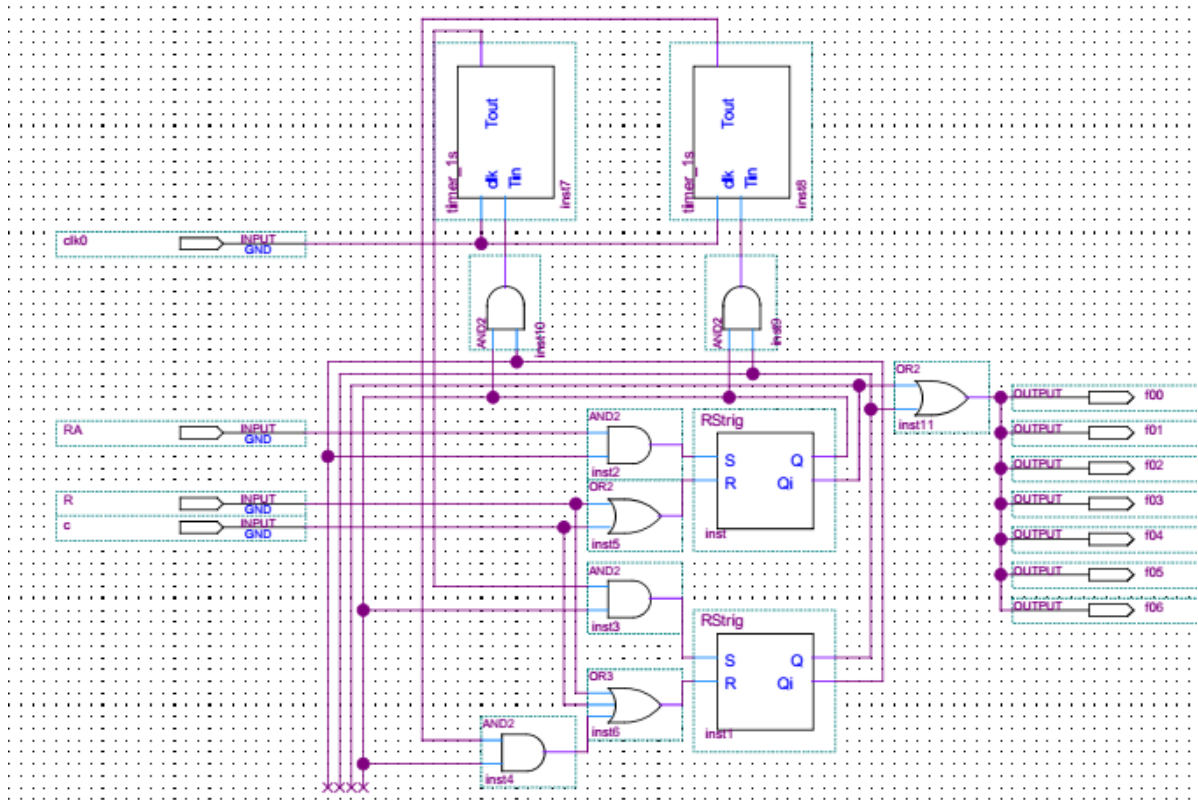


Рисунок 5.39 – Схема складена за другим графом переходу

Загальна схема:

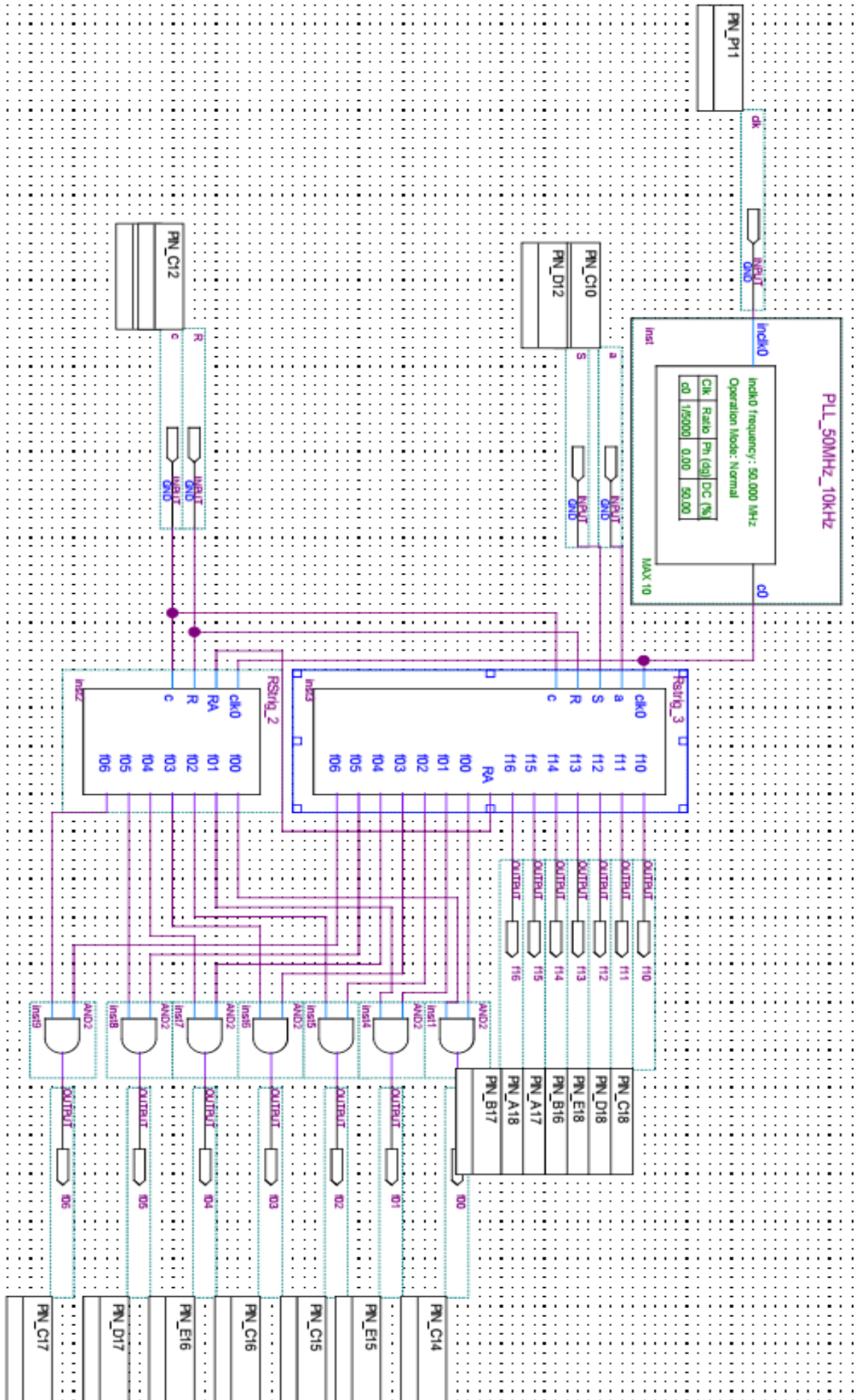


Рисунок 5.40 – Кінцева схема

5.5 Зміст звіту

- 1) Титульний аркуш.
- 2) Мета роботи, програма роботи.
- 3) Умова роботи схем згідно варіанту.
- 4) Логічний синтез схеми керування.
- 5) Програма на мові FBD.
- 6) Таблиці для вхідних/вихідних сигналів.
- 7) Висновок про відповідність роботи заданим умовам.

5.6. Контрольні питання

1. Які зовнішні пристрої підключені до ПЛІС?
2. Поясніть особливості роботи зі світлодіодами та семисегментними індикаторами.
3. Поясніть яким чином на мові FBD програмуються затримки часу.
4. Поясніть призначення входів та виходів однобітного суматора
5. Як у двійковій системі числення знаходиться сума та різниця двох чисел?
6. Що таке двійково-десятковий код та чим він відрізняється від двійкового?

Лабораторна робота №6

СИНТЕЗ ТА ДОСЛІДЖЕННЯ СХЕМ НА ТРИГЕРАХ НА ОСНОВІ ПЛАТИ DE10-LITE

Тривалість лабораторної роботи – 4 год.

Тривалість домашньої роботи – 4 год.

Мета роботи – поглибити вміння вирішувати практичні задачі синтезу логічних виразів для систем автоматизації методом RS-тригерів, навчитися розробити на основі цих виразів програми на мові FBD в програмі Quartus Prime та перевірити роботу їх працездатність на платі розробника DE10-Lite.

6.1 Основні теоретичні відомості

Тригер – це логічна схема зі зворотними зв'язками, яка має релейну характеристику керування і може перебувати в одному з двох усталених станів, забезпечуваних цими зв'язками. Тригер характеризується властивістю зберігати інформацію, тому, використовуючи вихідні сигнали тригерів як проміжні змінні, можна будувати багатотактні схеми.

За способом запису інформації тригери поділяють на несинхронізовані (асинхронні) та синхронізовані (синхронні). Багатотактні схеми за принципом дії також поділяють на асинхронні та синхронні. Для побудови асинхронних схем застосовують несинхронізовані RS-тригери. Синхронні схеми звичайно будують на синхронізованих JK-тригерах.

Для синтезу багатотактної схеми на тригерах умови її роботи подають у вигляді графа переходів.

Граф переходів – це графічне зображення послідовності роботи багатотактної схеми. Елементами графу є *вершини* і *ребра*. Вершини відповідають станам схеми і позначаються кружками. Ребра – це лінії зі стрілками, що з'єднують вершини і показують напрям переходу з одного стану схеми в інший.

Кількість вершин графу при синтезі асинхронних схем на RS -тригерах визначають з умови $2^n \geq S$, де S – кількість станів схеми; 2^n – кількість вершин графу; n – кількість тригерів. Вершини графу рекомендується розміщувати так, щоб коли $n = 2$, вони створювали конфігурацію 2×2 , якщо $n = 3$, – конфігурацію 4×2 , а якщо $n = 4$, – конфігурацію 4×4 . Вигляд графу переходів для $n = 3$ зображено на рис. 5.1.

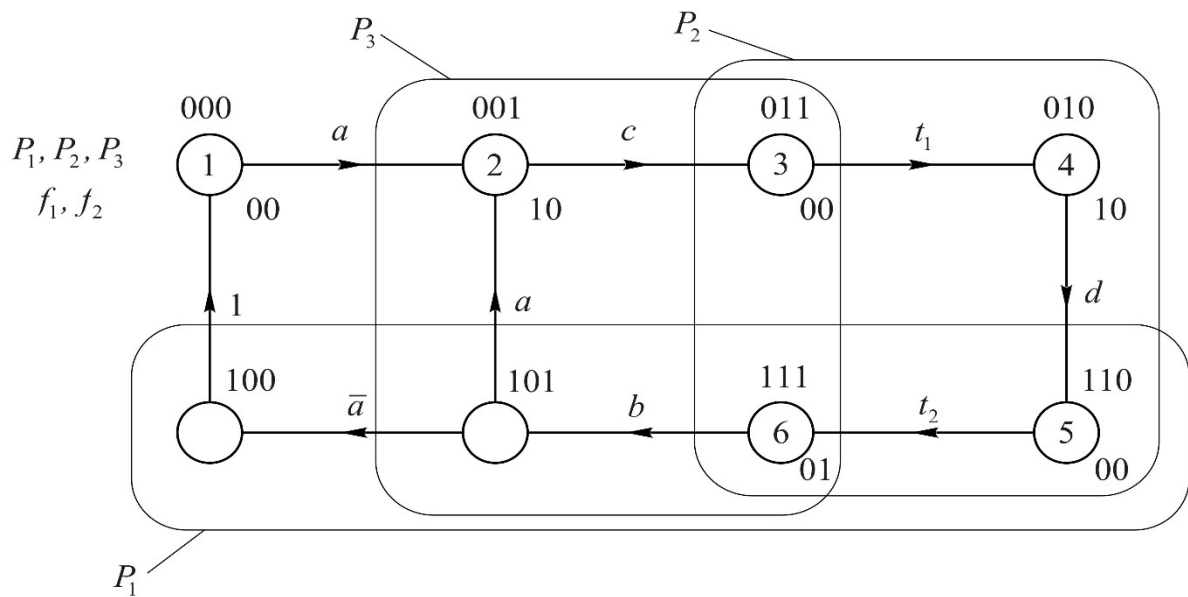


Рисунок 6.1 – Граф переходів для $n = 3$

Вихідні сигнали тригерів виконують роль проміжних змінних, тому тригери та їх вихідні сигнали позначають буквами P_1, P_2, P_3 . Ці позначення записують зверху ліворуч від графу.

Кожна вершина графу кодується набором значень вихідних сигналів тригерів. Коди обирають так, щоб для сусідніх вершин вони розрізнялися значенням тільки однієї змінної. У вихідному стані схеми (стан очікування) зазвичай припускають, що всі проміжні змінні дорівнюють нулеві. Значення проміжних змінних для кожної вершини записують над кружками у тій черговості, у якій записано позначення тригерів.

Всередині вершин записують номери станів схеми. Стани, між якими має відбуватися перехід згідно з умовами роботи схеми, розміщуються в сусідніх вершинах, якщо це можливо. Під вершинами з номерами станів записують значення вихідних змінних, що відповідають цим станам. Позначення вихідних змінних f_1, f_2, \dots, f_n записують ліворуч від графу під позначеннями тригерів.

Вершини, між якими повинні відбуватися переходи, з'єднують ребрами зі стрілками. Над стрілками або праворуч від них, якщо ребра спрямовано вертикально, записують позначення вхідних сигналів, що спричиняють ці переходи.

У схемах на асинхронних *RS*-тригерах переходи можна робити тільки між сусідніми вершинами. Якщо ця умова не виконується, то необхідно передбачити природні переходи (за рахунок подавання вхідного сигналу 1) через проміжні нестійкі стани.

6.2 Програма роботи

1. Виконати логічний синтез за умовами роботи схеми для заданого варіанту (див. табл. 6.1).
2. Скласти схему в Quartus Prime 15.1 Lite Edition згідно отриманих логічних виразів.
3. Записати програму на плату DE10-LITE.
4. Перевірити відповідність роботи заданим умовам.

Таблиця 6.1 – Варіанти завдань

Бригада	Номер завдання, яке визначає умову роботи
1	1; 7
2	2; 8
3	3; 9
4	4; 10

Умови роботи схем

1. Виконуючи пуск тумблером SW1 має виконуватися подальший алгоритм засвіченням світлодіодних індикаторів, з реалізацією інтервалу затримки часу за допомогою таймерів, за принципом: HEX00 - HEX11 - HEX21 - HEX21 - HEX34 - HEX45 - HEX56. При натисненні кнопки «Стоп» схема вимикається з будь-якого стану.

2. Виконуючи пуск тумблером SW1 має виконуватися подальший алгоритм засвіченням світлодіодних індикаторів, з реалізацією інтервалу затримки часу за допомогою таймерів, за принципом: HEX00 - HEX00, HEX03 -

HEX00, HEX03, HEX01 - HEX00, HEX03, HEX01, HEX15 - HEX00, HEX03, HEX01, HEX15, HEX10 - HEX00, HEX03, HEX01, HEX15, HEX10, HEX13 - HEX00, HEX03, HEX01, HEX15, HEX10, HEX13, HEX26. При натисненні кнопки «Стоп» схема вимикається з будь-якого стану.

3. При натисненні кнопки «Пуск» засвічується цифра 1 (на першому 7-сегментному індикаторі) ,а через затримку часу світиться цифра 4 (на другому 7-сегментному індикаторі, без затухання попередньої цифри).Далі при натисненні тумблера засвічується цифра 8 (на третьому 7-сегментному індикаторі), через затримку часу засвічується цифра 8(на четвертому 7-сегментному індикаторі) та 9 (на п'ятому 7-сегментному індикаторі), через затримку часу все вимикається. При натисненні кнопки «Стоп» схема вимикається з будь-якого стану.

4. При натисненні кнопки «Пуск» засвічується цифра 8 (на першому 7-сегментному індикаторі) та 9 (на другому 7-сегментному індикаторі, без затухання попередньої цифри), а через затримку часу засвічується 8 та 9 діод. Далі при натисненні тумблера засвічується цифра 3 (на третьому 7-сегментному індикаторі), через затримку часу засвічується буква А (на четвертому 7-сегментному індикаторі) та 7 (на п'ятому 7-сегментному індикаторі), через затримку часу засвічується 3 та 7 діод, далі схема вимикається. При натисненні кнопки «Стоп» схема вимикається з будь-якого стану.

5. При натисканні на «Пуск» мають загорітися діоди кратні 2 та, одночасно з ними на 7-сегментному індикаторі цифра 2. При натисканні на кнопку «Стоп» кожен діод має тухнути по черзі з певним інтервалом часу. На 7-сегментному індикаторі відображається кількість діодів, які були вимкнені. При натисненні кнопки «Стоп» схема вимикається з будь-якого стану.

6. При натисканні на «Пуск» мають загорітися непарні діоди та, одночасно з ними на 7-сегментному індикаторі цифра 1. При натисканні на кнопку «Стоп» кожен діод має тухнути по черзі з певним інтервалом часу. На 7-сегментному індикаторі відображається кількість діодів, які світяться. При натисненні кнопки «Стоп» схема вимикається з будь-якого стану.

7. При натисканні кнопки «Пуск» не загоряється жоден діод, при цьому, на 7-сегментному індикаторі світиться цифра 0. Через певну затримку часу t_1 загоряється діод номер 9, а на 7-сегментному індикаторі світиться цифра 9. Через затримку часу t_2 гасне діод номер 9 та число на 7-сегментному індикаторі. Загоряється 8 і 7 діод, а на 7-сегментному індикаторі світиться цифра 8 і 7. Через затримку часу t_3 гасне діоди номер 7,8 та число на 7-сегментному індикаторі. Загоряється 6, 4, 3 діоди, а на 7-сегментному індикаторі світиться цифра 6, 4, 3. Через затримку часу t_4 все вимикається. При натисненні кнопки «Стоп» схема вимикається з будь-якого стану.

8. Схема працює у двох режимах, перемикання між якими здійснюється за допомогою тумблера. У першому режимі схема генерує послідовність парних чисел, починаючи з нуля. Час затримки між числами повинен, починаючи з 1 секунди, збільшуватися на 1 секунду з кожним новим парним числом. При поверненні схеми у вихідний стан затримка часу знову починається з 1 секунди. Запуск схеми робити за допомогою кнопки «Пуск 1», зупинку за допомогою кнопки «Стоп 1». У другому режимі схема генерує одиниці, які світяться впродовж 3 секунд із паузою 4 секунди. Для керування пуском/зупинкою використовувати кнопки «Пуск 2» та «Стоп 2».

9. Схема працює у двох режимах, перемикання між якими здійснюється за допомогою тумблера. У першому режимі схема генерує послідовність непарних чисел, починаючи з одиниці. Час затримки між числами повинен, починаючи з 1 секунди, збільшуватися на 1 секунду з кожним новим непарним числом. При поверненні схеми у вихідний стан затримка часу знову починається з 1 секунди. Запуск схеми робити за допомогою кнопки «Пуск 1», зупинку за допомогою кнопки «Стоп 1». У другому режимі схема генерує послідовність чисел 1, 2, 3, 1, 2, 3, 1... із затримкою часу 5 секунд. Для керування пуском/зупинкою використовувати кнопки «Пуск 2» та «Стоп 2».

10. Схема працює у двох режимах, перемикання між якими здійснюється за допомогою тумблера. У першому режимі схема генерує послідовність чисел 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0.... Час затримки між числами

повинен бути 2 секунди. Запуск схеми робити за допомогою кнопки «Пуск 1», зупинку за допомогою кнопки «Стоп 1». У другому режимі схема генерує послідовність чисел 1, 3, 7, 1, 3, 7... із затримкою часу 5 секунд. Для керування пуском/зупинкою використовувати кнопки «Пуск 2» та «Стоп 2».

11. Схема працює у двох режимах, перемикання між якими здійснюється за допомогою тумблера. У першому режимі схема генерує послідовність чисел 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Після надходження числа 9 схема зупиняє роботу. Для продовження необхідно натиснути кнопку «Продовжити». Час затримки між числами повинен бути 2 секунди. Запуск схеми робити за допомогою кнопки «Пуск 1», зупинку за допомогою кнопки «Стоп 1». У другому режимі схема генерує послідовність чисел 1, 3, 7, 1, 3, 7... із затримкою часу 5 секунд. Для керування пуском/зупинкою використовувати кнопки, що і для першого режиму.

12. Схема працює у двох режимах, перемикання між якими здійснюється за допомогою тумблера. У першому режимі схема генерує послідовність парних чисел в зворотному порядку 8, 6, 4, 2, 0, 8, 6, 2.... Після надходження чисел 6 та 2 схема зупиняється. Для продовження її роботи необхідно натискати кнопку «Продовжити». Час затримки між числами повинен бути 2 секунди. Запуск схеми робити за допомогою кнопки «Пуск 1», зупинку за допомогою кнопки «Стоп 1». У другому режимі схема генерує ту саму послідовність чисел, але без зупинки. Для керування пуском/зупинкою використовувати кнопки, що і для першого режиму.

Увага!! Реалізацію таймерів дивитися в Лабораторній роботі №2.

6.3 Методичні вказівки до виконання лабораторної роботи

Приклад 1. Схема працює у двох режимах, перемикання між якими здійснюється за допомогою тумблера. У першому режимі схема генерує послідовність чисел в зворотному порядку 9,8,6,2,0,9,8,6,2.... Після надходження чисел 8 та 2 схема зупиняється. Для продовження її роботи необхідно натискати кнопку «Продовжити». Реалізувати час затримки між числами. Зупинку за допомогою кнопки «Стоп». У другому режимі схема

генерує ту саму послідовність чисел, але без зупинки. Для керування пуском/зупинкою використовувати кнопки, що і для першого режиму.

Синтез генератора будемо роботи виходячи з того, що виходи схеми підключаються до світлодіодів семисегментного індикатора, як показано на рис. 2. Зауважте, що активним є низький логічний рівень.

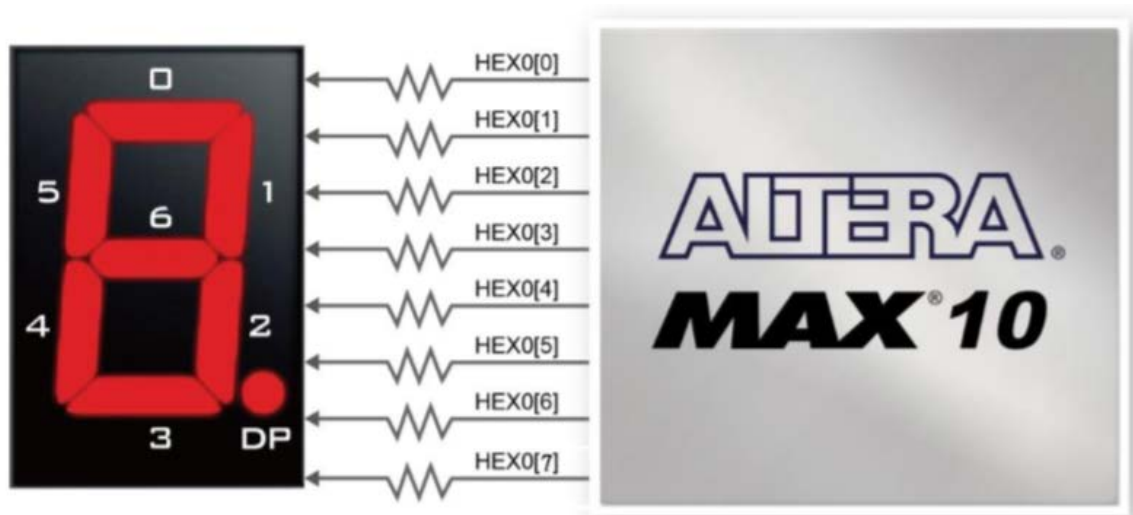


Рисунок 6.2 – Підключення виходів схеми до світлодіодних індикаторів

Таблиця 6.2 – Розпіновка LED

Назва сигналу	Номер PIN-коду FPGA	Опис	Стандарт I / O
HEX00	PIN_C14	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX01	PIN_E15	Seven Segment Digit 0[1]	3.3-V LVTTL
HEX02	PIN_C15	Seven Segment Digit 0[2]	3.3-V LVTTL
HEX03	PIN_C16	Seven Segment Digit 0[3]	3.3-V LVTTL
HEX04	PIN_E16	Seven Segment Digit 0[4]	3.3-V LVTTL
HEX05	PIN_D17	Seven Segment Digit 0[5]	3.3-V LVTTL
HEX06	PIN_C17	Seven Segment Digit 0[6]	3.3-V LVTTL
HEX07	PIN_D15	Seven Segment Digit 0[7], DP	3.3-V LVTTL
HEX10	PIN_C18	Seven Segment Digit 1[0]	3.3-V LVTTL
HEX11	PIN_D18	Seven Segment	3.3-V LVTTL

		Digit 1[1]	
HEX12	PIN_E18	Seven Segment Digit 1[2]	3.3-V LVTTL
HEX13	PIN_B16	Seven Segment Digit 1[3]	3.3-V LVTTL
HEX14	PIN_A17	Seven Segment Digit 1[4]	3.3-V LVTTL
HEX15	PIN_A18	Seven Segment Digit 1[5]	3.3-V LVTTL
HEX16	PIN_B17	Seven Segment Digit 1[6]	3.3-V LVTTL
HEX17	PIN_A16	Seven Segment Digit 1[7] , DP	3.3-V LVTTL
HEX20	PIN_B20	Seven Segment Digit 2[0]	3.3-V LVTTL
HEX21	PIN_A20	Seven Segment Digit 2[1]	3.3-V LVTTL
HEX22	PIN_B19	Seven Segment Digit 2[2]	3.3-V LVTTL
HEX23	PIN_A21	Seven Segment Digit 2[3]	3.3-V LVTTL
HEX24	PIN_B21	Seven Segment Digit 2[4]	3.3-V LVTTL
HEX25	PIN_C22	Seven Segment Digit 2[5]	3.3-V LVTTL
HEX26	PIN_B22	Seven Segment Digit 2[6]	3.3-V LVTTL
HEX27	PIN_A19	Seven Segment Digit 2[7] , DP	3.3-V LVTTL
HEX30	PIN_F21	Seven Segment Digit 3[0]	3.3-V LVTTL
HEX31	PIN_E22	Seven Segment Digit 3[1]	3.3-V LVTTL
HEX32	PIN_E21	Seven Segment Digit 3[2]	3.3-V LVTTL
HEX33	PIN_C19	Seven Segment Digit 3[3]	3.3-V LVTTL
HEX34	PIN_C20	Seven Segment Digit 3[4]	3.3-V LVTTL
HEX35	PIN_D19	Seven Segment Digit 3[5]	3.3-V LVTTL
HEX36	PIN_E17	Seven Segment Digit 3[6]	3.3-V LVTTL

HEX37	PIN_D22	Seven Segment Digit 3[7] , DP	3.3-V LVTTL
HEX40	PIN_F18	Seven Segment Digit 4[0]	3.3-V LVTTL
HEX41	PIN_E20	Seven Segment Digit 4[1]	3.3-V LVTTL
HEX42	PIN_E19	Seven Segment Digit 4[2]	3.3-V LVTTL
HEX43	PIN_J18	Seven Segment Digit 4[3]	3.3-V LVTTL
HEX44	PIN_H19	Seven Segment Digit 4[4]	3.3-V LVTTL
HEX45	PIN_F19	Seven Segment Digit 4[5]	3.3-V LVTTL
HEX46	PIN_F20	Seven Segment Digit 4[6]	3.3-V LVTTL
HEX47	PIN_F17	Seven Segment Digit 4[7] , DP	3.3-V LVTTL
HEX50	PIN_J20	Seven Segment Digit 5[0]	3.3-V LVTTL
HEX51	PIN_K20	Seven Segment Digit 5[1]	3.3-V LVTTL
HEX52	PIN_L18	Seven Segment Digit 5[2]	3.3-V LVTTL
HEX53	PIN_N18	Seven Segment Digit 5[3]	3.3-V LVTTL

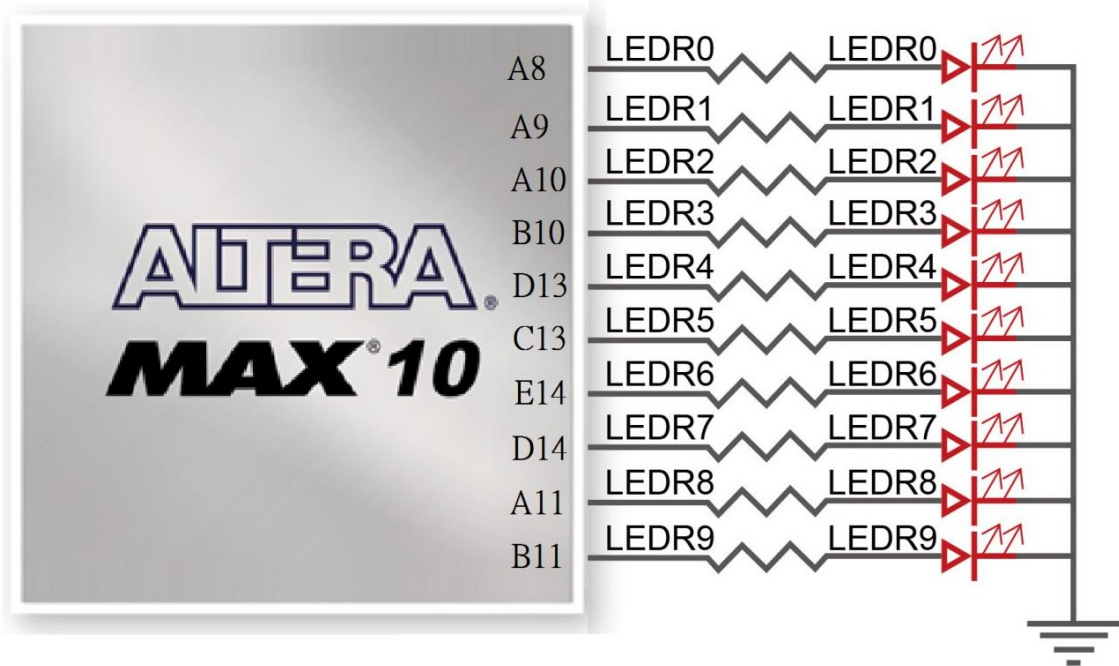


Рисунок 6.3 – Підключення між світлодіодами та MAX 10 FPGA

Таблиця 6.3 – Розпіновка сигналів світлодіодів

Назва сигналу	№ піну FPGA	I/O стандарт
LEDR0	PIN_A8	3.3-V LVTTTL
LEDR1	PIN_A9	3.3-V LVTTTL
LEDR2	PIN_A10	3.3-V LVTTTL
LEDR3	PIN_B10	3.3-V LVTTTL
LEDR4	PIN_D13	3.3-V LVTTTL
LEDR5	PIN_C13	3.3-V LVTTTL
LEDR6	PIN_E14	3.3-V LVTTTL
LEDR7	PIN_D14	3.3-V LVTTTL
LEDR8	PIN_A11	3.3-V LVTTTL
LEDR9	PIN_B11	3.3-V LVTTTL



Рисунок 6.4 – Підключення між слайд-перемикачами і MAX 10 FPGA

Таблиця 6.4 – Призначення контактів перемикачів

Назва сигналу	Номер PIN-коду FPGA	Опис	Стандарт I / O
SW0	PIN_C10	Slide Switch[0]	3.3-V LVTTTL
SW1	PIN_C11	Slide Switch[1]	3.3-V LVTTTL
SW2	PIN_D12	Slide Switch[2]	3.3-V LVTTTL
SW3	PIN_C12	Slide Switch[3]	3.3-V LVTTTL
SW4	PIN_A12	Slide Switch[4]	3.3-V LVTTTL
SW5	PIN_B12	Slide Switch[5]	3.3-V LVTTTL
SW6	PIN_A13	Slide Switch[6]	3.3-V LVTTTL
SW7	PIN_A14	Slide Switch[7]	3.3-V LVTTTL
SW8	PIN_B14	Slide Switch[8]	3.3-V LVTTTL

SW9	PIN_F15	Slide Switch[9]	3.3-V LVTTL
-----	---------	-----------------	-------------

Зробимо такі позначення вхідних і вихідних сигналів, а також сигналів таймерів, які потрібно розглядати як вхідні сигнали для графу переходів. Вхідні сигнали: a – команда «Пуск»; b – команда «Стоп»; c – команда «Пауза»; t_1, t_2, t_3 – сигнали таймерів, кожний з яких забезпечує затримки часу. Вихідні сигнали: $f_1, f_2, f_3, f_4, f_5, f_6, f_7$ – сигнали, що забезпечують вмикання окремих світлодіодів.

Побудову графу переходів починаємо з визначення кількості станів, у яких може перебувати схема автоматичного керування. Таких станів сім: 1 – вихідний стан; 2 – світиться число «9» протягом певного часу; 3 – світиться число «8» протягом певного часу; 4 – світиться число «6» протягом певного часу; 5 – світиться число «2» протягом певного часу; 6 – світиться число «2» протягом певного часу; 7 – світиться число «0» протягом певного часу

Виходячи з кількості станів схеми $7 < 2^3$, визначаємо кількість тригерів $n = 3$ і кількість вершин графу переходів $2^3 = 8$. Позначаємо проміжні змінні P_1, P_2, P_3 , будуємо вісім вершин графу і кодуємо їх комбінаціями значень проміжних змінних (рис. 6.5).

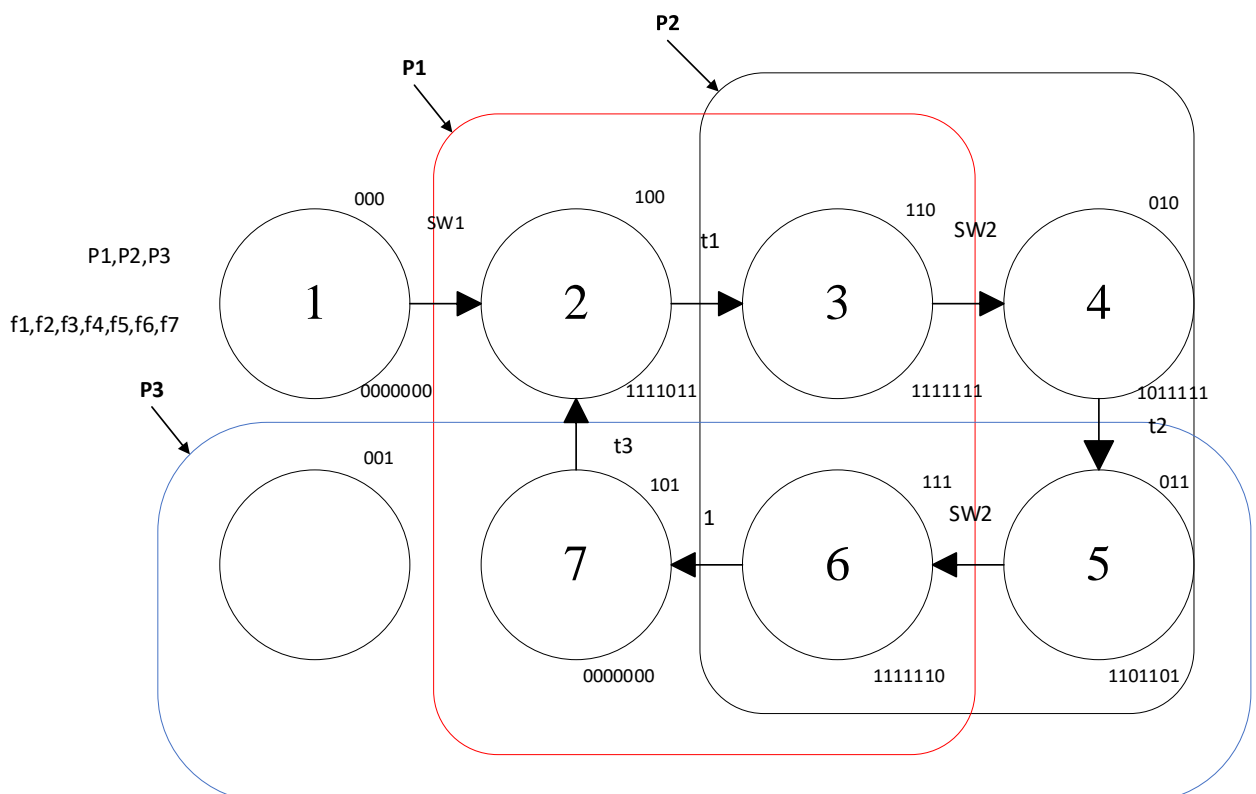


Рисунок 6.5 – Граф переходів до прикладу(для першого режиму)

У сусідніх вершинах розміщуємо стани, між якими потрібен перехід згідно з умовами роботи схеми. Номером 1 позначимо стан очікування. Із надходженням сигналу $SW1$ відбувається перехід у стан 2 ($f_1 = 1, f_2 = 1, f_3 = 1, f_4 = 0, f_5 = 1, f_6 = 1, f_7 = 1$) і світиться цифра «9». Зі стану 2 при надходженні сигналу таймера t_1 схема переходить у стан 3 ($f_1 = 1, f_2 = 1, f_3 = 1, f_4 = 1, f_5 = 1, f_6 = 1, f_7 = 1$) і світиться цифра «8». Зі стану 3 з надходженням сигналу $SW2$ схема переходить у стан 4 ($f_1 = 1, f_2 = 0, f_3 = 1, f_4 = 1, f_5 = 1, f_6 = 1, f_7 = 1$) у цьому стані світиться цифра «6». Зі стану 4 при надходженні сигналу таймера t_1 схема переходить у стан 5 ($f_1 = 1, f_2 = 1, f_3 = 0, f_4 = 1, f_5 = 1, f_6 = 0, f_7 = 1$) і світиться цифра «2». Зі стану 5 при надходженні сигналу $SW2$ схема переходить у стан 6 ($f_1 = 1, f_2 = 1, f_3 = 1, f_4 = 1, f_5 = 1, f_6 = 0, f_7 = 0$) і світиться цифра «0», зі стану 6 через одиничний перехід а далі при надходженні сигналу таймера t_1 схема переходить у стан 2. На цьому побудова графу переходів завершується.

Синтез схем на RS -тригерах виконують у такій послідовності.

1. За заданими умовами роботи схеми будують граф переходів.
2. Записують умови вмикання і скидання кожного тригера. Для цього охоплюють замкнутою лінією всі стани на графі переходів, у яких значення тієї ж самої проміжної змінної (вихідного сигналу тригера) дорівнює одиниці. Вхідні сигнали схеми, позначення яких розміщено на ребрах, що заходять в одержану замкнену область, встановлюють тригер у стан 1, а вхідні сигнали на ребрах, що виходять з цієї області, скидають тригер у стан 0.

Умови вмикання тригера записують у вигляді добутку сигналу на ребрі, що заходить в область, і сигналів решти тригерів, стан яких не змінюється під час переходу, позначеному ребром. Наприклад, якщо сигнал на ребрі, що заходить в область з одиничним значенням вихідного сигналу тригера P_1 , дорівнює a , а тригери P_2 і P_3 не перемикаються, а зберігають стани $P_2 = 1, P_3 = 0$, то умову вмикання тригера P_1 записують у вигляді

$$S_{P_1} = a p_2 \bar{p}_3.$$

Якщо в замкнену область входить кілька ребер, то умову вмикання тригера записують у вигляді суми добутоків відповідних сигналів, складених для кожного ребра.

Умову скидання тригера записують аналогічно для кожного ребра, що виходить з цієї області, і подають у вигляді формули R_p . Уведення сигналів решти тригерів в умови вмикання і скидання цього тригера гарантує здійснення переходів у потрібній послідовності.

Описану процедуру виконують для кожного тригера і визначають для них умови вмикання і скидання.

3. Записують вирази для вихідних сигналів схеми у вигляді комбінаційних функцій вихідних сигналів тригерів.

4. Якщо умовами роботи схеми передбачено технологічні затримки, то складають вирази для вхідних сигналів таймерів як комбінаційні функції вихідних сигналів тригерів.

5. За отриманими логічними формулами складають схему.

Проілюструємо викладену методику синтезу для побудованого графу переходів.

Охоплюємо замкнутою лінією ті стани на графі переходів, для яких $P_1 = 1$. У замкнену область заходить два ребра із сигналами SW1 та SW2. На першому переході $P_2 = 0, P_3 = 0$, а на другому $P_2 = 1, P_3 = 1$. Тому умова вмикання тригера P_1

$$S_{P_1} = SW1\bar{p}_2\bar{p}_3 + SW2p_2p_3.$$

Із замкненої області виходять ребро – із сигналам SW2, для переходу $P_2 = 1, P_3 = 0$. Отже,

$$R_{P_1} = SW2p_2\bar{p}_3.$$

Охоплюємо тепер замкненою лінією стани, для яких $P_2 = 1$. В область заходить ребро із сигналам t_1 . На переході $P_1 = 1$ та $P_3 = 0$, тому

$$S_{P_2} = t_1p_1\bar{p}_3.$$

Умову скидання тригера визначено сигналам 1 при цьому $P_1 = 1, P_3 = 1$ отже,

$$R_{P_2} = 1p_1p_3.$$

Нарешті охоплюємо замкненою лінією стани, для яких $P_3 = 1$. В отриману

область заходить одне ребро із сигналом t_1 і виходять одне ребро із сигналам t_1 .

Отже, з урахуванням станів тригерів P_1 і P_2 , отримаємо

$$S_{P_3} = t_2 \bar{p}_1 p_2;$$

$$R_{P_3} = t_3 \bar{p}_2 p_1$$

Формули для вихідних сигналів $f_1 \dots f_7$ записуємо як комбінаційні функції вихідних сигналів тригерів P_1, P_2, P_3 . Дійсно, функція $f_1 = 1$ у стані 2 тобто в стані, якому відповідають такі комбінації значень вихідних сигналів тригерів: $p_1 = 1, p_2 = 0, p_3 = 0$. Тому

$$f_1 = p_2 + p_1 \bar{p}_2.$$

Аналогічно, для функцій f_2, f_3, f_6 та f_7

$$f_2 = p_1 \bar{p}_3 + p_2 p_3;$$

$$f_3 = p_1 \bar{p}_3 + p_1 p_2 + p_2 \bar{p}_3;$$

$$f_4 = p_2 + p_1 \bar{p}_2;$$

$$f_5 = p_2$$

$$f_6 = p_1 \bar{p}_3 + p_1 p_2 + p_2 \bar{p}_3;$$

$$f_7 = p_1 \bar{p}_3 + p_1 p_2 + p_2 \bar{p}_1;$$

Таймер T_1 вмикається у стані 2 ($p_1 p_2 p_3 = 010$), таймер T_2 – у стані 3 ($p_1 p_2 p_3 = 011$), а таймер T_3 у стані 4 ($p_1 p_2 p_3 = 001$) тому

$$T_1 = p_1 \bar{p}_2 \bar{p}_3;$$

$$T_2 = \bar{p}_1 p_2 \bar{p}_3;$$

$$T_3 = p_1 \bar{p}_2 p_3;$$

Побудова графу переходу для другого режиму:

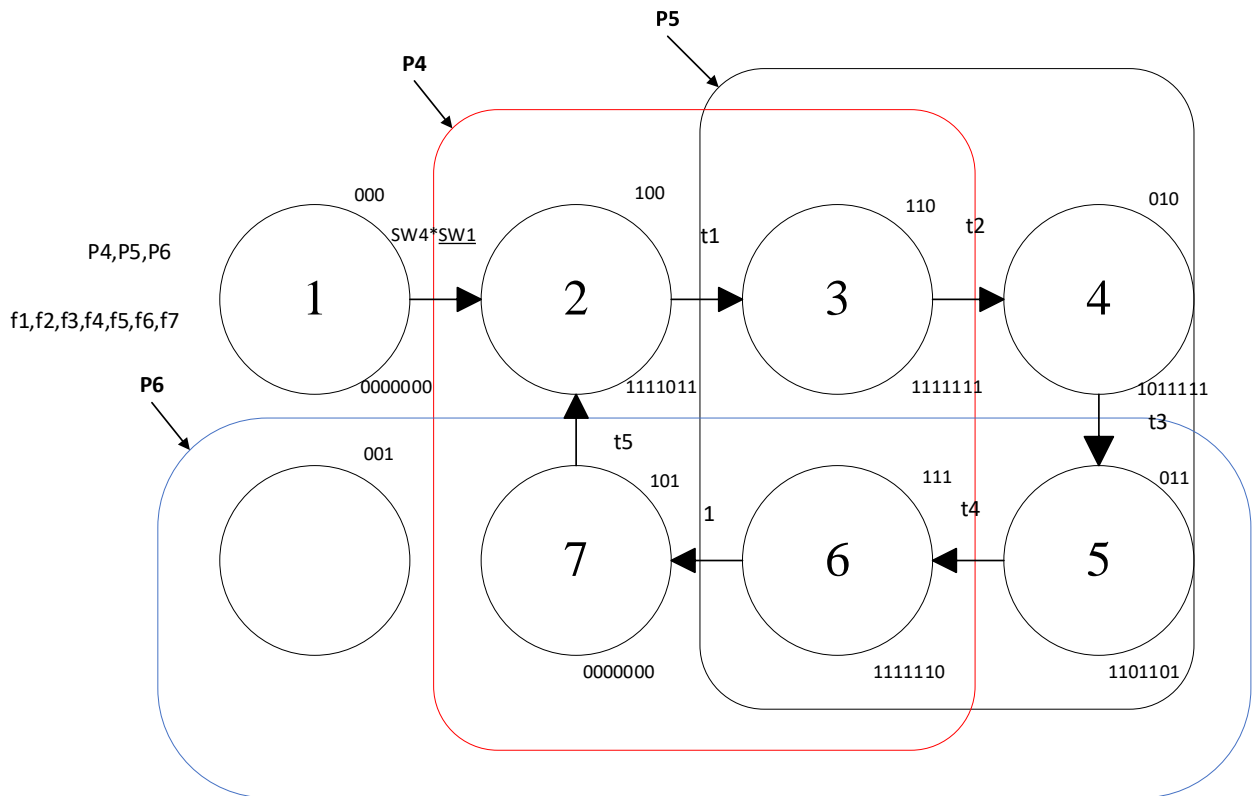


Рисунок 6.6 – Граф переходу для другого режиму

Складемо рівняння встановлення та скидання тригерів:

$$S_{P_4} = SW_4 \bar{S}W_1 \bar{p}_5 \bar{p}_6 + t_4 p_5 p_6.$$

$$R_{P_4} = t_1 p_5 \bar{p}_6.$$

$$S_{P_5} = t_1 p_4 \bar{p}_6.$$

$$R_{P_5} = 1 p_4 p_6.$$

$$S_{P_6} = t_3 \bar{p}_4 p_5;$$

$$R_{P_6} = t_5 \bar{p}_5 p_4$$

Запишемо рівняння для вихідних змінних:

$$f_1 = p_5 + p_4 \bar{p}_5.$$

$$f_2 = p_4 \bar{p}_6 + p_5 p_6;$$

$$f_3 = p_4 \bar{p}_6 + p_4 p_5 + p_5 \bar{p}_6;$$

$$f_4 = p_5 + p_4 \bar{p}_5;$$

$$f_5 = p_5$$

$$f_6 = p_4 \bar{p}_6 + p_4 p_5 + p_5 \bar{p}_6;$$

$$f_7 = p_4 \bar{p}_6 + p_4 p_5 + p_5 \bar{p}_4;$$

Рівняння для таймерів:

$$T_1 = p_4 \bar{p}_5 \bar{p}_6;$$

$$T_2 = p_4 p_5 \bar{p}_6;$$

$$T_3 = \bar{p}_4 p_5 \bar{p}_6;$$

$$T_4 = \bar{p}_4 p_5 p_6;$$

$$T_5 = p_4 \bar{p}_5 p_6;$$

На основі отриманих логічних виразів необхідно скласти схему в програмному середовищі Quartus Prime 15.1 Lite Edition для подальшого програмування плати DE10-Lite.

Створення нового проекту

Відкривши пакет Quartus Prime 15.1 Lite Edition вибираємо New Project Wizard. Потрапляємо у вікно для завдання поточної директорії проекту. Заповнюємо три рядки як показано на рис. 6.7 і натискаємо кнопку Next.

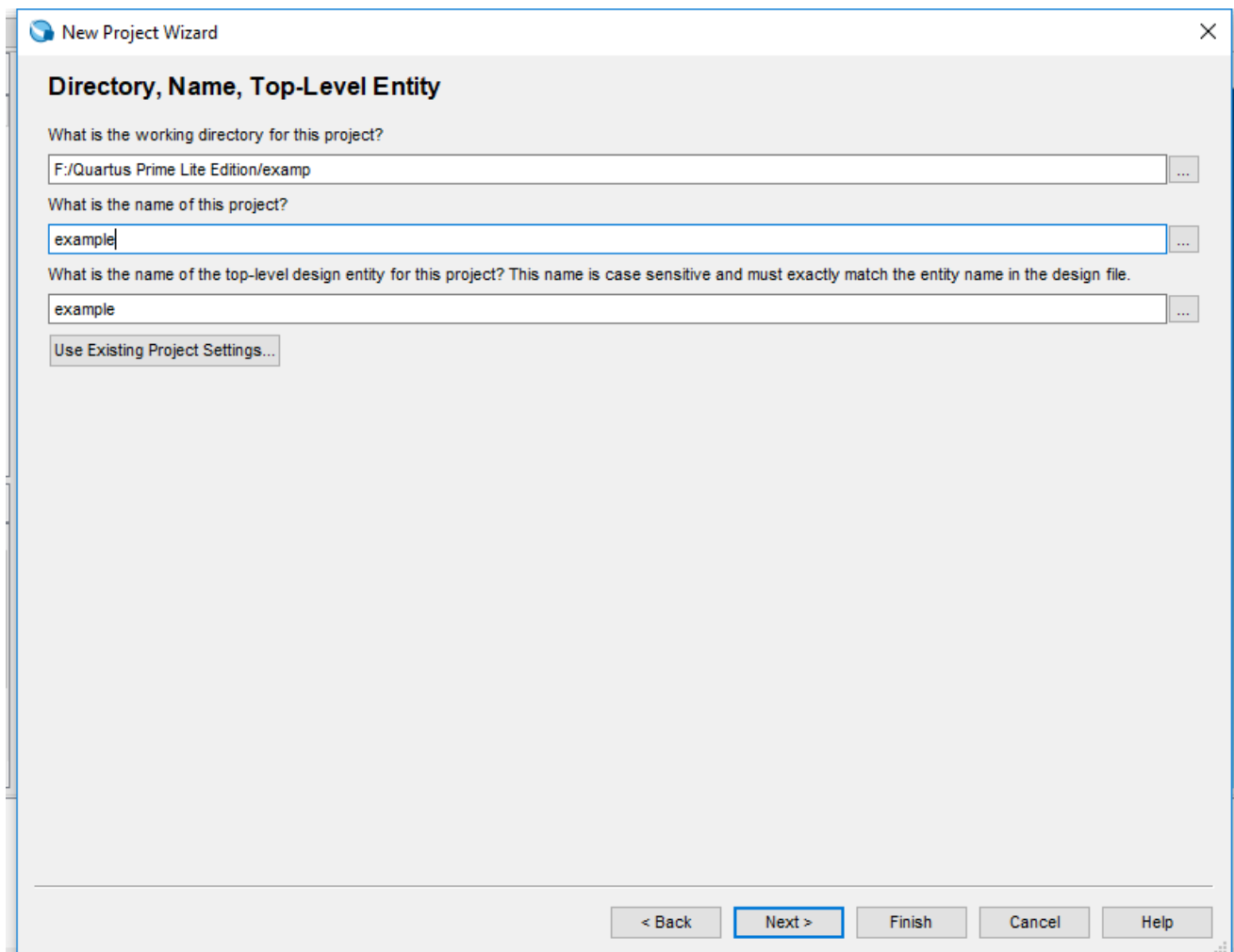


Рисунок 6.7 – Меню завдання поточної директорії проекту

В даному випадку проект буде названо `example` і буде створена директорія за адресою `F:/Quartus Prime Lite Edition/examp`.

Після натиснення кнопки `Next` з'являється вікно де потрібно підтвердити створення нової директорії.

З'являється вікно де можна вибрати вже існуючий файл з програмою. Натискаємо кнопку `Next`, пропускаючи це вікно.

Далі потрапляємо у вікно де потрібно вибрати сімейство та модель мікросхеми. Модель мікросхеми можна подивитись безпосередньо на стенді, вибрати відповідний пункт у вікні, як це і показано на рис. 6.8 та натиснути кнопку `Finish`. Після цього новий проект створено. Для уникнення помилок програмування ПЛІС ретельно перевіряйте правильність вибору моделі мікросхеми.

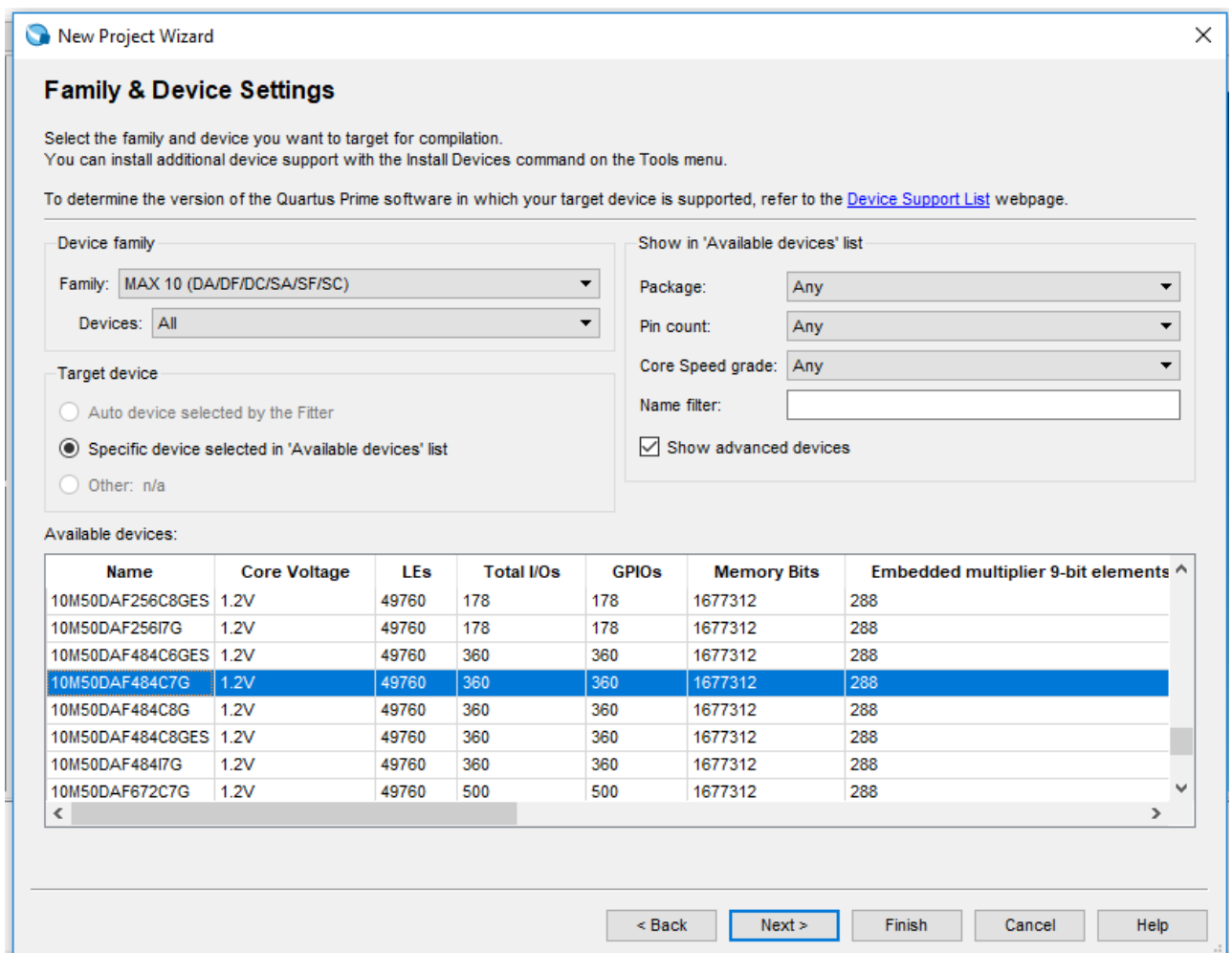


Рисунок 6.8 – Вікно вибору сімейства та моделі мікросхеми

Для того щоб додати новий файл до проекту вибираємо з меню File пункт New. У вікні що з'явилося обираємо мову на якій буде написана програма. Вибираємо BlockDiagram/SchematicFile та натискаємо ОК, рис. 6.9. Новий файл створено.

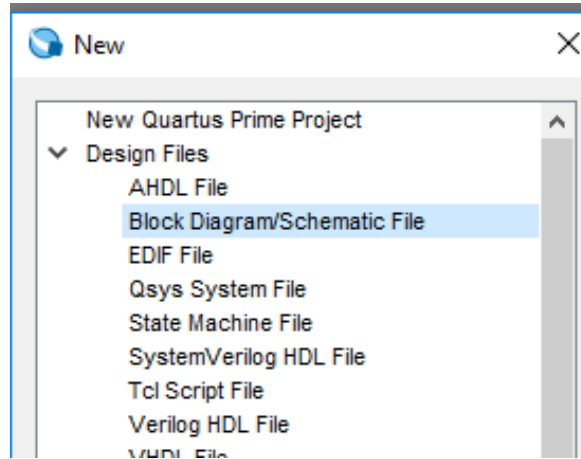


Рисунок 6.9 – Вікно вибору мови програмування та типу файлу

Створення блоків

Після створення нового проекту, відкриваємо новий файл формату BDF (Block Diagram/Schematic File). Будуємо RS-тригер на логічних елементах АБО-НІ, позначаємо відповідними літерами входи та виходи:

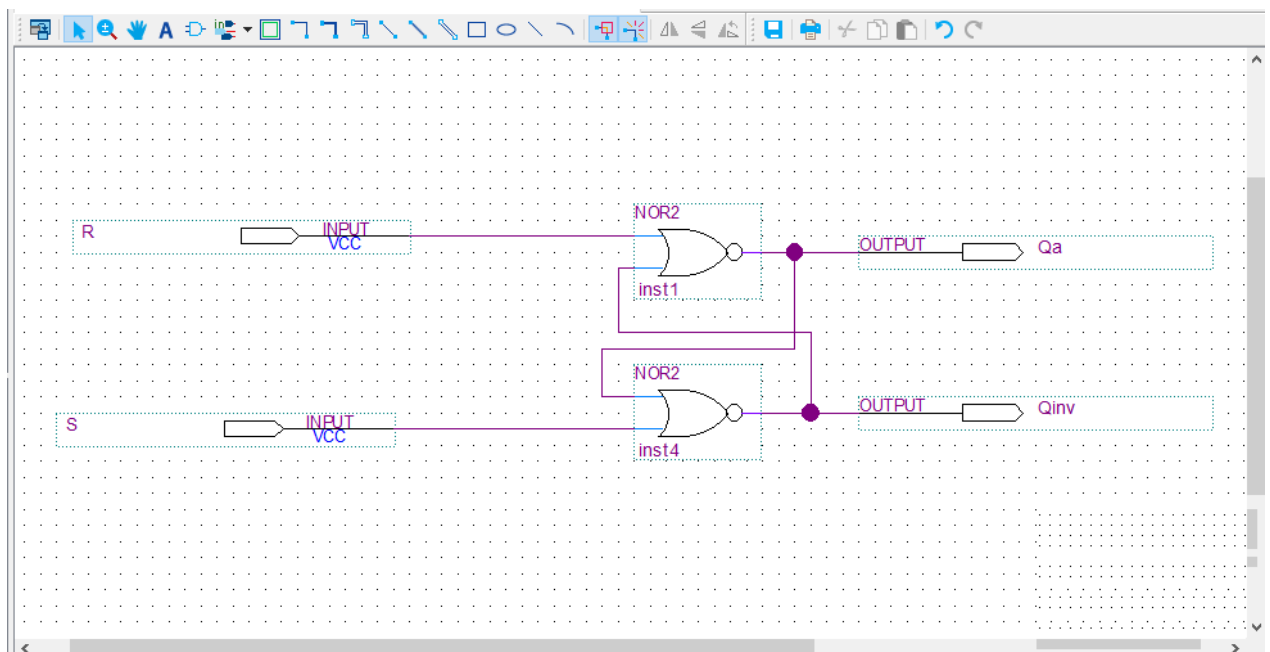


Рисунок 6.10 – Схема RS-тригера

Далі зберігаємо його під назвою, наприклад: RS-triger.

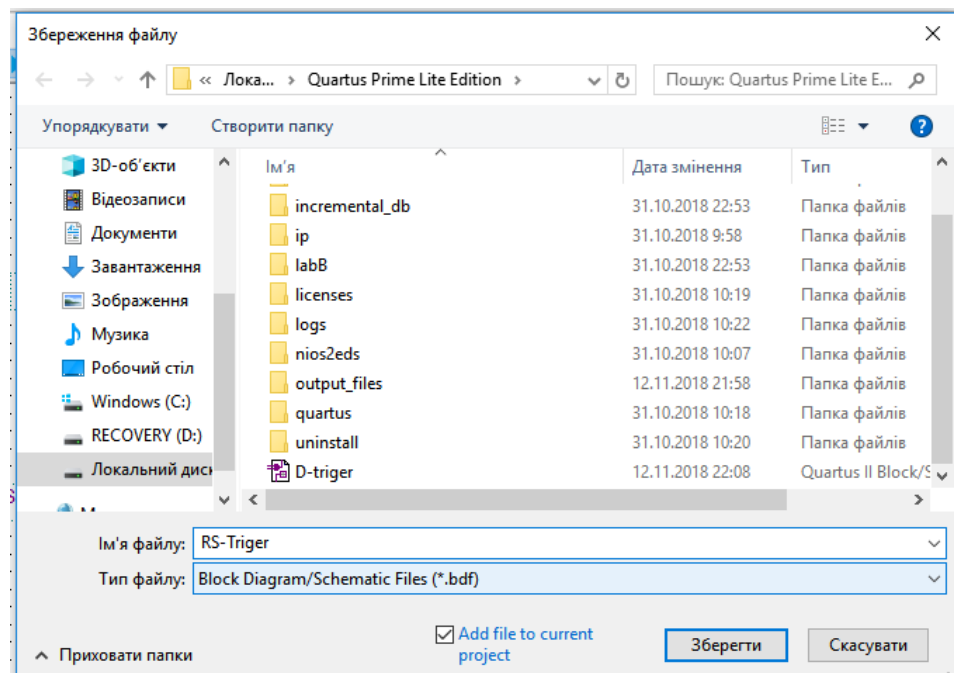


Рисунок 6.11 – Збереження проекту RS-triger

Потім вибираємо з меню: File/Create-Update/Create Symbol File for Current File. Ця дія створить новий компонент, який можна буде використовувати в подальших схемах.

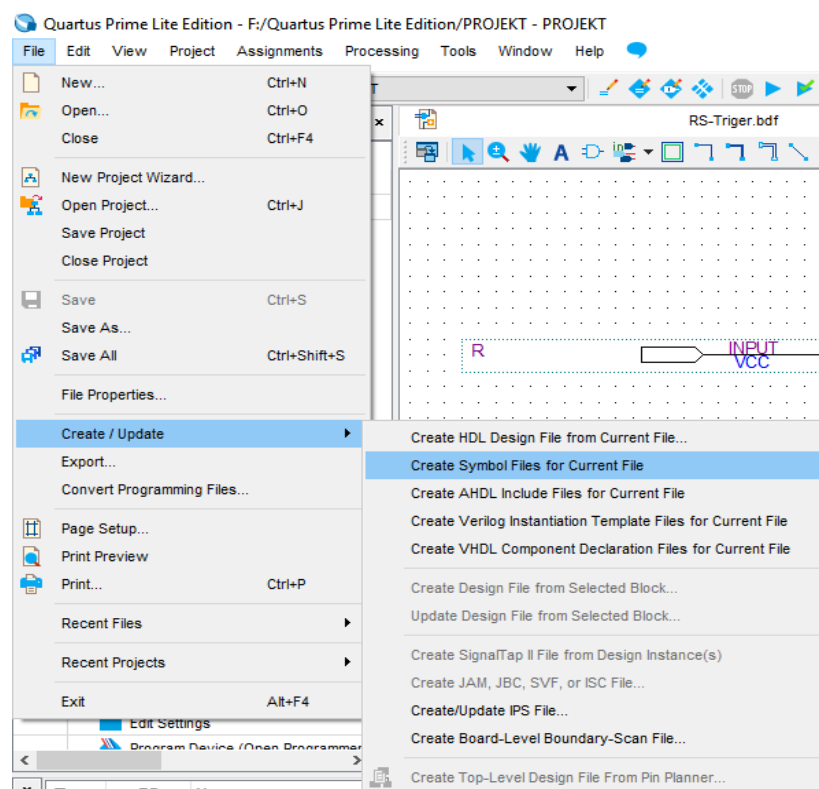


Рисунок 6.12 – Створення блоку RS-triger

По замовчуванню Quartus Prime 15.1 Lite Edition запропонує зберегти блок з такою назвою як наш початковий файл формату *.bdf. Нас це влаштовує, зберігаємо цей файл.

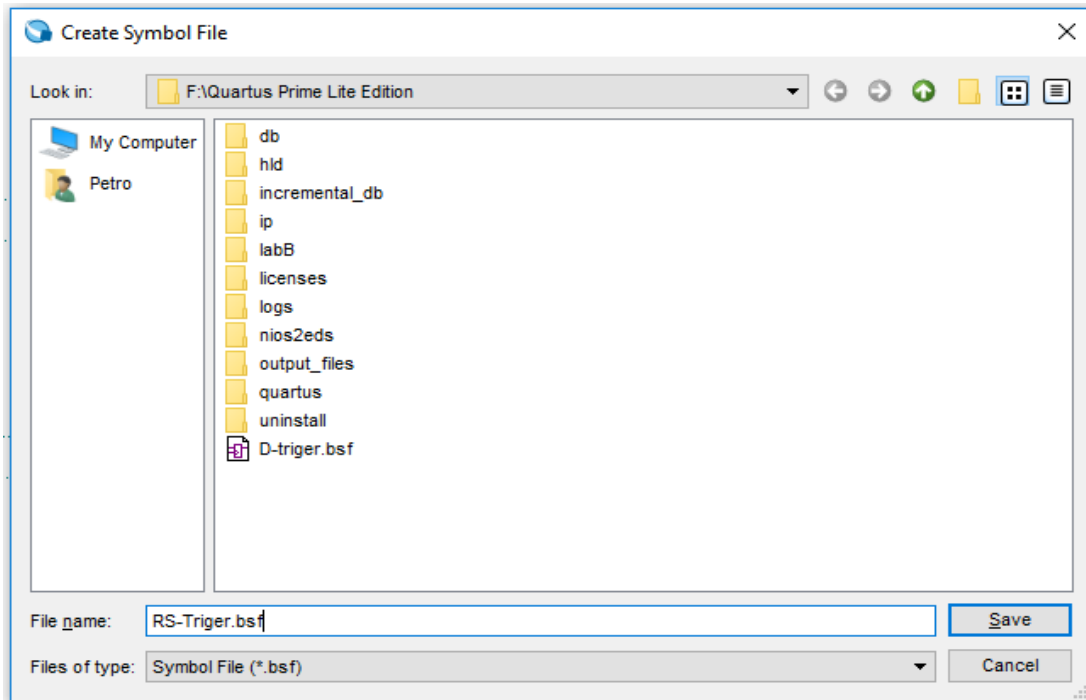


Рисунок 6.13 – Збереження блоку RS-triger

Підтверджуємо збереження блоку:

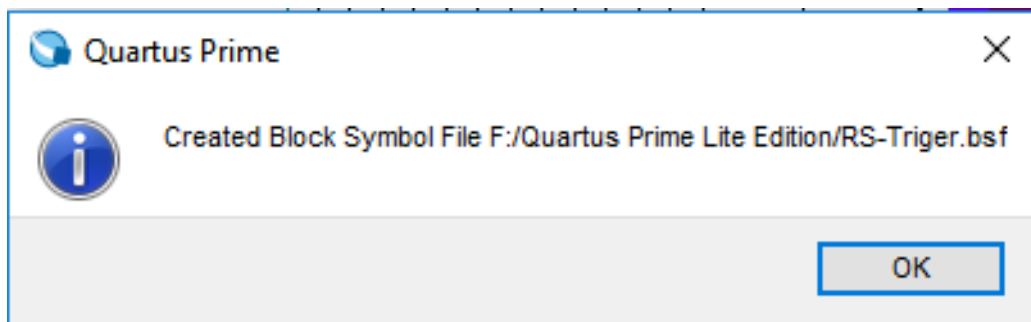
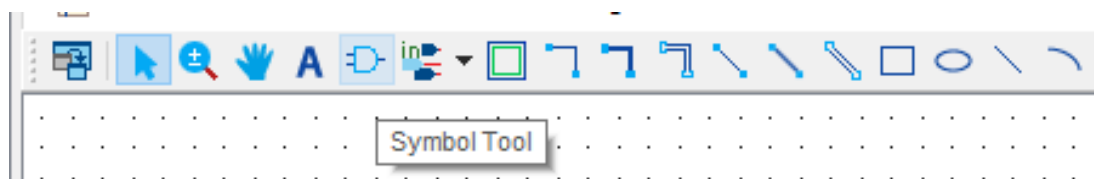


Рисунок 6.14 – Підтвердження збереження блоку RS-triger

Знаходимо в бібліотеці елементів створений тригер. Для цього натискаємо на панелі інструментів Symbol Tool, у вікні натискаємо Project в якому знаходиться тригер. Вибираємо його і користуємося у подальшому збиранні схеми.



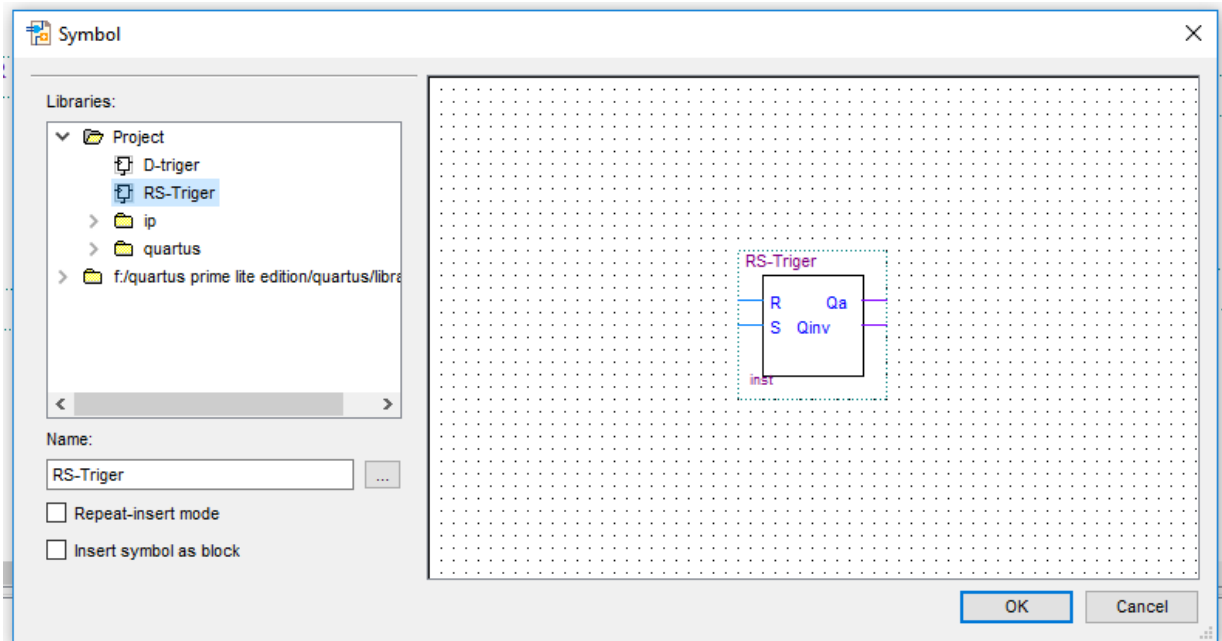


Рисунок 6.15 – Пошук елементу

Створення схеми проекту

Створюємо схему, для встановлення елементів – подвійний клік по робочому полю, у вікні, що відкрилося вибираємо вкладку Primitives -> Logic. Так само задаємо входи і виходи із вкладки Pins. Перед виконання призначення пінів слід компілювати проект

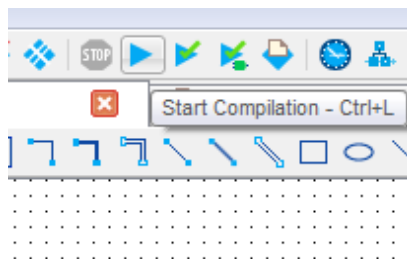


Рисунок 6.16 – Робоча область та панель інструментів

Після компіляції відкриваємо вкладку Assignments -> Pin planner. У графі Location починаємо вводити адресу необхідного піна, натискаємо на повну назву, коли вона з'явиться. Увага! Всі світлодіоди/кнопки/індикатори

заздалегідь під'єднані до певних контактів ПЛІС (див. User Manual), необхідно уважно вибирати адреси.

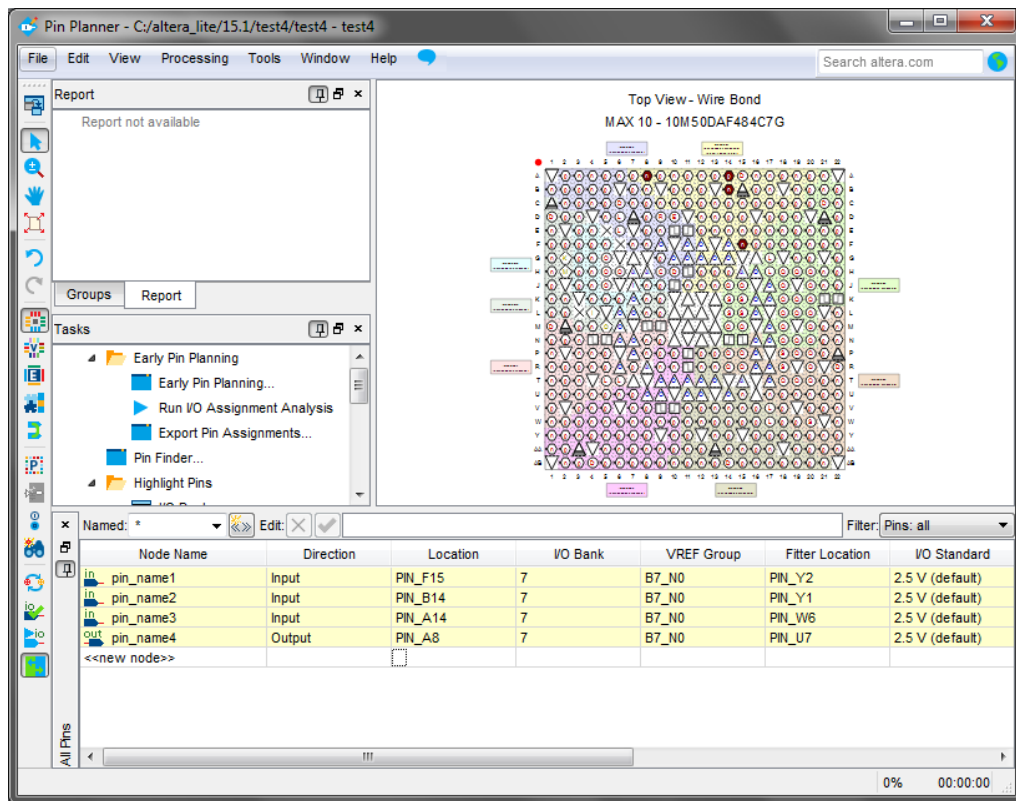


Рисунок 6.17 – Панель призначення виводів мікросхеми

Після цього знову компілюємо проект. На схемі біля виводів з'являється адреси. Можна програмувати ПЛІС.

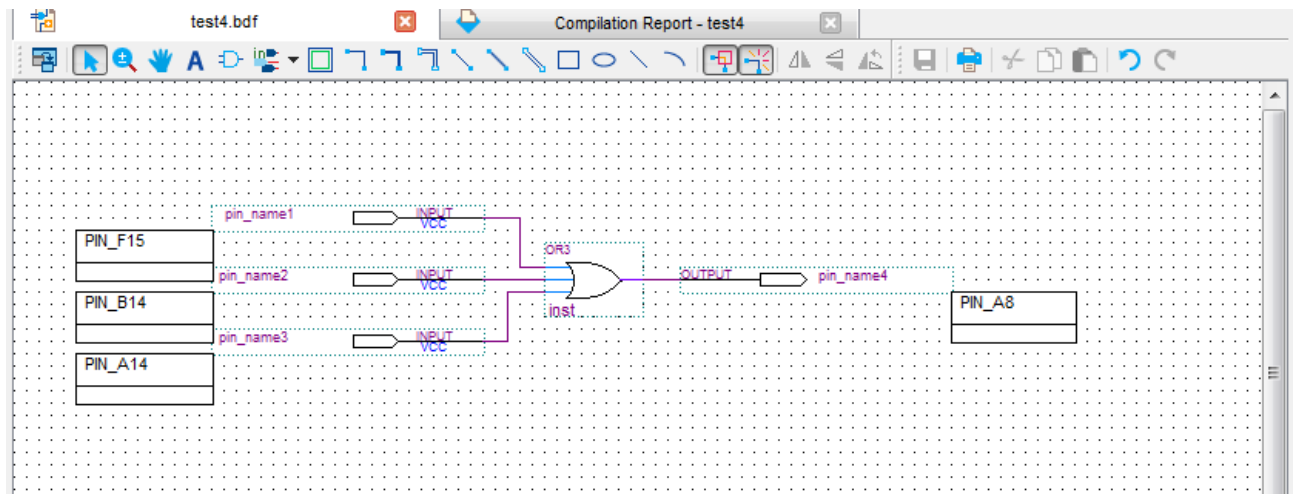


Рисунок 6.18 – Схема поточного проекту

На панелі зліва вибираємо вкладку Program Device, підключаємо плату до комп'ютера

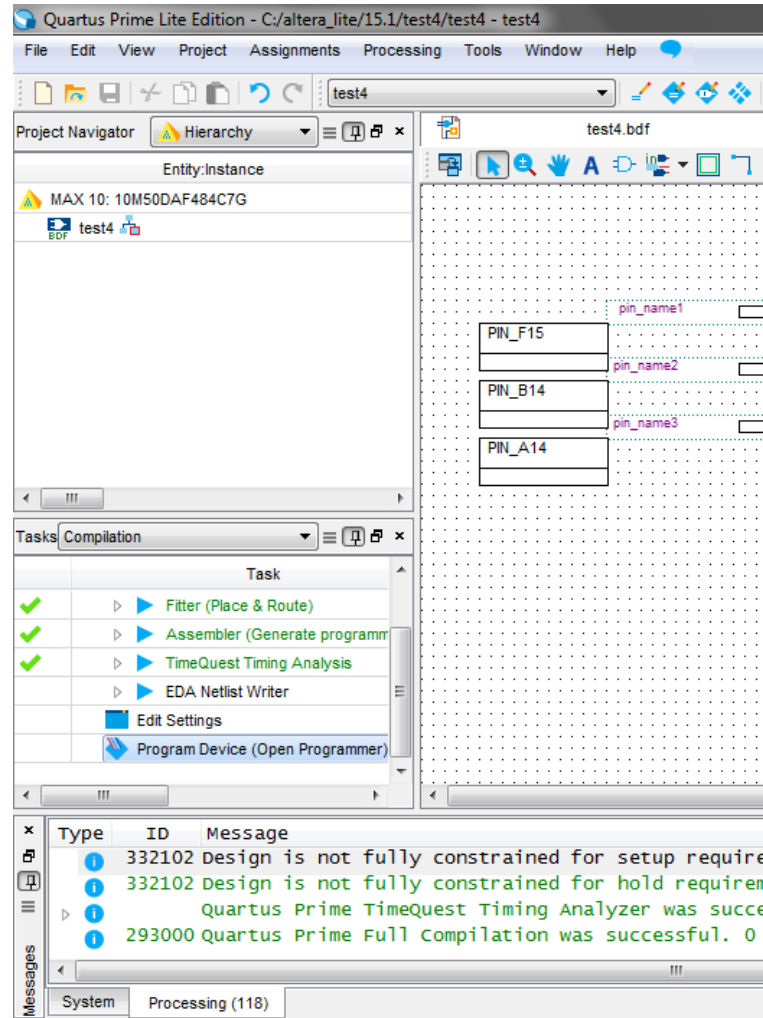


Рисунок 6.19 – Розташування вкладки Program Device

У вікні, що відкрилося, у вкладці Hardware вибираємо наш програматор:

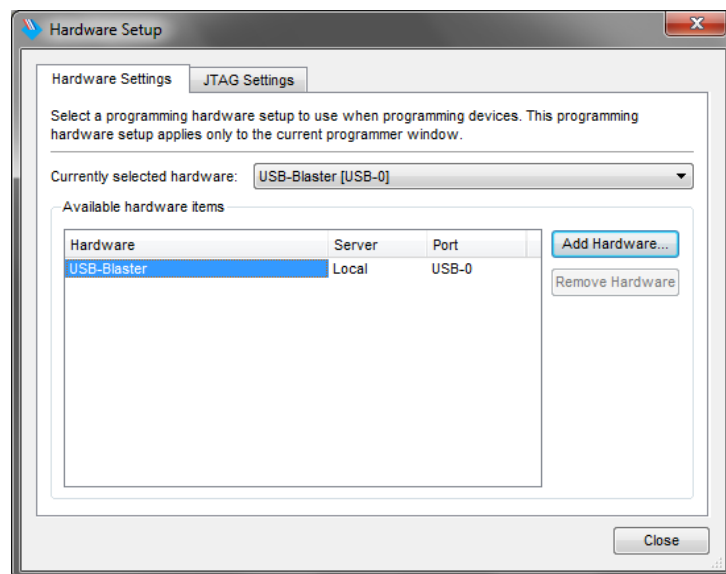


Рисунок 6.20 – Вікно вибору пристрою для програмування

У вкладці Autodetect вибираємо нашу мікросхему



Рисунок 6.21 – Вікно вибору пристрою

Її схематичне зображення з'явиться у вікні. Натискаємо на нього, та вибираємо на панелі зліва пункт Change file, вибираємо файл прошивки з розширенням .sof

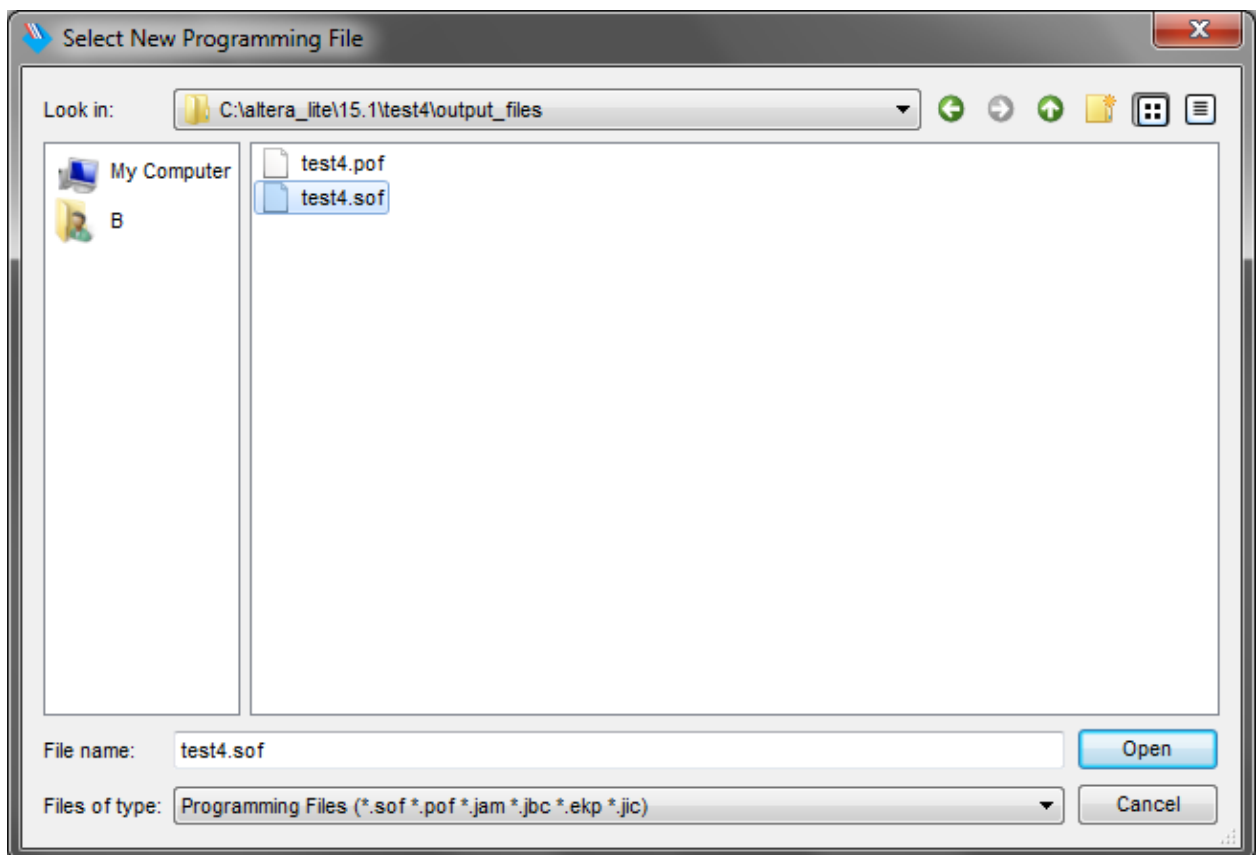


Рисунок 6.22 – Вікно вибору файлу

Натискаємо чекбокс Program/Configure, після цього натискаємо Start

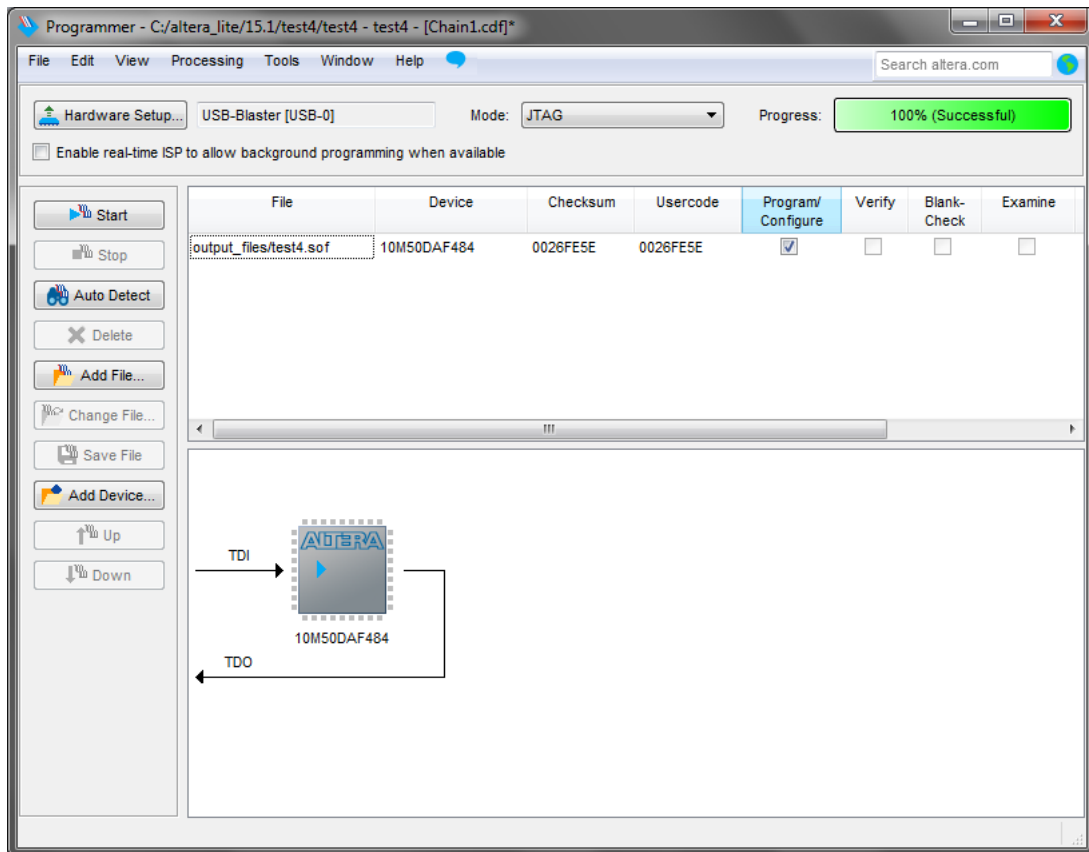


Рисунок 6.23 – Вікно параметрів програмування

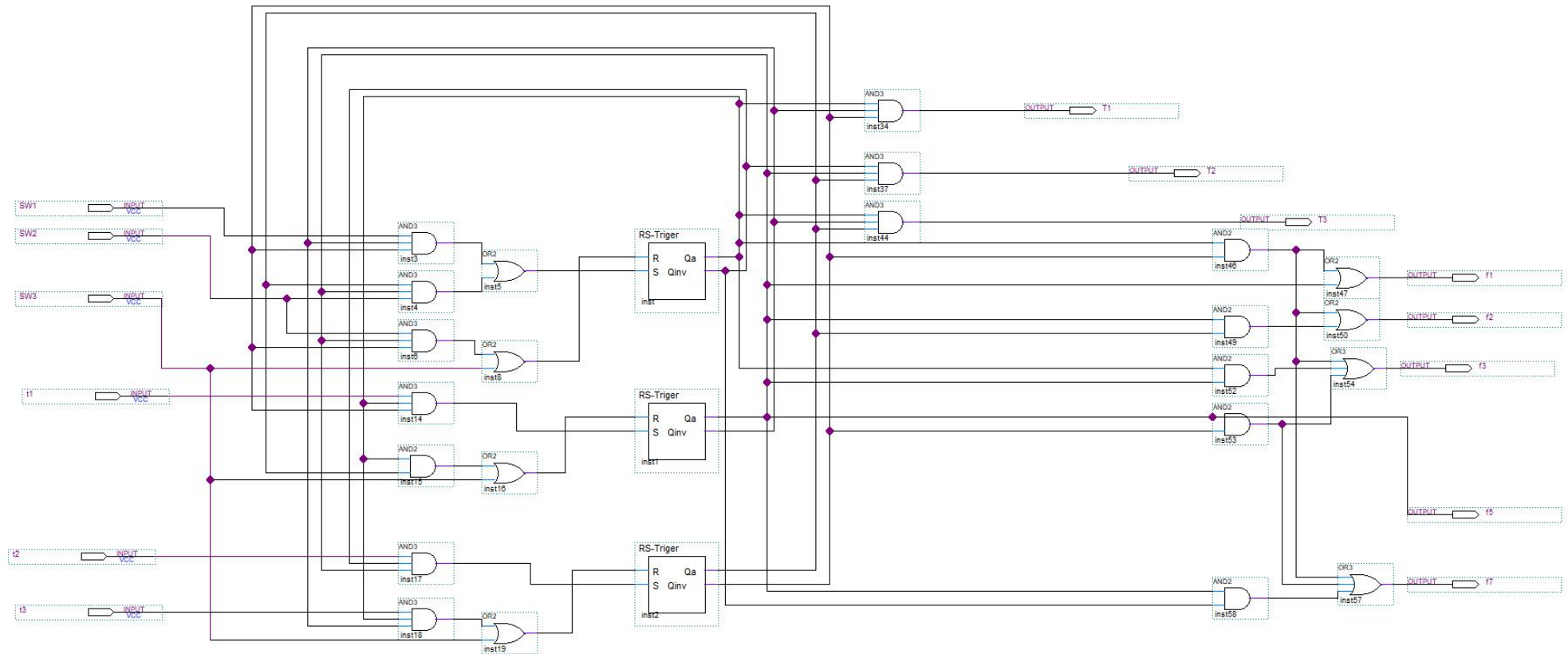


Рисунок 6.24 – Готова схема першого граф переходу в Quartus Prime Lite

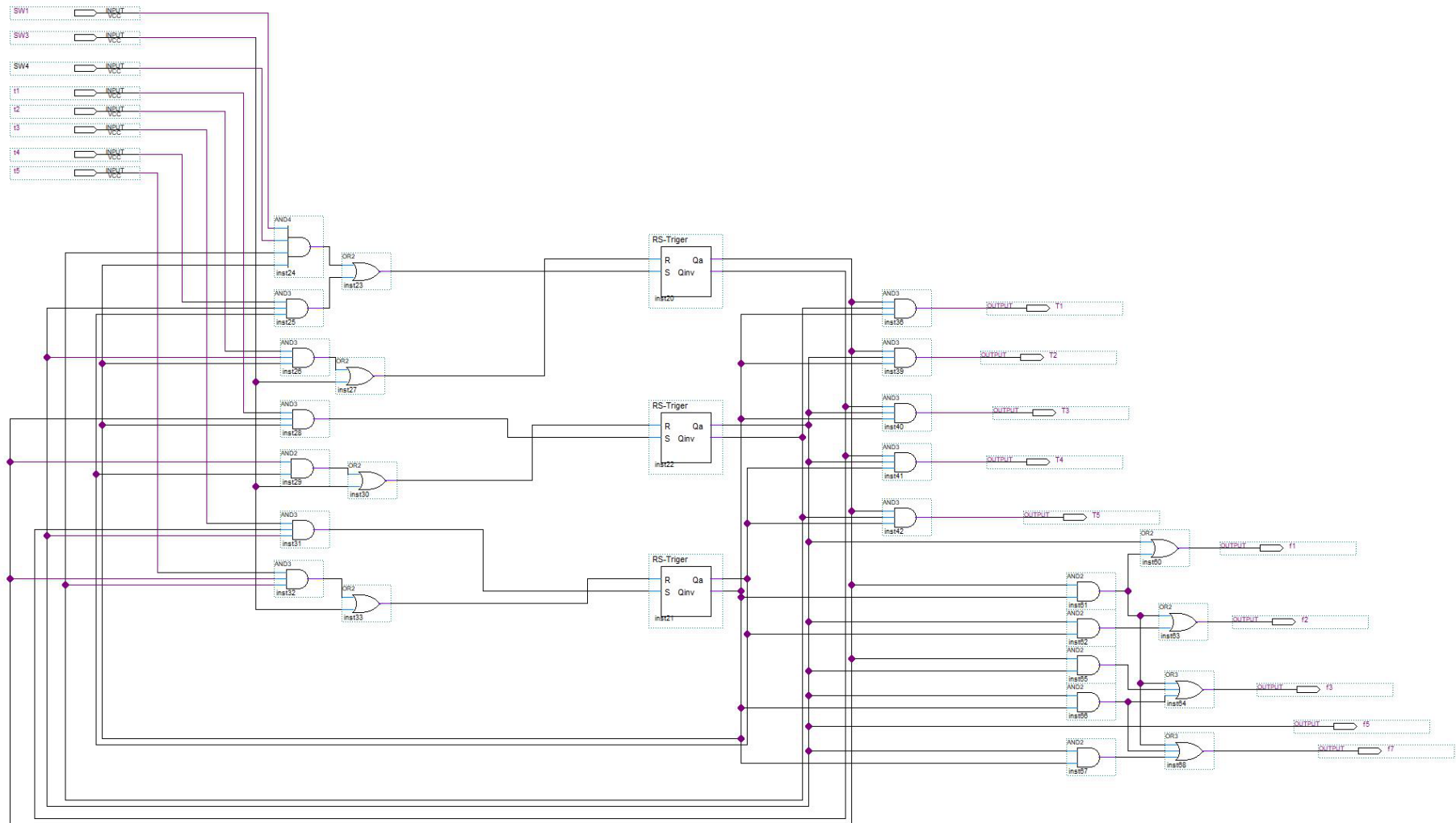


Рисунок 6.25 – Готова схема другого граф переходу в Quartus Prime Lite

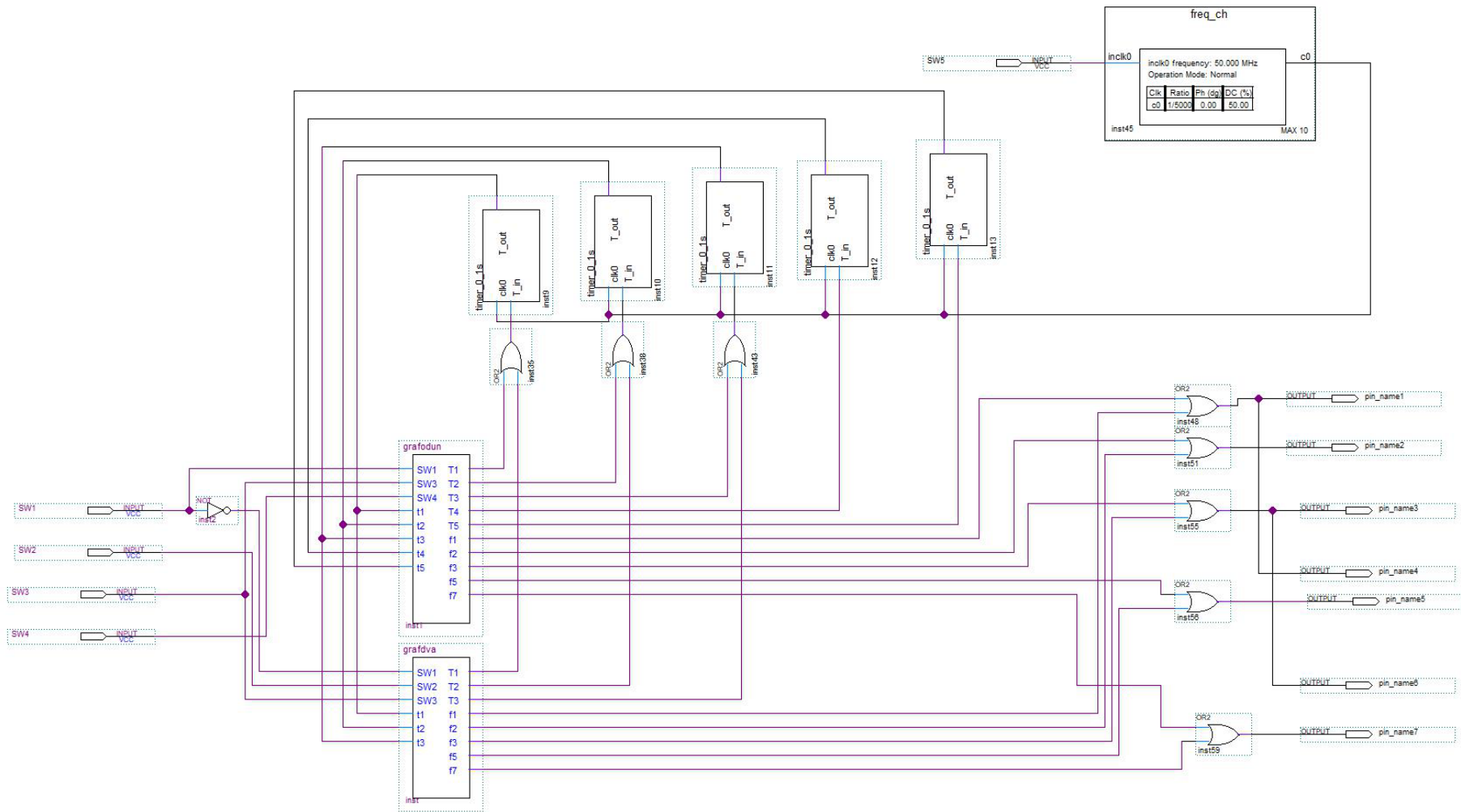


Рисунок 6.26 – Схема в зборі відповідно до завдання прикладу

6.4 Зміст звіту

- 1) Титульний аркуш.
- 2) Мета роботи, програма роботи.
- 3) Умови роботи схем.
- 4) Логічний синтез схем керування.
- 5) Реалізація завдання в Quartus Prime 15.1 Lite Edition.
- 6) Висновок про відповідність роботи заданим умовам.

6.5 Контрольні запитання

1. Що таке тригер? Які його особливості?
2. Що таке граф переходів? Як на основі графу переходів зобразити умови роботи схеми?
3. Як записуються умови включення тригера?
4. Як записуються умови скидання тригера?
5. Як на основі графу переходів записати вирази для вихідних змінних і для таймерів?

Лабораторна робота №7

ДОСЛІДЖЕННЯ ЛІЧИЛЬНИКІВ ІМПУЛЬСІВ

Тривалість лабораторної роботи – 4 год.

Тривалість домашньої роботи – 4 год.

Мета роботи – набути практичні вміння синтезувати схеми для систем автоматизації із використанням мікросхем лічильників, скласти відповідні програми на мові FBD у середовищі Quartus II та практично перевірити їх працездатність на основі плати розробника DE10-Lite.

7.1 Основні теоретичні відомості

Дана плата виконана на базі ПЛІС MAX 10 10M50DAF484C7G, яка відноситься до класу програмованих вентильних матриць. В даній ПЛІС міститься 50 000 програмованих логічних елементів.

До складу плати входить USB Blaster, та засоби вводу/виводу: 6 семисегментних індикаторів, 10 світлодіодів, 10 перемикачів, 2 кнопки, 3 годинника, акселерометр, модуль пам'яті SDRAM64MB, 36 входів/виходів загального призначення, Arduino Header, VGA.

З усього цього нам знадобляться кнопки, назви сигналів та відповідні піни для яких наведено у табл. 7.1. Ці кнопки будуть виконувати роль генератора імпульсів.

Таблиця 7.1 – Розпіновка кнопок

Назва сигналу	Розташування піна
KEY0	PIN_B8
KEY1	PIN_A7

Також знадобляться двопозиційні перемикачі. Відповідні сигнали та піни наведені у табл. 7.2. Ми будемо використовувати ці перемикачі для інших функції з вводу даних.

Таблиця 7.2 – Розпіновка перемикачів

Назва сигналу	Розташування піна
SW0	PIN_C10
SW1	PIN_C11
SW2	PIN_D12
SW3	PIN_C12
SW4	PIN_A12
SW5	PIN_B12
SW6	PIN_A13
SW7	PIN_A14
SW8	PIN_B14
SW9	PIN_F15

Вивід інформації буде здійснюватися на 10 світлодіодів, розпіновку яких можна побачити у табл. 7.3.

Таблиця 7.3 – Розпіновка світлодіодів

Назва сигналу	Розташування піна
LEDR0	PIN_A8
LEDR1	PIN_A9
LEDR2	PIN_A10
LEDR3	PIN_B10
LEDR4	PIN_D13
LEDR5	PIN_C13
LEDR6	PIN_E14
LEDR7	PIN_D14
LEDR8	PIN_A11
LEDR9	PIN_B11

та 6 семисегментних індикаторів, сигнали та піни яких показано у табл. 7.4.

Таблиця 7.4 – Розпіновка семисегментних індикаторі

Назва сигналу	Розташування піна
HEX00	PIN_C14
HEX01	PIN_E15
HEX02	PIN_C15
HEX03	PIN_C16
HEX04	PIN_E16
HEX05	PIN_D17
HEX06	PIN_C17
HEX07	PIN_D15
HEX10	PIN_C18
HEX11	PIN_D18
HEX12	PIN_E18
HEX13	PIN_B16
HEX14	PIN_A17
HEX15	PIN_A18
HEX16	PIN_B17
HEX17	PIN_A16
HEX20	PIN_B20
HEX21	PIN_A20
HEX22	PIN_B19
HEX23	PIN_A21
HEX24	PIN_B21
HEX25	PIN_C22
HEX26	PIN_B22
HEX27	PIN_A19
HEX30	PIN_F21

HEX31	PIN_E22
HEX32	PIN_E21
HEX33	PIN_C19
HEX34	PIN_C20
HEX35	PIN_D19
HEX36	PIN_E17
HEX37	PIN_D22
HEX40	PIN_F18
HEX41	PIN_E20
HEX42	PIN_E19
HEX43	PIN_J18
HEX44	PIN_H19
HEX45	PIN_F19
HEX46	PIN_F20
HEX47	PIN_F17
HEX50	PIN_J20
HEX51	PIN_K20
HEX52	PIN_L18
HEX53	PIN_N18
HEX54	PIN_M20
HEX55	PIN_N19
HEX56	PIN_N20
HEX57	PIN_L19

Знайомство з мікросхемами серії 74

В даній лабораторній роботі, при складанні (розробці) схеми на мові FBD у відповідності до попередньо виконаного синтезу, основний уклон зроблено на застосування студентами мікросхем серії 74 які присутні в бібліотеці Quartus Prime.

Такий метод виконання лабораторної роботи призначений для надання студентам інформації та набуття ними навичок щодо використання уніфікованих стандартів, а також зменшення часу розробки та зниження кількості помилок під-час будування схем.

В лабораторній роботі будуть приведені всі потрібні мікросхеми для успішного її виконання, більш детально з призначенням мікросхем можна ознайомитись на сайті: https://uk.wikipedia.org/wiki/Мікросхеми_серії_74xx.

З конструкцією мікросхеми та її описом, а також основними властивостями та описом роботи в тому чи іншому режимі можна ознайомитись на сайті: https://en.wikipedia.org/wiki/List_of_7400-series_integrated_circuits.

Мікросхема 7493

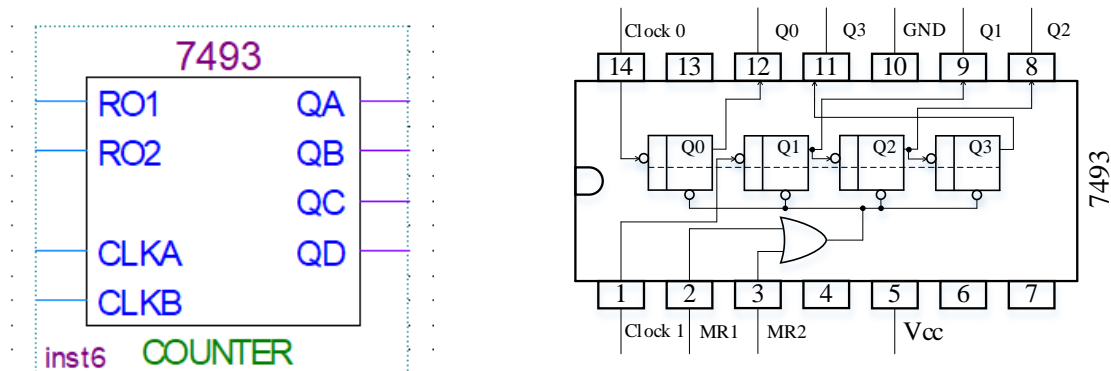


Рисунок 7.1 – Мікросхема 7493

Мікросхема 7493 складається з чотирьох тригерів, внутрішньо пов'язаних між собою таким чином, що утворюються два лічильника-дільника: один на 2 і іншої на 8.

Всі тригери мікросхеми 7493 мають один спільний вхід скидання, за допомогою якого показання лічильників можуть встановлюватися в "0" в будь-який час (на виводи 2 і 3 подається високий рівень напруги).

Перший тригер мікросхеми 7493 внутрішньо не пов'язаний з іншими ступенями, завдяки чому можливі такі варіанти використання:

1. Рахунок до 16. Вихід Q0 (QA) з'єднується з входом тактових імпульсів Clock1 (CLKB). Вхідний сигнал подається на вхід Clock0 (CLKA), а вихідний сигнал знімається з виходу Q3 (QD). Ця схема рахує в двійковому коді до 16 (0 - 15) і на 16-му імпульсі повертається в нульовий стан;

2. Рахунок до 2 і рахунок до 8. У цьому випадку перший тригер використовується в якості подільника на 2, а другий, третій і четвертий тригери - як дільник на 8.

Перемикання мікросхеми 7493 завжди відбувається по негативному фронту тактового імпульсу. У нормальному режимі роботи на один з двох входів скидання MR1 (RO1) або MR2 (RO2) має подаватися напруга низького рівня (0), при рахунку до 16 – на обидва входи подається 0.

Мікросхема 74193

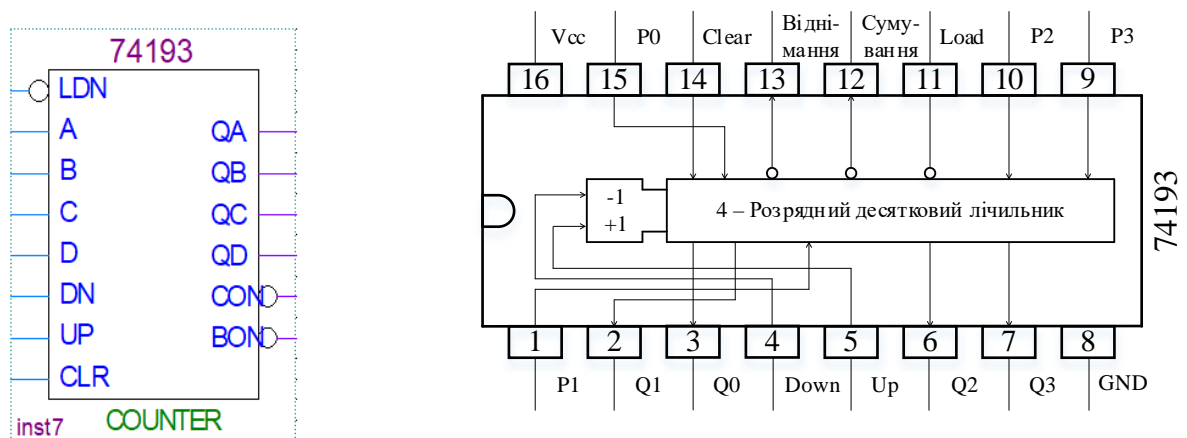


Рисунок 7.2 – Мікросхема 74193

Мікросхема 74193 містить синхронний програмований 4-розрядний двійковий лічильник з окремими входами тактових імпульсів для рахунку імпульсів в прямому і зворотному напрямках, а також з входом скидання.

На входи P0-P3 (A-D) подається чотирьох-бітний двійковий код, який визначає початкове значення лічильника. Подальший відлік буде відбуватися від цього числа. Але окрім цього на вхід Load (LDN) – подають короткочасний імпульс напруги низького рівня (0). З-зі входів Q0-Q3 (QA-QD) знімається код 4 – розрядного двійкового значення числа.

Щоб скинути показники лічильника 74193 на вхід Clear (CLR) подається короткочасний імпульс напруги високого рівня (1). Процес скидання показань відбувається незалежно від тактових імпульсів.

Значення, що зберігається в лічильнику 74193, послідовно збільшується на 1 при кожному перепаді напруги на вході прямого рахунку тактових імпульсів Up (UP) з низького рівня на високий (позитивний фронт). Кожен позитивний фронт тактового імпульсу на вході зворотного рахунку Down (DN) зменшує показання лічильника. У будь-якому випадку на один з двох входів тактових імпульсів мікросхеми 74193 має подаватися напруга високого рівня (1).

При прямому рахунку при досягненні значення 15 на виході перенесення 12 (CON) виникає негативний імпульс. При рахунку в зворотному напрямку при досягненні значення 0 на виході 13 (BON) з'являється короткочасний негативний імпульс.

Для створення багатоступінчастого лічильника з мікросхеми 74193 виходу 13 (BON, перенесення в зворотному напрямку) з'єднують з входом зворотного рахунку Down (DN) наступному ступені, а виходу 12 (CON, перенесення в прямому напрямку) - з входом прямого рахунку Up (UP) наступному ступені.

Мікросхема 74154

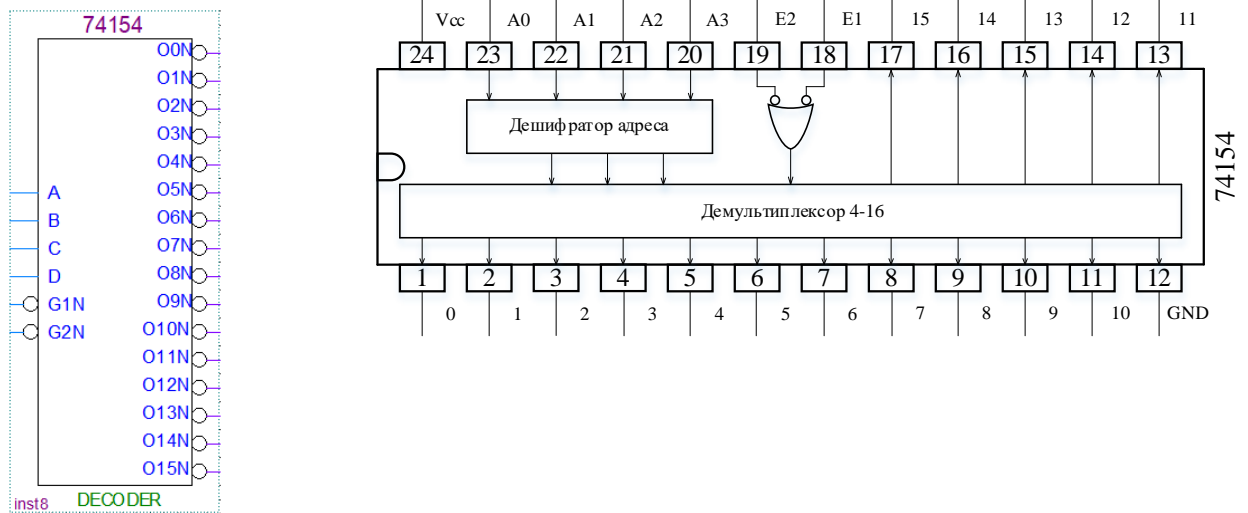


Рисунок 7.3 – Мікросхема 74154

Мікросхема 74154 перетворює 4-розрядний двійковий код в сигнал низького рівня на одному з 16 виходів.

Коли 4-розрядний двійковий код надходить на адресні входи мікросхеми 74154 A0 - A3 (A-D), то на відповідному виході встановлюється напруга низького рівня, а на інших виходах - високого.

Однак це відбувається лише в тому випадку, якщо на обидва входи дозволу мікросхеми 74154 E1 і E2 (G1N і G2N) подається напруга низького рівня (0). Якщо на один або обидва входи дозволу подається напруга високого рівня, таку напругу встановлюється і на всіх виходах.

Якщо на один з входів дозволу мікросхеми 74154 подається напруга низького рівня (0), а інший розглядається як інформаційний вхід, то обраний через адресні входи вихід матиме той же логічний рівень, що і другий вихід. Таким чином, мікросхема 74154 використовується в якості демультіплексора або багатоканального комутатора даних. При цьому якщо на обидва входи E1 і E2 (G1N і G2N) подати напругу високого рівня (1), то на виходах сигнал зникне, але потім при зміні високого рівня напруги на низький сигнал на виходах знову з'явиться.

Мікросхема 7402

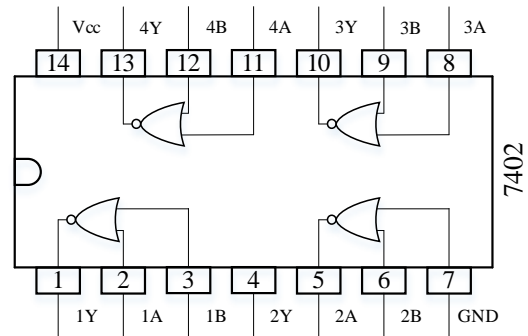
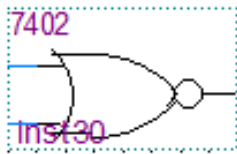


Рисунок 7.4 – Мікросхема 7402

Мікросхема 7402 містить чотири окремих логічних елемента АБО-НІ з двома входами кожен.

При подачі напруги високого рівня на один або обидва входи елемента мікросхеми 7402 на виході встановлюється напруга низького рівня. Якщо на обидва входи подається напруга низького рівня, то на виході встановлюється напруга високого рівня.

Мікросхема 7404

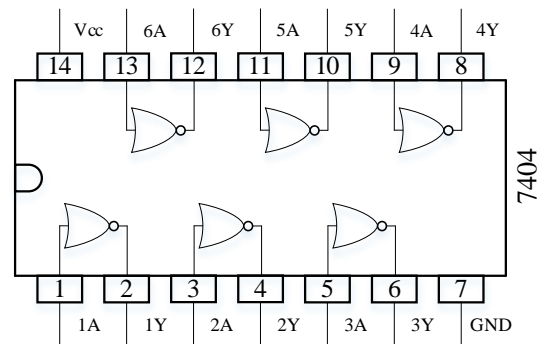
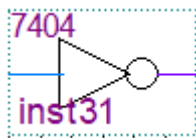


Рисунок 7.5 – Мікросхема 7404

Мікросхема 7404 містить шість окремих інверторів. При подачі напруги низького рівня на вхід кожного з них на виході мікросхеми 7404 встановлюється напруга високого рівня, і навпаки.

Мікросхема 7408

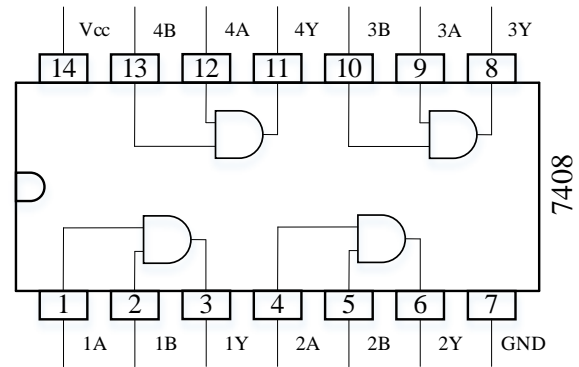
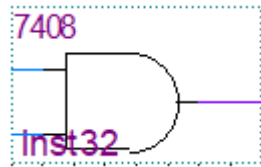


Рисунок 7.6 – Мікросхема 7408

Мікросхема 7408 містить чотири окремих логічних елемента І з двома входами кожен. При подачі напруги низького рівня на один або обидва входи кожного елемента на виході мікросхеми 7408 встановлюється напруга низького рівня. Якщо на обидва входи подається напруга високого рівня, то на виході формується напруга високого рівня.

Мікросхема 7411

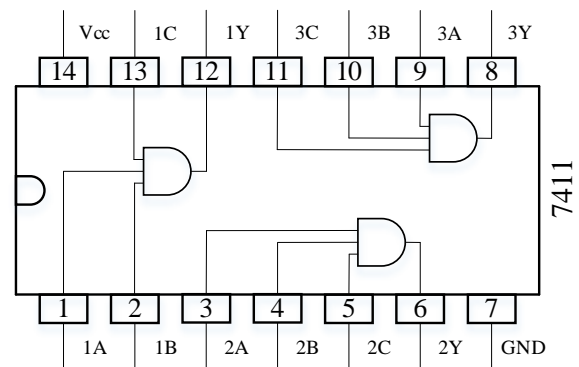
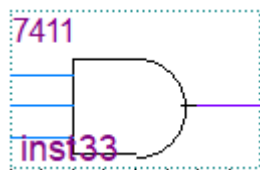


Рисунок 7.7 – Мікросхема 7411

Мікросхема 7411 містить три окремих логічних елемента І, у кожного з яких по три входи.

При подачі напруги низького рівня на один або кілька входів кожного елемента мікросхеми 7411 на виході встановлюється напруга низького рівня.

Якщо на всі входи подається напруга високого рівня, то і на виході формується напруга високого рівня.

Мікросхема 7432

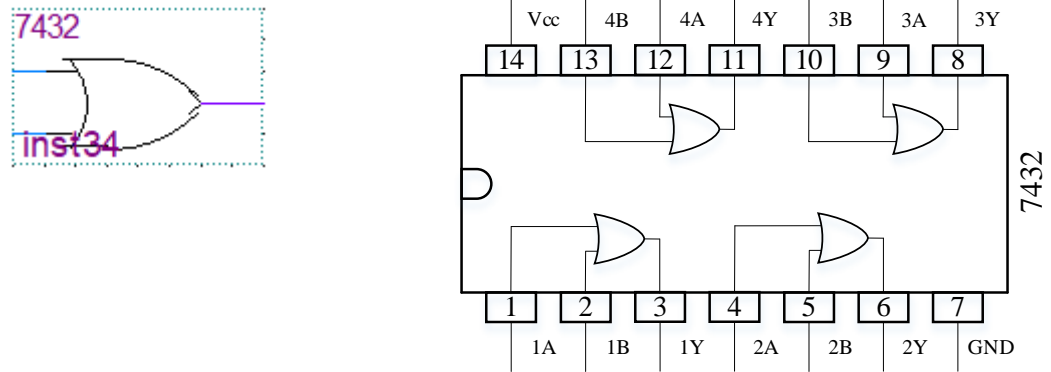


Рисунок 7.8 – Мікросхема 7432

Мікросхема 7432 містить чотири окремих логічних елемента АБО з двома входами на кожному.

При подачі напруги високого рівня на один або обидва входи кожного елемента мікросхеми 7432 на виході встановлюється напруга високого рівня. Якщо на обидва входи подається напруга низького рівня, то на виході формується напруга низького рівня.

7.2 Програма роботи

1. Ознайомитися з функціональною схемою плати та розуміти призначення кожного елемента та їх розташування.
2. Ознайомитися з приведеними прикладами мікросхем 74 серії.
3. За таблицею 5 обрати свій варіант.
4. Виконати логічний синтез завдань за вказаним методом.
5. У програмному середовищі Quartus Prime скласти на мові FBD програму, яка вирішує поставлене завдання.
6. Перевірити правильність виконання завдання, виконати симуляцію роботи.
7. Оформити протокол по лабораторній роботі та зробити висновки

Номер бригади	Номер завдання
1	1, 12, 13
2	2, 11, 14
3	3, 10, 15
4	4, 9, 16
5	5, 8, 17
6	6, 7, 18

7.3 Завдання до лабораторної роботи

На основі лічильників

1. Розробити та впровадити схему (на основі мікросхем 7493 серії), яка послідовно спалахує цифрами від 0 до 16 на 7-сегментному дисплеї. Кожна цифра повинна відображатися на дисплеї, а при надходженні наступного імпульсу на лічильник змінюватись.

2. Розробити та впровадити схему (на основі мікросхем 74193 серії), яка послідовно спалахує цифрами від 0 до 16 на 7-сегментному дисплеї при надходженні імпульсів прямої послідовності. Кожна цифра повинна відображатися на дисплеї, а при надходженні наступного імпульсу на лічильник змінюватись.

3. Розробити та впровадити схему (на основі мікросхем 74193 серії), яка послідовно спалахує цифрами від 16 до 0 на 7-сегментному дисплеї при надходженні імпульсів зворотної послідовності. Кожна цифра повинна відображатися на дисплеї, а при надходженні наступного імпульсу на лічильник змінюватись.

4. Розробити та впровадити схему (на основі мікросхем 74193 серії), яка послідовно спалахує цифрами від 16 до 0 та від 0 до 16 на 7-сегментному дисплеї при надходженні імпульсів як прямої так і зворотної послідовності. Кожна цифра повинна відображатися на дисплеї, а при надходженні наступного імпульсу на лічильник змінюватись.

5. Розробити та впровадити схему (на основі мікросхем 74193 серії), яка послідовно спалахує цифрами від 6 (з попереднім встановленням) до 16 на 7-сегментному дисплеї при надходженні імпульсів прямої послідовності. Причому при знаходженні 16 імпульсу на 7-сегментному дисплеї повинна знову з'являтися цифра 6, тобто та що була вибрана в якості попередньо встановленої. Кожна цифра повинна відображатися на дисплеї, а при знаходженні наступного імпульсу на лічильник змінюватись.

6. Розробити та впровадити схему (на основі мікросхем 74193 серії), яка починає відлік від довільно вибраного числа (n) в попередній установці. Необхідно продемонструвати роботу лічильника в прямій послідовності – від (n) до 16, та в зворотній послідовності від (n) до 0. Кожна цифра повинна відображатися на дисплеї, а при надходженні наступного імпульсу на лічильник - змінюватись. Число n рекомендується вибирати в межах від 3 до 8.

Методом граф переходів на основі RS – тригерів та лічильників

7. Виконати автоматизацію стенду по намотці витків для обмотки двигуна на основі мікросхеми 7493. Схема має 4 кнопки: «Пуск», «Педаль» (Пуск2), «Стоп», «Кінцевий вимикач» і 3 вихідних функції: «Мотор», «Механічне гальмо», «Обнулення лічильника» Після надходження сигналу «Пуск» знімається механічне гальмо і обнуляється лічильник. Після натиснення оператора ногою на кнопку «Педаль» вмикається двигун який через редуктор, починає намотувати на спеціальне рамку витки обмотки. На рамці розташований спеціальний бойок, який після кожного нового кола (витка обмотки) на короткий час натискає на кінцевий вимикач. Рамка повинна здійснити 3 обертів, після чого двигун повинен зупинитись. Після зупинки двигуна оператор натискає кнопку «Стоп» і система повертається в вихідне положення.

8. Виконати автоматизацію стенду по намотці витків для обмотки двигуна на основі мікросхеми 7493. Схема має 4 кнопки: «Пуск», «Педаль» (Пуск2), «Стоп», «Кінцевий вимикач» і 3 вихідних функції: «Мотор», «Механічне гальмо», «Значення лічильника в даний момент». Після надходження сигналу «Пуск» знімається механічне гальмо і обнуляється

лічильник. Після натиснення оператора ногою на кнопку «Педадь» вмикається двигун який через редуктор, починає намотувати на спеціальне рамку витки обмотки. На рамці розташований спеціальний бойок, який після кожного нового кола (витка обмотки) на короткий час натискає на кінцевий вимикач. Рамка повинна здійснити 10 обертів, після чого двигун повинен зупинитись. Після зупинки двигуна оператор натискає кнопку «Стоп» і система повертається в вихідне положення.

9. Виконати автоматизацію стенду по намотці витків для обмотки двигуна на основі мікросхеми 74193. Схема має 4 кнопки: «Пуск», «Педадь» (Пуск2), «Стоп», «Кінцевий вимикач» і 3 вихідних функції: «Мотор», «Механічне гальмо», «Обнулення лічильника». Після надходження сигналу «Пуск» знімається механічне гальмо і обнуляється лічильник. Після натиснення оператора ногою на кнопку «Педадь» вмикається двигун який через редуктор, починає намотувати на спеціальне рамку витки обмотки. На рамці розташований спеціальний бойок, який після кожного нового кола (витка обмотки) на короткий час натискає на кінцевий вимикач. Рамка повинна здійснити 5 обертів (при цьому число 5 повинно бути попередньо встановленої, а лічильник зменшуватись на 1), після чого двигун повинен зупинитись. Після зупинки двигуна оператор натискає кнопку «Стоп» і система повертається в вихідне положення.

10. Виконати автоматизацію стенду по намотці витків для обмотки двигуна на основі мікросхеми 74193. Схема має 4 кнопки: «Пуск», «Педадь» (Пуск2), «Стоп», «Кінцевий вимикач» і 3 вихідних функції: «Мотор», «Механічне гальмо», «Значення лічильника в даний момент». Після надходження сигналу «Пуск» знімається механічне гальмо і обнуляється лічильник. Після натиснення оператора ногою на кнопку «Педадь» вмикається двигун який через редуктор, починає намотувати на спеціальне рамку витки обмотки. На рамці розташований спеціальний бойок, який після кожного нового кола (витка обмотки) на короткий час натискає на кінцевий вимикач. Рамка повинна здійснити 4 обертів (при цьому число 4 повинно бути попередньо встановленої, а лічильник зменшуватись на 1), після чого двигун

повинен зупинитись. Після зупинки двигуна оператор натискає кнопку «Стоп» і система повертається в вихідне положення.

11. Виконати автоматизацію стенду по намотці витків для обмотки двигуна на основі мікросхеми 74193. Схема має 4 кнопки: «Пуск», «Педаль» (Пуск2), «Стоп», «Кінцевий вимикач» і 3 вихідних функції: «Мотор», «Механічне гальмо», «Обнулення лічильника». Після надходження сигналу «Пуск» знімається механічне гальмо і обнуляється лічильник. Після натиснення оператора ногою на кнопку «Педаль» вмикається двигун який через редуктор, починає намотувати на спеціальне рамку витки обмотки. На рамці розташований спеціальний бойок, який після кожного нового кола (витка обмотки) на короткий час натискає на кінцевий вимикач. Рамка повинна здійснити 2 обертів (при цьому число 14 повинно бути попередньо встановленої, а лічильник збільшуватись на 1), після чого двигун повинен зупинитись. Після зупинки двигуна оператор натискає кнопку «Стоп» і система повертається в вихідне положення.

12. Виконати автоматизацію стенду по намотці витків для обмотки двигуна на основі мікросхеми 74193. Схема має 4 кнопки: «Пуск», «Педаль» (Пуск2), «Стоп», «Кінцевий вимикач» і 3 вихідних функції: «Мотор», «Механічне гальмо», «Значення лічильника в даний момент». Після надходження сигналу «Пуск» знімається механічне гальмо і обнуляється лічильник. Після натиснення оператора ногою на кнопку «Педаль» вмикається двигун який через редуктор, починає намотувати на спеціальне рамку витки обмотки. На рамці розташований спеціальний бойок, який після кожного нового кола (витка обмотки) на короткий час натискає на кінцевий вимикач. Рамка повинна здійснити 4 обертів (при цьому число 12 повинно бути попередньо встановленої, а лічильник збільшуватись на 1), після чого двигун повинен зупинитись. Після зупинки двигуна оператор натискає кнопку «Стоп» і система повертається в вихідне положення.

На основі лічильників на мікросхем 74 серії

13. Впровадити схему з безперервним рухом цифри (яка відповідає номеру вашої бригади) в праву сторону на 7 – сегментних індикаторах.

Початковий вихідний стан індикаторів 0, при однократному натисканні на кнопку з'являється цифра, замість нуля. Рух цифри продовжується в заданому напрямку через кожен наступний індикатор. При цьому там де була цифра – повинен знову з'явитись нуль. За основу схеми взяти мікросхему 7493.

14. Впровадити схему з безперервним рухом цифри (яка відповідає номеру вашої бригади) в ліву сторону на 7 – сегментних індикаторах. Початковий вихідний стан індикаторів «Крапка», при двократному натисканні на кнопку з'являється цифра, замість крапки. Рух цифри продовжується в заданому напрямку через кожен наступний індикатор. При цьому там де була цифра – повинна знову з'явитись «Крапка». За основу схеми взяти мікросхему 7493.

15. Впровадити схему з безперервним рухом цифри (яка відповідає номеру вашої бригади) в праву сторону на 7 – сегментних індикаторах. Початковий вихідний стан лівого індикатора 0, а кожного наступного на 1 більший (від 0 до 5). При однократному натисканні на кнопку з'являється цифра вашого варіанту, замість початкових значень цифри. Рух цифри продовжується в заданому напрямку через кожен наступний індикатор. При цьому там де була цифра – повинна знову з'явитись початкове значення. За основу схеми взяти мікросхему 74193.

16. Впровадити схему з безперервним рухом цифри (яка відповідає номеру вашої бригади) в ліву сторону на 7 – сегментних індикаторах. Початковий вихідний стан правого індикатора 0, а кожного наступного на 1 більший (від 0 до 5). При однократному натисканні на кнопку з'являється цифра вашого варіанту, замість початкових значень цифри. Рух цифри продовжується в заданому напрямку через кожен наступний індикатор. При цьому там де була цифра – повинна знову з'явитись початкове значення. За основу схеми взяти мікросхему 74193.

17. Впровадити схему з безперервним рухом цифри (яка відповідає порядковому номеру місяця вашого народження) в обидві сторони на 7 – сегментних індикаторах. Початковий вихідний стан правого індикатора 0, а кожного наступного на 1 більший (від 0 до 5). При однократному натисканні на

кнопку з'являється цифра вашого варіанту, замість початкових значень цифри. Рух цифри продовжується в заданому напрямку через кожен наступний індикатор. При цьому там де була цифра – повинна знову з'явитись початкове значення. За основу схеми взяти мікросхему 74193.

18. Впровадити схему з безперервним рухом цифри (яка відповідає номеру вашої лабораторної роботи) в обидві сторони на 7 – сегментних індикаторах. Початковий вихідний стан лівого індикатора 0, а кожного наступного на 1 більший (від 0 до 5). При однократному натисканні на кнопку з'являється цифра вашого варіанту, замість початкових значень цифри. Рух цифри продовжується в заданому напрямку через кожен наступний індикатор. При цьому там де була цифра – повинна знову з'явитись початкове значення. За основу схеми взяти мікросхему 74193.

7.4 Методичні вказівки до виконання лабораторної роботи

Приклад 1. Розробити та впровадити схему (на основі мікросхем 74 серії), яка послідовно спалахує цифрами від 6 (з попереднім встановленням) до 16 на 7-сегментному дисплеї при надходженні імпульсів прямої послідовності. Причому при надходженні 0 імпульсу на 7-сегментному дисплеї повинна знову з'являтися цифра 6, тобто та що була вибрана в якості попередньо встановленої. Кожна цифра повинна відображатися на дисплеї, а при надходженні наступного імпульсу на лічильник змінюватись.

1. Для виконання поставленого завдання в нас є всі необхідні мікросхеми, окрім тієї, яка б безпосередньо перетворила сигнал на виході 74154 в код для 7- сегментного індикатора. Схема такого блоку показана на рис. 7.9.

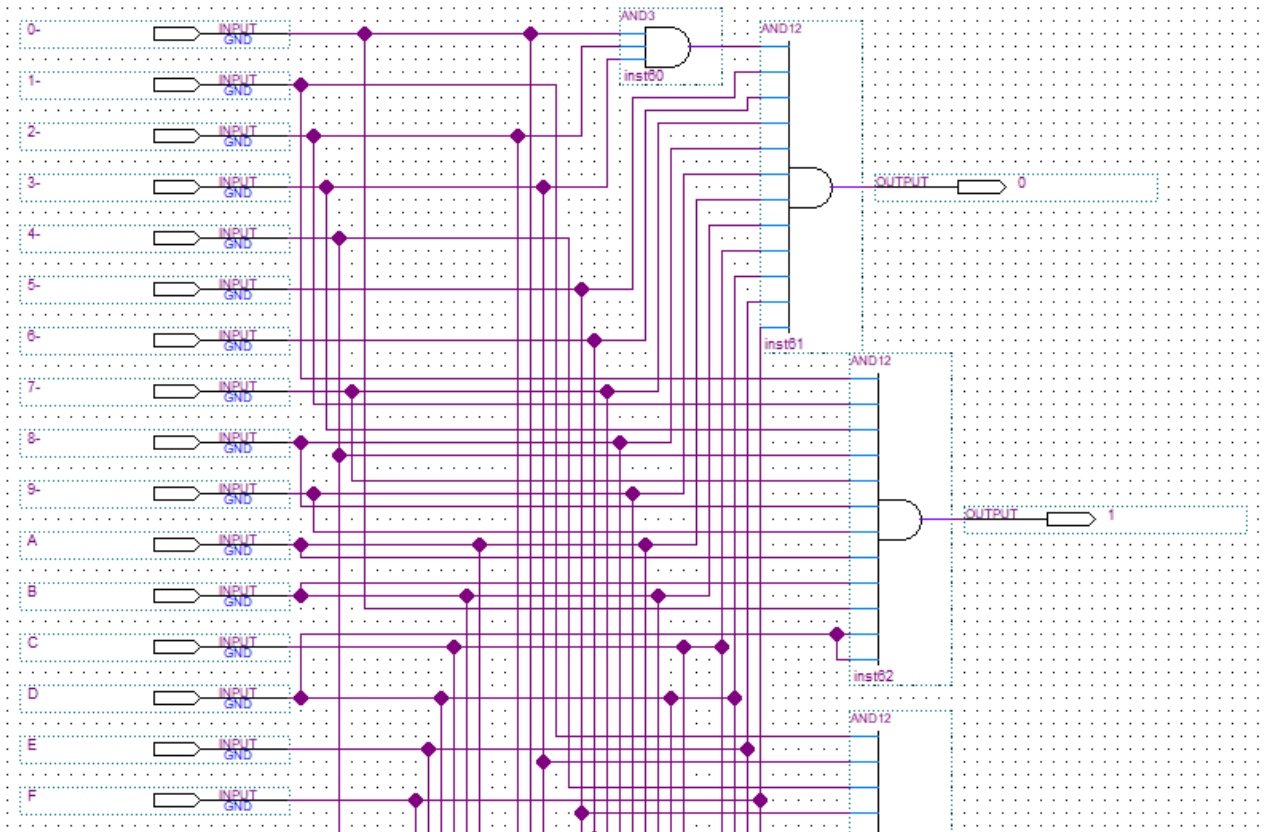


Рисунок 7.9 – Перетворювач коду для семи-сегментного індикатора

Як видно на малюнку, використання функцій АБО розподіляє всі сигнали з виходу 74154 на відповідні входи сегментів 7- сегментного індикатора.

2. Далі для зручності виділяємо всю схему що вийшла і створюємо новий блок для нашої бібліотеки рис. 7.10.

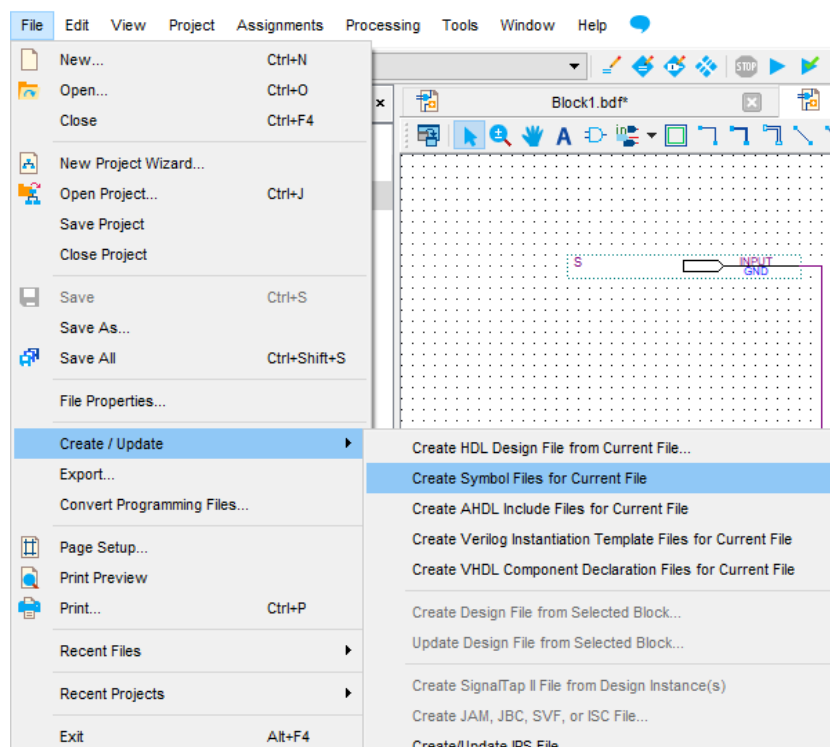


Рисунок 7.10 – Створення блоку

3. Далі переходимо в нашу бібліотеку і шукаємо збережений файл (рис. 7.11). На цьому ж рисунку показано папку в якій зберігаються мікросхеми 74 серії.

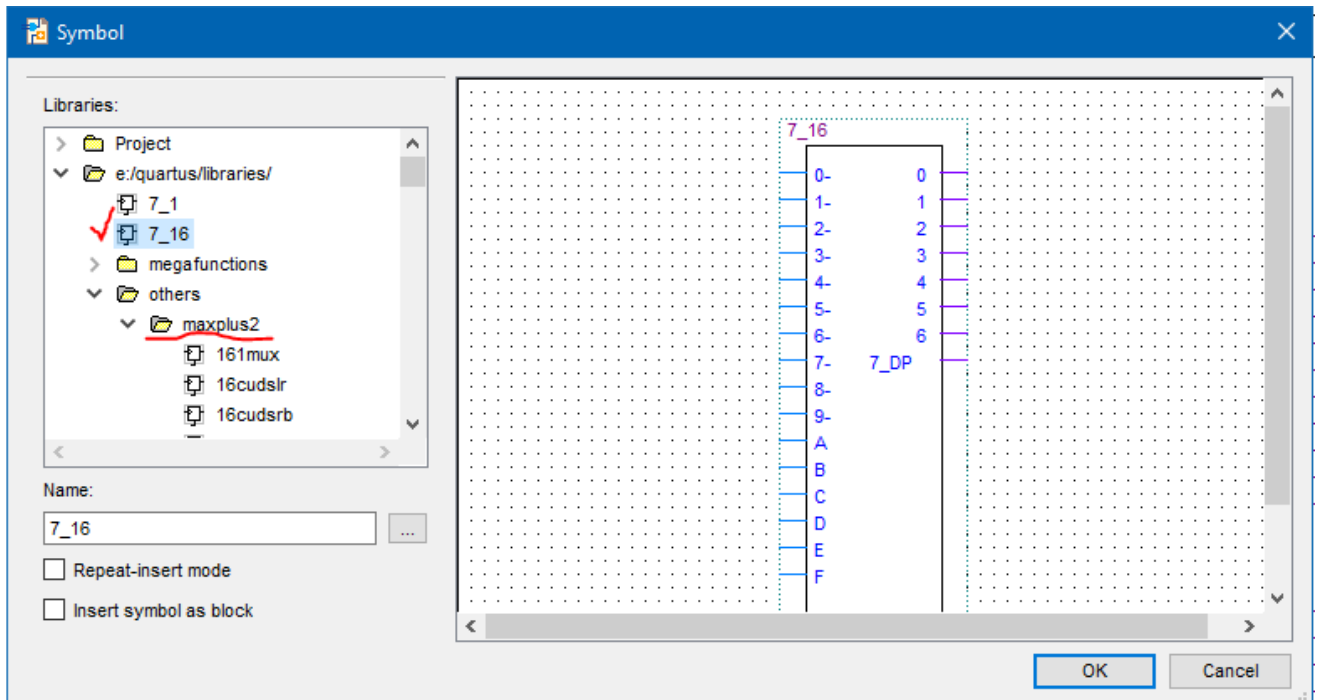


Рисунок 7.11 – Бібліотека з мікросхемами.

4. Далі створюємо новий проект і починаємо збирати схему рис. 7.12. За допомогою блоків VCC і GND – задаємо попередньо встановлене значення в двійковому коді (9 в десятковому = 1001 в двійковому).



Рисунок 7.13 – Приклад підпису пінів

5. За допомогою кнопок В8/А7 подаємо імпульси на лічильник мікросхеми 74193. З її виходу знімається 4 – розрядний двійковий код, який подається на вхід мікросхеми 74154 де перетворюється на сигнал на одному з виходів. А далі подається на наш створений блок 7_16, з якого надходить на відповідні сегменти 7-сегментного індикатора.
Вихід з ВОН подається сигнал, який при перемиканні таймера з 0 на 16 подає високу напругу на інверсний вхід LDN, і таймер замість початку рахунку з числа 16, починає свій відлік з попередньо встановленого числа, тобто 9.
6. Далі виконуємо симулювання, розпіновку на платі і завантажуюмо даний проект на ALTERA, і вже там пробуємо те що вийшло.

Приклад 2. Виконати автоматизацію стенду по намотці витків для обмотки двигуна на основі мікросхеми 7493. Схема має 4 кнопки: «Пуск», «Педаль» (Пуск2), «Стоп», «Кінцевий вимикач» і 4 вихідних функції: «Мотор», «Механічне гальмо», «Обнулення лічильника», «Значення лічильника в даний момент». Після надходження сигналу «Пуск» знімається механічне гальмо і обнуляється лічильник. Після натиснення оператора ногою на кнопку «Педаль» вмикається двигун який через редуктор, починає намотувати на спеціальне рамку витки обмотки. На рамці розташований спеціальний бойок, який після кожного нового кола (витка обмотки) на короткий час натискає на кінцевий вимикач. Рамка повинна здійснити 7 обертів, після чого двигун має зупинитись. Після зупинки двигуна оператор натискає кнопку «Стоп» і система повертається в вихідне положення.

1. За завданням складаємо графоперехід на 2 тригера рис. 7.14:

Для полегшення роботи прийняті такі позначення до кнопок і вихідних змінних - Пуск – А; Педаль – В; Стоп –С; Кінцевий вимикач – К; Мотор – М;

Механічне гальмо – Mg ; Обнулення лічильника – D ; Кількість витків намотаних на рамку - N .

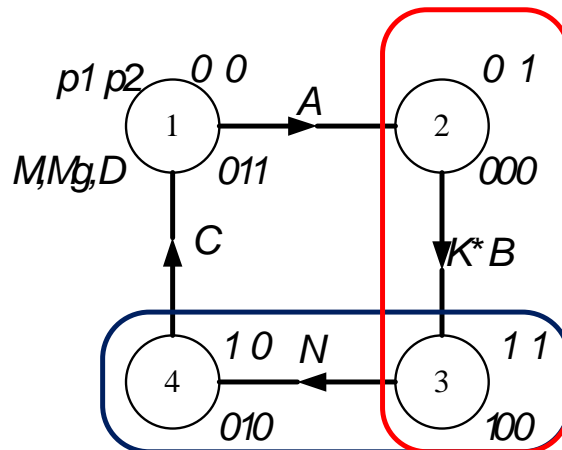


Рисунок 7.14 – Графоперехід

2. На основі графопереходу записуємо рівняння для встановлення і скидання тригерів, а також для вихідних змінних:

$$S_{p1} = (K * B) * p_2$$

$$S_{p2} = A * \bar{p}_1$$

$$R_{p1} = C * \bar{p}_2$$

$$R_{p2} = N * p_1$$

$$M = p_1 * p_2$$

$$Mg = \bar{p}_1 * \bar{p}_2 + p_1 * \bar{p}_2$$

$$D = \bar{p}_1 * \bar{p}_2$$

3. Складаємо схему на логічних елементах рис. 7.15.

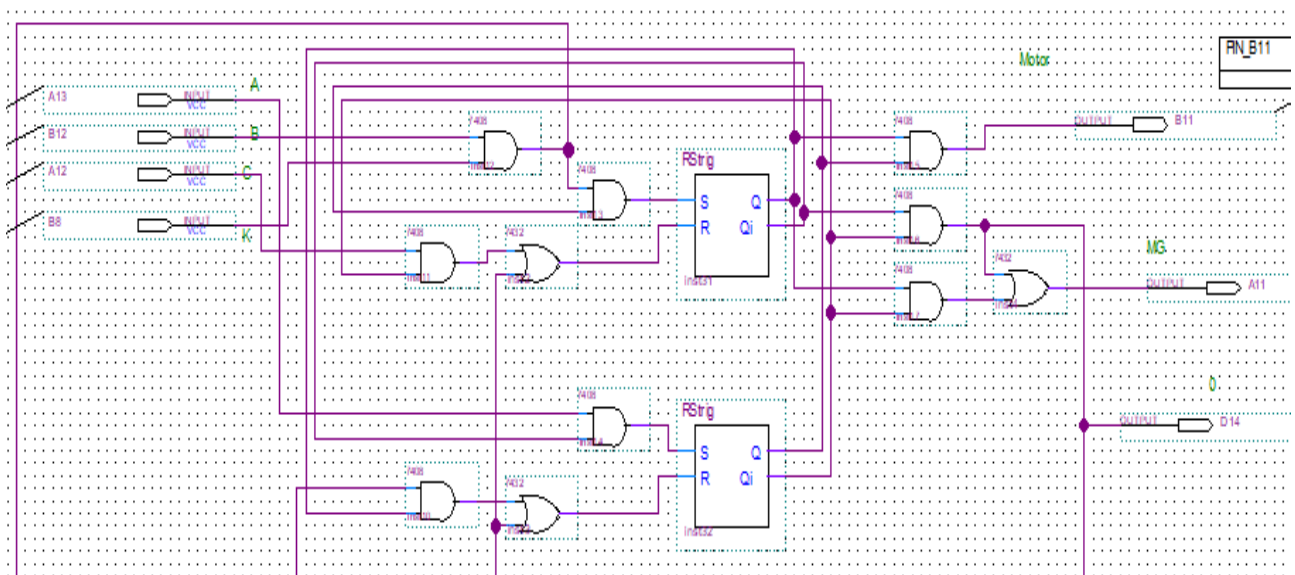


Рисунок 7.15 – Схема на основі рівнянь до прикладу 2

4. Тепер додаємо мікросхему лічильника 7493 і ще дві мікросхеми 74154 та 7_16 рис. 7.16. І підключаємо всі інші контакти.

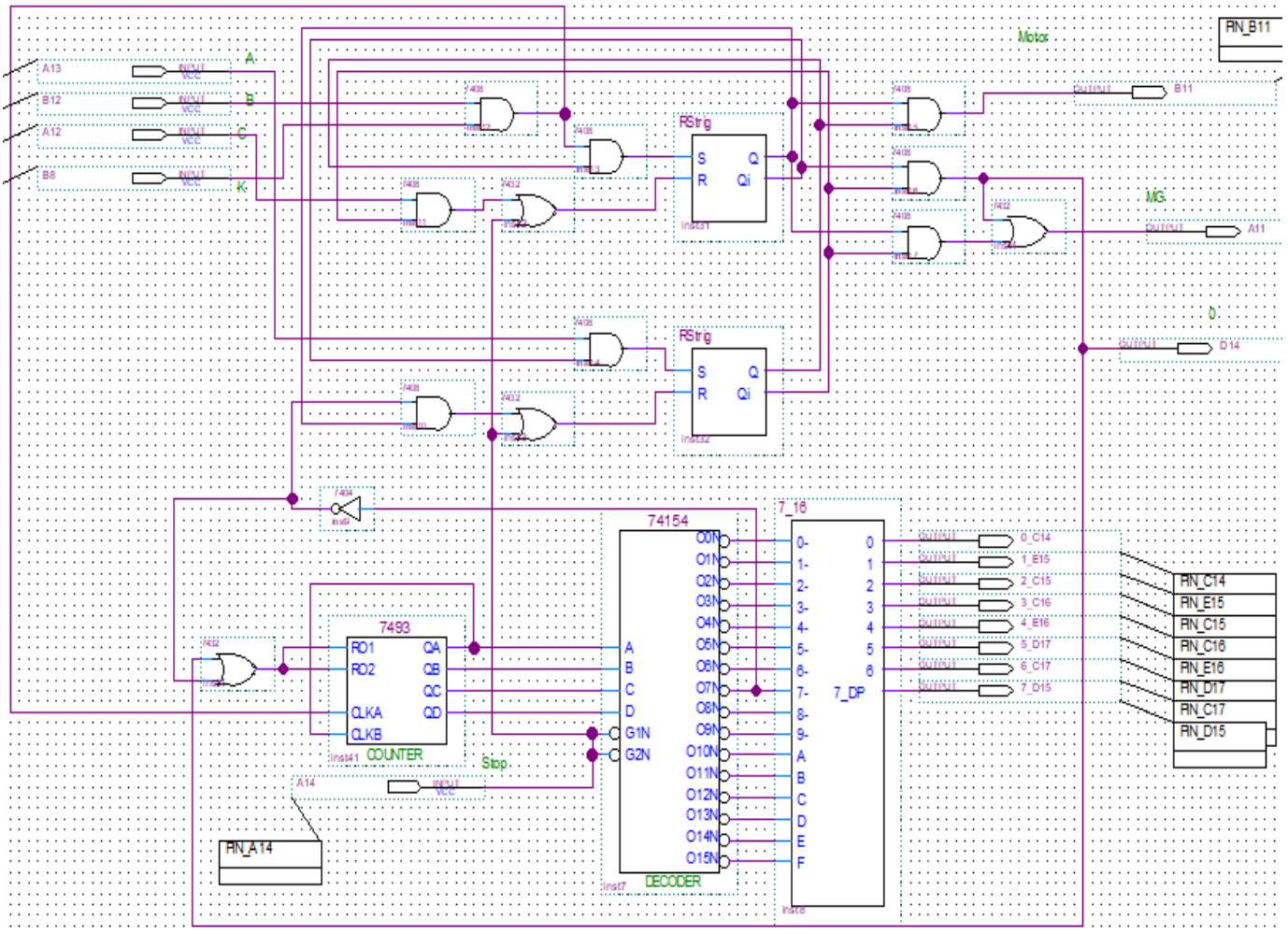


Рисунок 7.16 – Схема до прикладу 2

На рис. 7.16 можна побачити, що до входів скидання тригера під'єднано тумблер A14, - це потрібно для того, щоб після завантаження програми на нашу плату скинути RS – тригери. Оскільки після завантаження їх стан може бути не визначений, і синтез працюватиме некоректно.

На входи RO1-RO2 лічильника 7493 через мікросхему 7432 подається два сигнали, оскільки один скитує мікросхему перед початком роботи, а інший після досягнення необхідного значення (в даному випадку 7).

5. Дані виконуємо симулювання, розпіновку на платі і завантажуюмо даний проект на ПЛІС.
6. За допомогою кнопок перевіряємо чи коректно працює граф перехід, і перевіряємо на виконання умов синтезу.

Приклад 3. Впровадити схему з безперервним рухом цифри 9 в обидві сторони на семи-сегментних індикаторах. Початковий вихідний стан індикатора 0. При однократному натисканні на кнопку з'являється цифра 9, замість 0. Рух цифри продовжується в заданому напрямку через кожен наступний індикатор. При цьому там де була цифра – повинна знову з'явитись початкове значення. За основу схеми взяти мікросхему 74193.

1. Для початку розташовуємо мікросхему самого лічильника 74193, та підключаємо до неї кнопки +1/-1 та скидання лічильника. Далі до неї підключаємо мікросхему 74154 демультиплексора 4-16 рис. 7.17.

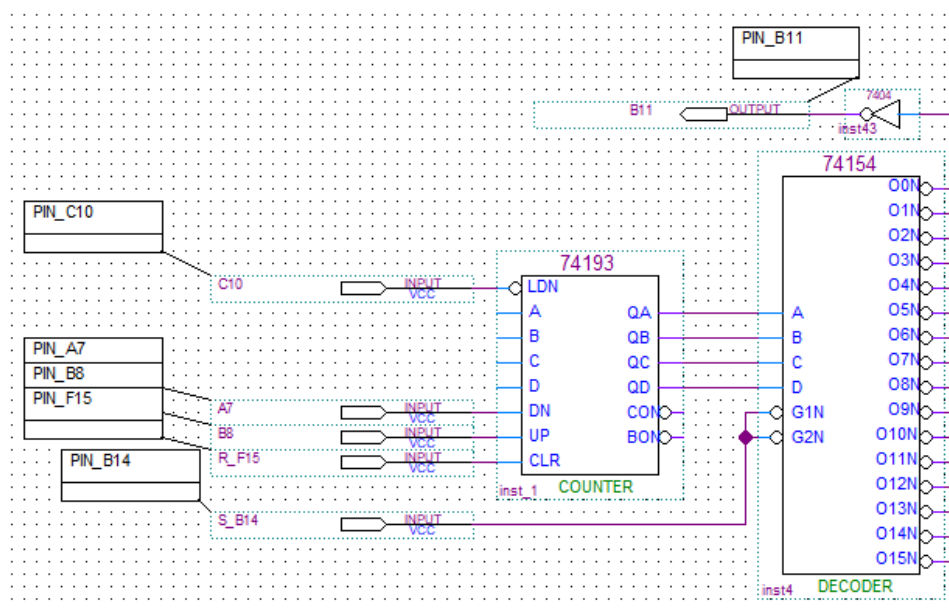


Рисунок 7.17 – Схема підключення лічильника

2. Далі встановлюємо 6 попередньо створених блоків «7-16» для перетворення сигналу з 74154 в код для семи-сегментного індикатора. За допомогою елементів І та АБО-НІ робимо розводку між блоками «7-16» рис. 7.18.

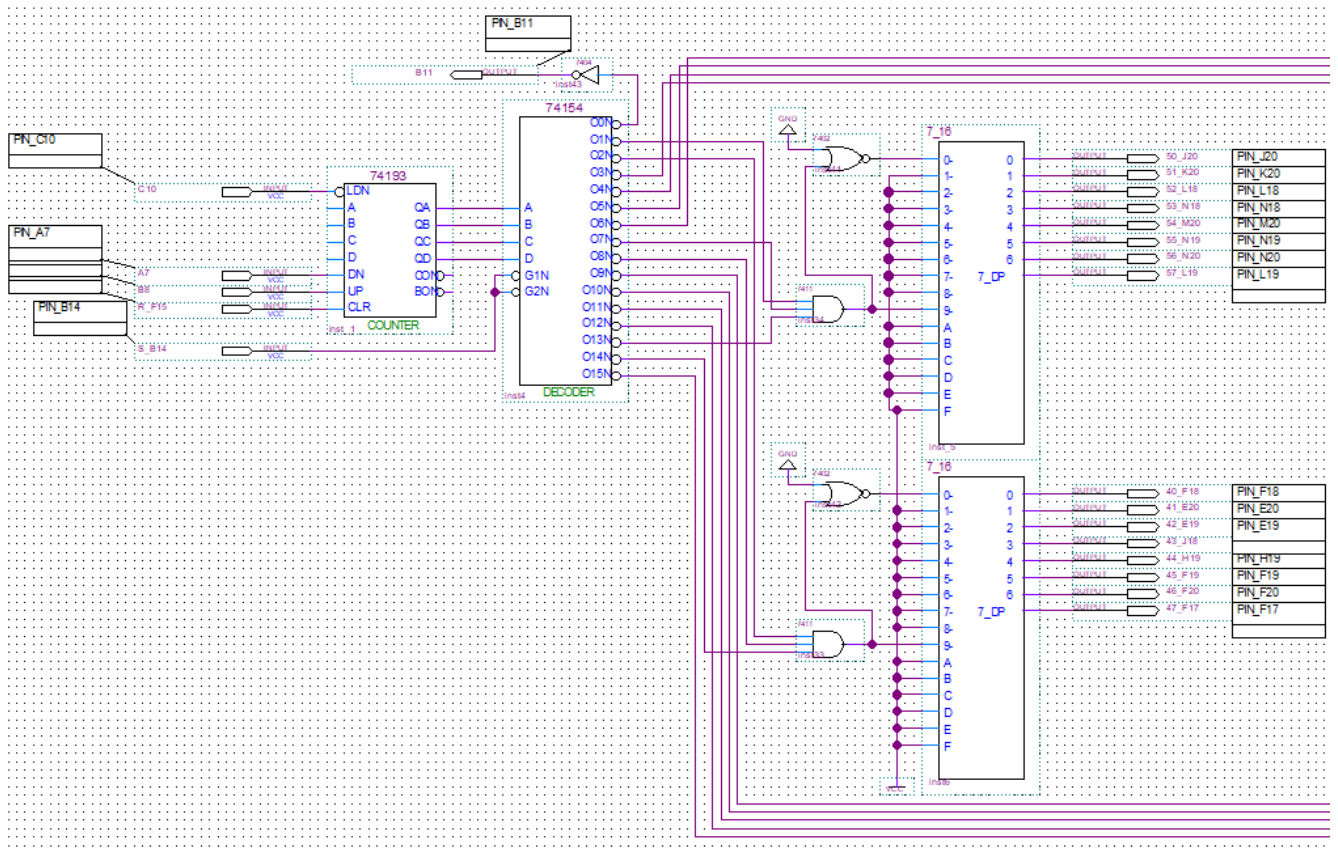


Рисунок 7.18 – Розводка схеми до прикладу 3

3. В даному випадку ми будемо використовувати цифри 0 та 9, тому на всі інші цифри ми подаємо логічну одиницю за допомогою блока VCC, щоб вони не засвітились рис. 7.19-7.20.
4. Оскільки за умовами завдання в початковому вихідному стані індикатори повинні показувати 0, то на нульову ніжку «7-16» через елемент 7402 подаємо логічний нуль, тобто сигнал блока GND.
5. На другу ніжку мікросхеми 7402 ми подаємо вихідний сигнал який запалює цифру 9 на відповідному 7 – сегментному індикаторі, а на саму мікросхему ми подаємо ту черговість сигналів яка нам потрібна.
6. Дані виконуємо симулювання, розпіновку на платі і завантажуюмо даний проект на ALTERA.

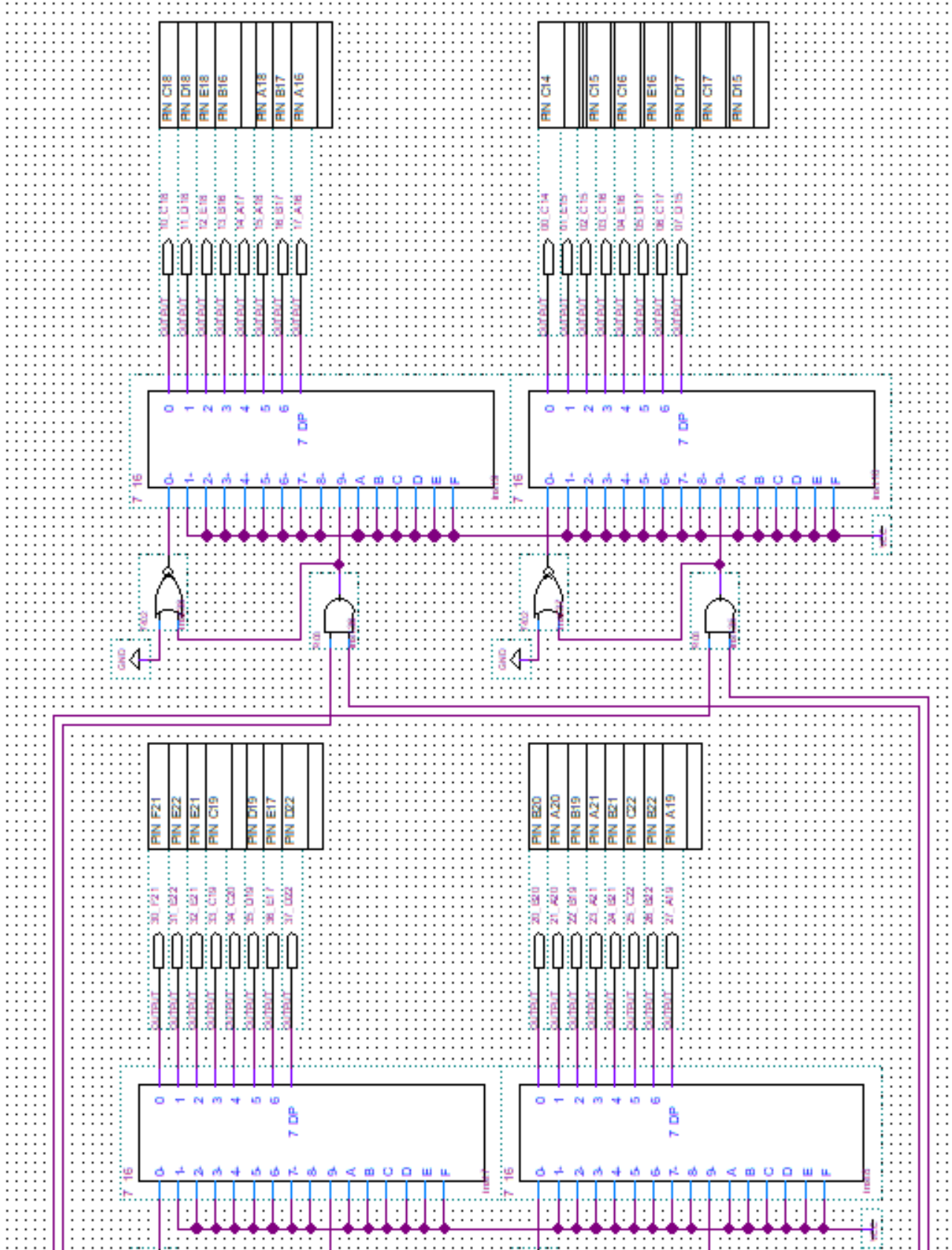


Рисунок 7.19 – Схема до прикладу 3

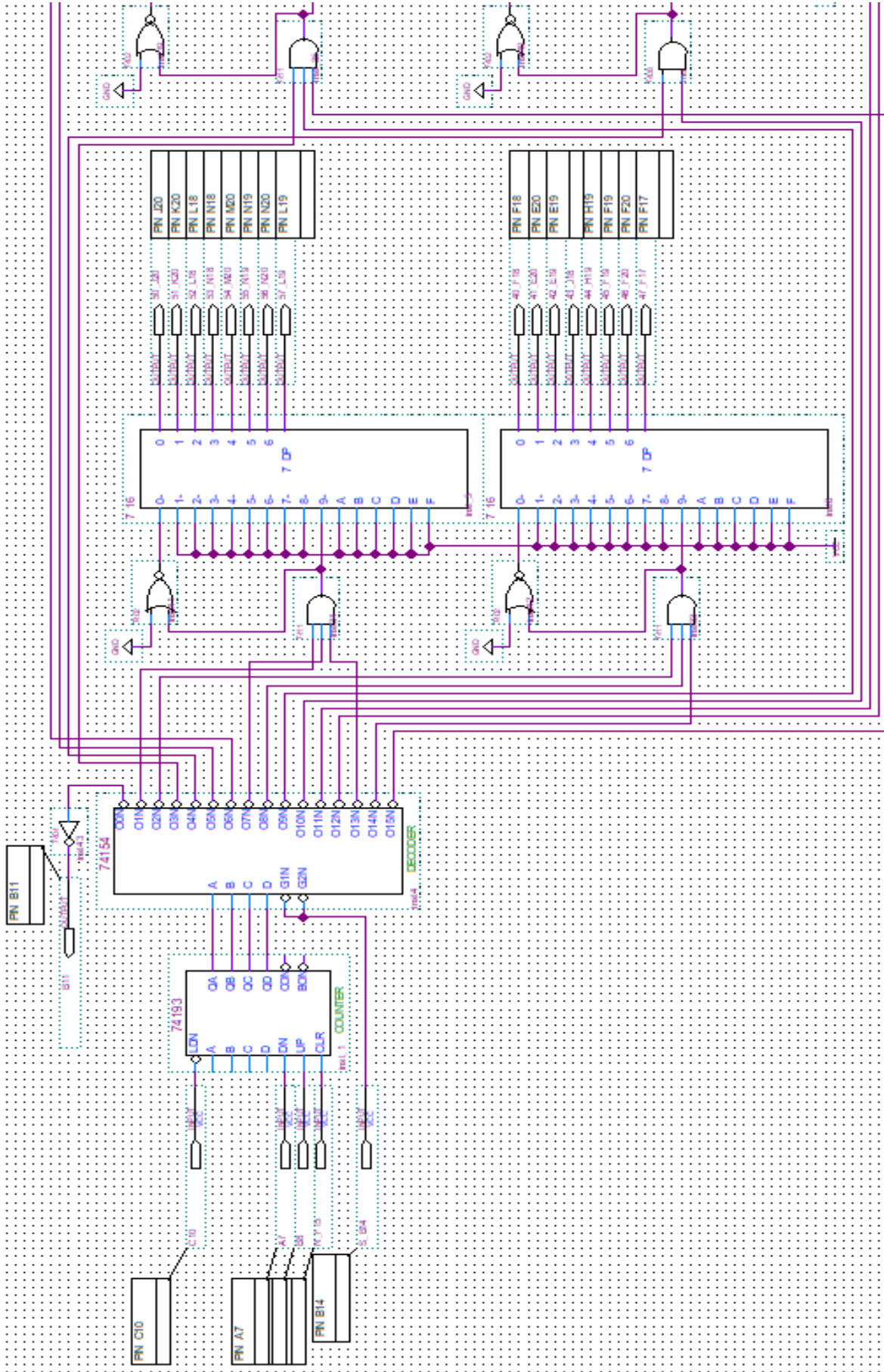


Рисунок 7.20 – Схема до прикладу 3

7.5 Зміст звіту

- 1) Титульний аркуш.
- 2) Мета роботи, програма роботи.
- 3) Умова роботи схем згідно варіанту.
- 4) Логічний синтез схеми керування.
- 5) Програма на мові FBD
- 6) Таблиці для вхідних/вихідних сигналів.
- 7) Висновок про відповідність роботи заданим умовам.

7.6 Контрольні питання

1. Перелічіть основні групи мікросхеми 74 серії.
2. Поясніть принцип використання лічильників 7493 та 74193 та при розв'язанні яких задач вони можуть використовуватися.
3. Поясніть принцип роботи мікросхеми 74154. В яких режимах вона може працювати?
4. Як схематично можна збільшити розрядність лічильника та дешифратора 74 серії?
5. Які елементарні логічні операції реалізовано на мікросхемах 74 серії. Наведіть приклади.
6. Яким чином створюється новий FBD блок у Quartus Prime?

Лабораторна робота № 8

ЛОГІЧНИЙ СИНТЕЗ ТА ДОСЛІДЖЕННЯ СХЕМ АВТОМАТИЗАЦІЇ ТИПОВИХ ПРОМИСЛОВИХ МЕХАНІЗМІВ НА ОСНОВІ ПРОГРАМОВАНИХ ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМ (ПЛІС)

Тривалість лабораторної роботи – 4 год.

Тривалість домашньої роботи – 4 год.

Мета роботи – поглибити практичні вміння синтезу одноктактних та багатотактних схем, навчитися будувати програми на мові FBD за отриманими логічними виразами у середовищі Quartus II, програмувати ПЛІС сімейства Cyclone II та практично перевіряти працездатність розроблених схем для систем автоматизації.

8.1 Основні теоретичні відомості

8.1.1 Мінімізація логічних функцій за допомогою карт Карно

Карта Карно – один з графічних способів подання логічних функцій. Для функцій n змінних вона складається з 2^n клітинок, причому кожна клітинка відповідає певному набору змінних. Вигляд карт Карно для функцій 2,3 і 4-ти змінних зображений на рис. 8.1. Вхідні змінні розміщуються з зовнішніх сторін карти проти її рядків або стовпців.

Значення вхідної змінної стосується усіх клітинок у рядку або стовпці і дорівнює 1, якщо проти рядка або стовпця є дужка з позначенням цієї змінної. Для решти рядків і стовпців значення змінної дорівнює 0. У клітинках карти записується те значення функції, яке вона має при наборах вхідних змінних, що відповідають цим клітинкам.

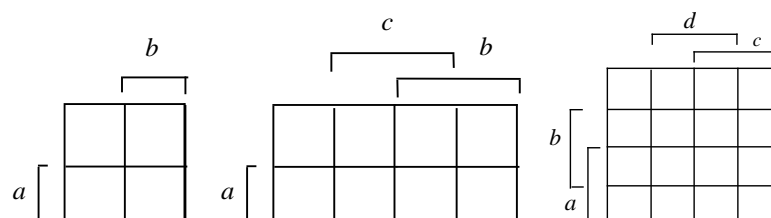


Рисунок 8.1 – Форма запису карт Карно

Функцію, задану алгебраїчним виразом, можна подати у вигляді карти Карно. Для цього передусім задану функцію треба подати в диз'юнктивній (ДНФ) або кон'юнктивній (КНФ) нормальній формі. При використанні карт Карно для мінімізації логічних функцій необхідно побудувати карту для відповідної кількості змінних і нанести на неї задану функцію. Потім слід об'єднати сусідні клітинки з одиницями в контури, записати вирази для контурів і скласти їх диз'юнкцію.

8.1.2 Синтез багатотактних схем методом таблиць переходів та карт Карно

Схема називається багатотактною, якщо стан її виходів залежить не тільки від набору вхідних сигналів у даний момент часу, але й від послідовності їх надходження.

Синтез схеми методом таблиць переходів і карт Карно виконується в такій послідовності:

- 1) складання первинної таблиці переходів;
- 2) стиснення первинної таблиці переходів;
- 3) визначення кількості і розміщення станів проміжних змінних;
- 4) складання карт Карно для проміжних і вихідних змінних і визначення за ними алгебричних виразів.

Складання первинної таблиці переходів. Цей етап починається з визначення кількості вихідних станів схеми. Оскільки умови роботи схеми не передбачають жодних обмежень на послідовність надходження вхідних сигналів, тому слід розглянути усі можливі комбінації вхідних сигналів і для кожної з них визначити кількість можливих станів схеми. Після визначення кількості станів схеми будемо первинну таблицю переходів.

Стисненням первинної таблиці переходів. Ця операція виконується з метою скорочення кількості рядків таблиці шляхом сполучення кількох рядків в один.

Визначення кількості і розміщення станів проміжних змінних. Кількість проміжних змінних визначається виходячи з того, що кожному рядку стисненої

таблиці переходів повинна відповідати своя певна комбінація значень проміжних змінних.

Складання карт Карно для проміжних і вихідних змінних і визначення за ними алгебричних виразів. Карти Карно складають так, щоб вхідні змінні розміщувалися зверху карт так само, як і у середній частині таблиці переходів, а проміжні змінні – ліворуч.

8.1.3 Синтез схем керування на основі циклограм

Циклограма – це графічне зображення послідовності роботи окремих елементів схеми у часі.

Синтез виконують у такій послідовності:

1. Записують умову спрацьовування у вигляді формули f' , до якої входить сигнал, змінювання стану якого визначає початок вмикального такту. В умові спрацьовування він записується в тому стані, в якому перебував у вмикальному такті. Наприклад, для циклограми на рис. 8.2 умова спрацьовування.

$$f' = a$$

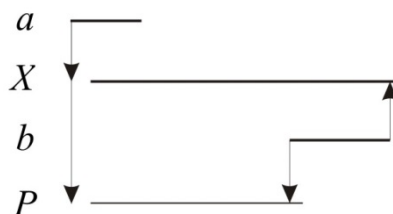


Рисунок 8.2 – Циклограма

2. Виконують першу перевірку реалізованості циклограми. При цьому перевіряють, чи зберігається значення $f' = 1$ протягом усього вмикального періоду.

Якщо ця умова виконується, то попередньо записана функція f' є остаточною і змінюванню не підлягає. Якщо ж функція f' змінює своє значення протягом вмикального періоду, звичайно застосовується самоблокування і умова спрацьовування записується у вигляді

$$f' = a + x$$

Для циклограми на рис. 8.2 необхідно ввести самоблокування, тому що сигнал a змінює свій стан протягом вмикального періоду.

3. Записують умову неспрацьовування у вигляді формули $\overline{f''}$. До цієї умови входить сигнал, що змінює свій стан на початку вимикального такту. Сигнал цього елемента записують у вигляді інверсії того стану, в якому він перебуває у вимикальному такті. Для циклограми на рис 8.2 умова неспрацьовування

$$\overline{f''} = \overline{b} = b.$$

4. Виконують другу перевірку реалізованості циклограми. Перевіряють, чи зберігається значення $\overline{f''} = 1$ протягом усього вмикального періоду. Якщо ця умова виконується, то записана функція $\overline{f''}$ є остаточною. Якщо ж функція $\overline{f''}$ змінює своє значення протягом вмикального періоду, то необхідно ввести проміжний елемент P , який змінює свій стан протягом періоду вмикання вихідного елемента, але після того, як змінить значення функція $\overline{f''}$, і залишається у цьому стані принаймні до кінця періоду вмикання вихідного елемента.

Умову неспрацьовування у разі введення проміжного елемента записують у вигляді інверсії кон'юнкції двох сигналів f'' і p , тобто $\overline{f''p}$. Стани сигналів f'' і p беруть такими, які вони є у вимикальному такті.

Для циклограми на рис.8.2 друга перевірка не задовольняється, тому введено проміжну змінну P , лінію дії якої показано на рисунку. Після введення проміжної змінної умова неспрацьовування матиме вигляд:

$$\overline{f''} = \overline{b\overline{p}} = b + p.$$

5. Записують попередню формулу вмикання елемента у вигляді кон'юнкції умов спрацьовування та неспрацьовування, отриманих після виконання першої та другої перевірок. Для циклограми на рис. 8.2

$$X = f' \overline{f''} = (a + x)(b + p).$$

6. Виконують третю перевірку реалізованості циклограми. Для цього функцію $f' \overline{f''}$ треба подати в диз'юнктивній нормальній формі (ДНФ),

розкривши дужки, і перевірити, чи не набуває значення 1 будь-яка елементарна кон'юнкція в одержаній ДНФ протягом вимикального періоду. Якщо таких елементарних кон'юнкцій немає, то третя перевірка задовольняється і логічну формулу для вихідного елемента змінювати не треба. Якщо ж така кон'юнкція є, то це спричинює помилкове спрацьовування вихідного елемента. У цьому разі треба ввести проміжний елемент P , який мав би різні значення для комбінації сигналів, що відповідає цій кон'юнкції, у вмикальному і вимикальному періодах.

Сигнал проміжного елемента додається у вигляді кон'юнкції до тієї елементарної кон'юнкції, яка дорівнює одиниці одночасно у вмикальному та вимикальному періодах. Стан цього елемента беруть таким, який він має у вмикальному періоді для комбінації сигналів, що відповідають розглядуваній кон'юнкції.

Для циклограми на рис.8.2

$$X = (a + x)(b + p) = ab + ap + xb + xp$$

і третя перевірка задовольняється, тому що жодна з комбінацій сигналів ($a = 1, b = 1; a = 1, p = 1; x = 1, b = 1; x = 1, p = 1$) не трапляється у вимикальному періоді елемента X . Тому формула $X = (a + x)(b + p)$ є остаточною.

7. Складають формули для всіх проміжних елементів, які були введені під час виконання перевірок реалізованості циклограми. Для них логічні формули складаються за такими ж самими правилами, як і для вихідних.

Для циклограми на рис. 8.2 введено один проміжний елемент P . Визначимо для нього логічну формулу.

Умова спрацьовування

$$f'_p = a,$$

умова неспрацьовування

$$\overline{f''_p} = \overline{b}.$$

Перша перевірка: значення $f'_p = 1$ не зберігається протягом усього вмикального періоду, тому застосовуємо самоблокування і записуємо умову спрацьовування у вигляді:

$$f'_p = a + p.$$

Друга і третя перевірки задовольняються. Остаточна логічна формула для проміжного елемента P

$$P = (a + p)\bar{b}$$

8.1.4 Синтез схем автоматики на тригерах

При синтезі схем на тригерах умови роботи багатотактних схем подаються у вигляді графів переходів.

Граф переходів – це графічне зображення послідовності роботи багатотактної схеми. Елементами графа є *вершини* і *ребра*. Вершини відповідають станам схеми і позначаються кружками. Ребра – це лінії із стрілками, що з'єднують вершини і показують напрям переходу з одного стану схеми в інший.

Кількість вершин графа при синтезі асинхронних схем на RS – тригерах визначається з умови $2^n \geq S$, де S – кількість станів схеми; 2^n – кількість вершин графа; n – кількість тригерів.

Вершини графа рекомендується розмішувати так, щоб при $n=2$ вони створювали конфігурацію 2×2 , при $n=3$ – конфігурацію 4×2 , а при $n=4$ – конфігурацію 4×4 . Вигляд графа переходів при $n=3$ показано на рис. 8.3.

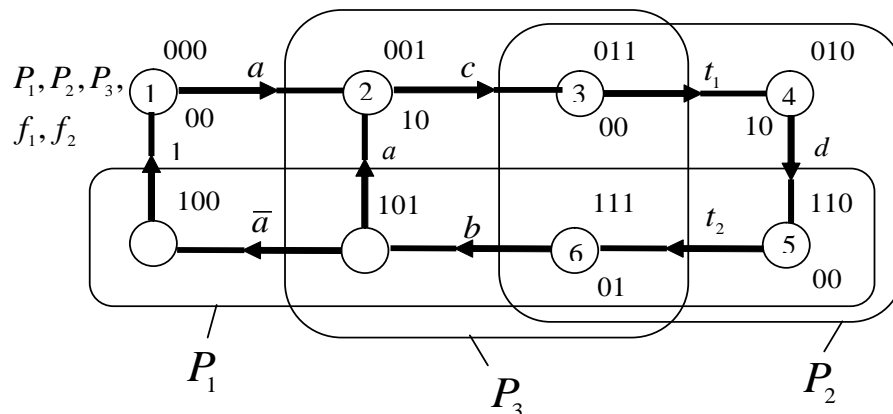


Рисунок 8.3 – Вигляд графа переходів при $n=3$

Вихідні сигнали тригерів виконують роль проміжних змінних, тому тригери та їх вихідні сигнали позначаються буквами P_1, P_2, \dots . Ці позначення записують зверху ліворуч від графа.

Кожна вершина графа кодується набором значень вихідних сигналів тригерів. Коди вибираються так, щоб для сусідніх вершин вони відрізнялися значенням тільки однієї змінної. У вихідному стані схеми (стан очікування) звичайно приймають, що усі проміжні змінні дорівнюють нулеві. Значення проміжних змінних для кожної вершини записуються над кружками у тій черговості, в якій записані позначення тригерів.

Вершини, між якими повинні відбуватися переходи, з'єднують ребрами із стрілками. Над стрілками або праворуч від них, якщо ребра спрямовано вертикально, записують позначення вхідних сигналів, що спричинюють ці переходи.

При побудові схем на асинхронних RS – тригерах переходи можна робити тільки між сусідніми вершинами. Якщо ця умова не виконується, то необхідно передбачити так звані природні переходи (за рахунок подавання вхідного сигналу одиниця) через проміжні нестійкі стани.

Як приклад побудуємо граф переходів, що описує роботу схеми керування механізмом M , який працює в режимі циклів, що повторюються. У вихідному положенні механізму натиснутий кінцевий вимикач $SQ1$. Цикл роботи починається після надходження команди "Пуск". Механізм переміщується в положення, що фіксується кінцевим вимикачем $SQ2$, стоїть протягом часу Δt_1 , переміщується в положення, що фіксується кінцевим вимикачем $SQ3$, стоїть протягом часу Δt_2 , а потім повертається в вихідне положення. Для повторення циклу необхідно знов подати команду "Пуск". Якщо ця команда надходить безперервно, то після відпрацювання одного циклу автоматично починається наступний.

Приймемо такі позначення вхідних і вихідних сигналів, а також сигналів таймерів, які необхідно розглядати як вхідні сигнали для графа переходів. Вхідні сигнали: a – команда "Пуск"; b, c, d – сигнали кінцевих вимикачів $SQ1, SQ2, SQ3$ відповідно; t_1, t_2 – сигнали таймерів, що дають затримки Δt_1 і Δt_2 . Вихідні сигнали: f_1 – команда на переміщення механізму з вихідного положення, f_2 – команда на повернення механізму в вихідне положення.

Побудову графа переходів починаємо з визначення кількості станів, в яких може перебувати схема автоматичного керування. Таких станів шість: 1 – вихідне положення; 2 – переміщення з вихідного положення; 3 – стоянка протягом часу Δt_1 ; 4 – подальше переміщення; 5 – стоянки протягом часу Δt_2 ; 6 – повернення в вихідне положення.

Виходячи з кількості станів схеми $6 < 2^3$, визначаємо кількість тригерів $n=3$ і кількість вершин графа переходів $2^3=8$. Позначаємо проміжні змінні P_1, P_2, P_3 , будуємо 8 вершин графа і кодуємо їх комбінаціями значень проміжних змінних (рис. 8.3).

Кожному стану схеми ставимо у відповідність одну з вершин графа, причому стани, між якими повинен відбуватися перехід згідно з умовами роботи схеми, розміщуємо у сусідніх вершинах. Вільні вершини використовуємо для переходу з стану 6 в стан 2 під час роботи схеми в режимі циклів, що повторюються, або в стан 1 при відпрацьовуванні поодиноких циклів.

Позначення вихідних сигналів f_1 і f_2 записано під позначеннями проміжних змінних. Значення f_1 і f_2 для кожного стану схеми записуємо під відповідними вершинами графа.

Синтез схеми полягає у записі умов вмикання і скидання кожного тригера. Для цього охоплюють замкнутою лінією всі стани на графі переходів, в яких значення вихідного сигналу даного тригера дорівнює одиниці. Вхідні сигнали схеми, позначення яких стоять на ребрах, що заходять в одержану замкнуту область, встановлюють тригер в стан 1, а вхідні сигнали на ребрах, що виходять з цієї області, скидають тригер в стан 0.

Умови вмикання тригера записуються у вигляді добутку сигналу на ребрі, що заходить в область, і сигналів решти тригерів, стан яких не змінюється при переході, позначеному ребром. Наприклад, якщо сигнал на ребрі, що заходить в область з одиничним значенням вихідного сигналу тригера P_1 , дорівнює a , а тригери P_2 і P_3 не перемикаються, а зберігають стан $P_2=1$, $P_3=0$, то умова вмикання тригера P_1 записується у вигляді

$$S_{P_1} = ap_2 \overline{p_3}.$$

Якщо в замкнуту область входить кілька ребер, то умова вмикання тригера записується у вигляді суми добутоків відповідних сигналів, складених для кожного ребра.

Умова скидання тригера записується аналогічно для кожного ребра, що виходить з даної області, і подається у вигляді формули R_{P_i} . Описану процедуру виконують для кожного тригера і визначають для них умови вмикання і скидання.

Застосувавши описану процедуру визначення умов вмикання і скидання тригерів, для графа переходів на рис. 8.3 отримаємо:

$$S_{P_1} = dp_2 \overline{p_3}; R_{P_1} = a\overline{p_2}p_3 + \overline{p_2}\overline{p_3};$$

$$S_{P_2} = c\overline{p_1}p_3; R_{P_2} = bp_1p_3;$$

$$S_{P_3} = a\overline{p_1}\overline{p_2} + t_2p_1p_2; R_{P_3} = t_1\overline{p_1}p_2 + \overline{a}p_1\overline{p_2}.$$

Формули для вихідних сигналів f_1 і f_2 записуються як комбінаційні функції вихідних сигналів тригерів P_1, P_2, P_3 . Дійсно, функція $f_1=1$ в станах 2 і 4, тобто в станах, яким відповідають такі комбінації значень вихідних сигналів тригерів: $p_1=0, p_2=0, p_3=1$ і $p_1=0, p_2=1, p_3=0$. Тому

$$f_1 = \overline{p_1}\overline{p_2}p_3 + \overline{p_1}p_2\overline{p_3}.$$

Аналогічно, функція $f_2=1$ в стані 6 ($p_1p_2p_3=111$), тобто

$$f_2 = p_1p_2p_3.$$

Таймер T_1 вмикається в стані 3 ($p_1p_2p_3=011$), а таймер T_2 – в стані 5 ($p_1p_2p_3=110$), тому

$$T_1 = \overline{p_1}p_2p_3;$$

$$T_2 = p_1p_2\overline{p_3}.$$

Синтез синхронних схем на JK -тригерах виконується аналогічно. Різниця полягає у тому, що для синхронних схем при побудові графа переходів є припустимими переходи між будь-якими вершинами графа, а не обов'язково сусідніми, тобто природні переходи не потрібні. Умови вмикання тригерів записують у вигляді формули J_{P_i} , умови скидання – у вигляді K_{P_i} .

8.2 Програма роботи

Перед початком лабораторної роботи група ділиться на бригади відповідно до кількості робочих місць. Кожному студенту бригади необхідно вибрати один із варіантів (1-6), який містить 4 завдання, з таблиці 8.1. Для кожного з чотирьох завдань необхідно:

1. Виконати логічний синтез схеми керування, та отримати логічні вирази, що описують схему (**виконується вдома під час СРС**).

2. За виразами у пункті 1 скласти схему на логічних елементах в пакеті програмного забезпечення Quartus II.

3. Зробити позначення виводів вхідних та вихідних змінних.

4. Провести часову симуляцію роботи схеми.

5. Записати програму до ПЛІС.

6. Підключити кнопки, світлодіоди, мікродвигуни і т.п. до входів/виходів ПЛІС відповідно до заданої кількості вхідних/вихідних сигналів, зробити «підтяжку» на GND вхідних сигналів.

7. Перевірити відпрацювання заданих умов роботи на стенді.

8. Показати зібрану схему викладачу та продемонструвати її роботу (**викладач підтверджує своїм підписом у протоколі працездатність схеми**);

9. Оформити звіт по лабораторній роботі та зробити висновки.

Таблиця 8.1 – Розподіл номерів завдань по варіантах

Бригада №	№ студента в бригаді	Номери завдань
1	1.1	1.1, 2.1, 3.1, 4.1
	1.2	1.2, 2.24, 3.24, 4.2
	1.3	1.3, 2.3, 3.3, 4.3
	1.4	1.4, 2.4, 3.4, 4.4
	1.5	1.5, 2.6, 3.7, 4.8
	1.6	1.6, 2.23, 3.9, 4.10
2	2.1	1.21, 2.5, 3.5, 4.5
	2.2	1.6, 2.6, 3.23, 4.6
	2.3	1.22, 2.7, 3.22, 4.7
	2.4	1.8, 2.8, 3.8, 4.8,
	2.5	1.9, 2.10, 3.11, 4.12
	2.6	1.24, 2.11, 3.12, 4.13
3	3.1	1.9, 2.9, 3.9, 4.9
	3.2	1.10, 2.10, 3.10, 4.10
	3.3.	1.11, 2.11, 3.11, 4.20
	3.4	1.12, 2.12, 3.12, 4.17,
	3.5	1.13, 2.2, 3.4, 4.5
	3.6	1.14, 2.3, 3.6, 4.7
4	4.1	1.13, 2.13, 3.13, 4.19
	4.2	1.14, 2.14, 3.14, 4.14
	4.3	1.15, 2.15, 3.15, 4.15
	4.4	1.23, 2.16, 3.16, 4.16
	4.5	1.10, 2.10, 3.11, 4.11
	4.6	1.5, 2.6, 3.7, 4.18
5	5.1	1.17, 2.17, 3.5, 4.8
	5.2	1.5, 2.18, 3.2, 4.19
	5.3	1.8, 2.2, 3.17, 4.17
	5.4	1.4, 2.3, 3.18, 4.22
	5.5	1.18, 2.19, 3.7, 4.8
	5.6	1.6, 2.20, 3.9, 4.23
6	6.1	1.7, 2.21, 3.19, 4.21
	6.2	1.19, 2.22, 3.6, 4.20
	6.3	1.7, 2.7, 3.20, 4.7
	6.4	1.20, 2.8, 3.21, 4.24,
	6.5	1.9, 2.10, 3.11, 4.12
	6.6	1.10, 2.11, 3.12, 4.13

8.3 Завдання до лабораторної роботи

1. Умови роботи схем задано наступними логічними функціями. Скласти схему на логічних елементах та перевірити її працездатність.

$$1.1 \quad f_1 = ab\bar{c}d + abd + \bar{a}c\bar{e} + \bar{a}bc$$

$$f_2 = (a + \bar{b} + \bar{c})(a + \bar{c} + d)(\bar{c} + d + e)$$

$$f_3 = (\bar{a} + b + c)(\bar{c} + d)\bar{b}cd$$

$$1.2 \quad f_1 = \bar{a}bcd + abc + a\bar{c}e + \bar{a}b\bar{d}$$

$$f_2 = (a + \bar{b} + c)(\bar{a} + c + d)(\bar{c} + d + e)$$

$$f_3 = (\bar{a} + b + d)(a + \bar{c} + d)bc$$

$$1.3 \quad f_1 = \bar{a}bcd + \bar{a}bc + \bar{a}ce + ab$$

$$f_2 = (a + \bar{b} + c)(a + \bar{e} + b)(\bar{c} + \bar{b} + \bar{e})$$

$$f_3 = (\bar{a} + b + d)(c + d)\bar{b}cd$$

$$1.4 \quad f_1 = \bar{a}bce + \bar{a}bd + \bar{a}cb + ab\bar{e}$$

$$f_2 = (a + b + c)(a + d + e)(b + c + \bar{d})$$

$$f_3 = (\bar{a}b + \bar{c}d + be)(\bar{a}bcd\bar{e} + de)$$

$$1.5 \quad f_1 = \bar{a}bcd + \bar{a}bc + b\bar{c}e + \bar{a}bc$$

$$f_2 = (a + \bar{b} + \bar{c})(a + \bar{c} + e)(\bar{c} + d + e)$$

$$f_3 = (a + b + \bar{d})(c + e)\bar{a}bcd$$

$$1.6 \quad f_1 = \bar{a}bcd + \bar{a}bc + b\bar{c}e + abc$$

$$f_2 = (\bar{a} + b + \bar{c})(a + c + \bar{e})(c + \bar{d} + e)$$

$$f_3 = (a + b + \bar{d})(c + e)\bar{a}bcd$$

$$1.7 \quad f_1 = ab\bar{c}d + ab\bar{c} + bc\bar{e} + abc$$

$$f_2 = (\bar{a} + \bar{b} + c)(\bar{a} + c + \bar{e})(c + d + \bar{e})$$

$$f_3 = (a + \bar{b} + d)(c + e)\bar{a}bcd$$

$$1.8 \quad f_1 = abc\bar{d} + \bar{a}b\bar{c} + \bar{b}c\bar{e} + \bar{a}b\bar{c}$$

$$f_2 = (\bar{a} + \bar{b} + \bar{c})(\bar{a} + c + e)(\bar{c} + d + \bar{e})$$

$$f_3 = (\bar{a} + b + e)(c + \bar{e})cd$$

$$1.9 \quad f_1 = \bar{a}bcd\bar{d} + \bar{a}b\bar{c} + \bar{b}c\bar{e} + \bar{a}b\bar{c}$$

$$f_2 = (\bar{a} + \bar{b} + \bar{c})(\bar{a} + \bar{c} + \bar{e})(c + d + \bar{e})$$

$$f_3 = (a + c + d + \bar{e})(c + \bar{e})\bar{b}d$$

$$1.10 f_1 = \overline{(a+b+c)} + \overline{(b+d+e)} + \overline{abd} + \overline{bcd}$$

$$f_2 = (a+b+c)(c+d+e)(b+e)$$

$$f_3 = (a+b+\overline{c})(\overline{a}+b)de$$

$$1.11 f_1 = \overline{abcd} + \overline{bce} + \overline{abc} + \overline{bce}$$

$$f_2 = (\overline{a}+b+e)(c+d+\overline{e})(b+\overline{d}+e)$$

$$f_3 = \overline{ad}(\overline{a}+b+c)(b+e)$$

$$1.12 f_1 = \overline{(a+\overline{b}+c)} + \overline{(b+\overline{d}+e)} + \overline{abcde}$$

$$f_2 = abcd + bc\overline{e} + ab\overline{c} + \overline{bce}$$

$$f_3 = \overline{abc}(c+d+\overline{e}) + \overline{abd} + \overline{acd}$$

$$1.13 f_1 = \overline{(a+b+\overline{c})} + \overline{(b+d+\overline{e})} + \overline{acde}$$

$$f_2 = (a+b)(\overline{c}+d)(b+e)(\overline{d}+a)$$

$$f_3 = \overline{abc}(c+d+\overline{e}) + \overline{abd} + \overline{acd}$$

$$1.14 f_1 = \overline{(\overline{a}+\overline{b}+c)} + \overline{(\overline{b}+\overline{d}+e)} + bde$$

$$f_2 = (a+b)(c+\overline{d})(b+e)(d+\overline{a})$$

$$f_3 = \overline{abc}(c+d+\overline{e}) + \overline{abd} + \overline{acd}$$

$$1.15 f_1 = \overline{abc} + \overline{bd\overline{e}} + \overline{abcde}$$

$$f_2 = (\overline{a}+b)(\overline{c}+d)(\overline{b}+e)(\overline{d}+a)$$

$$f_3 = \overline{abcde} + \overline{abc} + \overline{cde} + \overline{cbd}$$

$$1.16 f_1 = \overline{abc} + bde + \overline{abcd\overline{e}}$$

$$f_2 = (a+\overline{b})(c+d)(b+\overline{e})(d+\overline{a})$$

$$f_3 = \overline{ade} + \overline{abc} + \overline{acde} + \overline{ade}$$

$$1.17 f_1 = \overline{(a+b+\overline{c})} + \overline{(b+d+\overline{e})} + \overline{abde}$$

$$f_2 = (\overline{a}+b)(c+e)(\overline{b}+e)(d+a)$$

$$f_3 = \overline{abd}(a+b+\overline{c})(b+e)$$

$$1.18 f_1 = \overline{(\overline{a}+b+c)}(b+d+e) + \overline{abcd\overline{e}}$$

$$f_2 = \overline{ab} + bcd + \overline{bde}$$

$$f_3 = \overline{ad} + db + (a+\overline{c})(b+\overline{e})$$

$$\begin{aligned}
1.19 \quad f_1 &= \overline{(a + b + c)} + bde + ac\bar{e} \\
f_2 &= (a + \bar{b} + c)(c + d + \bar{e})(b + \bar{e}) \\
f_3 &= de(a + b + \bar{c})(a + b) \\
1.20 \quad f_1 &= \bar{a}bcd + \bar{b}\bar{c}e + \overline{(a + b + c)} \\
f_2 &= (a + \bar{b})(c + \bar{d})(b + \bar{e})(d + \bar{a}) \\
f_3 &= \bar{a}d(\bar{a} + b + c)(b + \bar{e}) \\
1.21 \quad f_1 &= \bar{a}bcd + \overline{(a + \bar{b} + c)} + \bar{a}ce + ab \\
f_2 &= (a + \bar{b} + \bar{c})(a + \bar{c} + e)(\bar{c} + d + e) \\
f_3 &= (a + c + d + \bar{e})(c + \bar{e})\bar{b}d \\
1.22 \quad f_1 &= \overline{(a + b + c)} + bde + a\bar{c}de \\
f_2 &= (a + b)(c + d)(b + e)(\bar{d} + a) \\
f_3 &= \bar{a}bc(c + d + \bar{e}) + ab\bar{d}e + \bar{a}cd \\
1.23 \quad f_1 &= \overline{(a + \bar{b} + c)} + \overline{(b + \bar{d} + \bar{e})} + bde \\
f_2 &= (a + b)(c + \bar{d})(b + e)(d + \bar{e}) \\
f_3 &= \bar{a}bc(c + d + \bar{e}) + abd + \bar{a}c\bar{d} \\
1.24 \quad f_1 &= \bar{a}bcd + \bar{a}bc + \bar{b}\bar{c}e + \bar{a}bc \\
f_2 &= (\bar{a} + \bar{b} + c)(\bar{a} + c + \bar{e}) + cd\bar{e} \\
f_3 &= (a + c + \bar{e})(c + \bar{e})\bar{b}d
\end{aligned}$$

2. Виконати логічний синтез багатотактної схеми та перевірити її працездатність (в задачах з двигунами в якості вихідних змінних використовувати мікродвигуни M1 та M2).

2.1 Штаба під час руху перекриває фотодатчики a і b (рис. 8.4, а), при цьому сигнали датчиків набувають значення одиниця ($a=1, b=1$). Схема повинна виробляти сигнали, що визначають напрям руху штаби (ліворуч або праворуч). Якщо жоден з датчиків не перекритий, то вихідні сигнали схеми дорівнюють нулю.

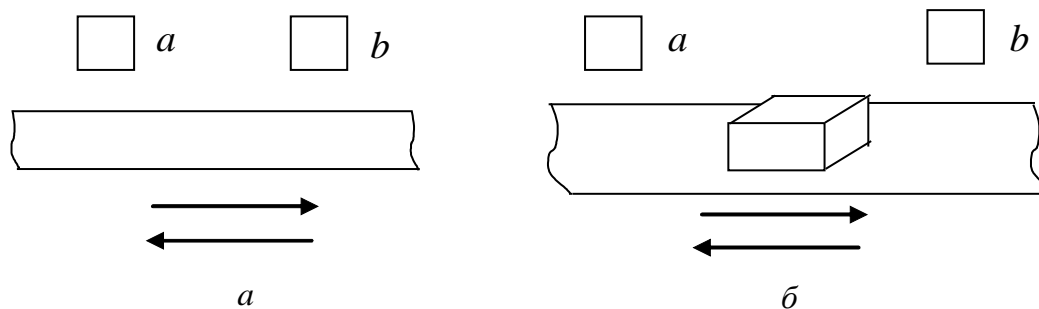


Рисунок 8.4 – Розміщення датчиків

2.2 Розташовані перед реверсивним транспортером датчики a і b (рис. 8.4, б) призначені для керування системою сигналізації про напрям руху виробу, що проходить повз них. Сигнал f_1 на виході системи дорівнює одиниці, якщо останній вибір рухався ліворуч, а сигнал $f_2 = 1$, якщо праворуч. У вихідному стані схеми обидва вихідні сигнали дорівнюють нулю.

2.3 Схема керування двигунами $M1$, $M2$ працює таким чином. При натисненні першої кнопки вмикається двигун $M1$, при натисненні другої – вмикається двигун $M2$ і вимикається двигун $M1$, при натисненні третьої – вимикається $M2$. Кнопки з самоповерненням і натискаються по черзі. Схема не повинна реагувати на порушення черговості натискання кнопок.

2.4 Схема виробляє сигнали f_1 і f_2 , які керують насосами $M1$ і $M2$. Насоси відкачують рідину з накопичуючого басейну. У басейні розміщено три датчики рівня: d_1 - нижній; d_2 – середній; d_3 – верхній. При досягненні рідиною відповідного рівня вихідний сигнал датчика набуває значення одиниця. У вихідному стані рівень рідини нижчий за рівень, що фіксується датчиком d_1 . При підніманні рідини до середнього рівня ($d_2 = 1$) вмикається насос $M1$. Якщо після цього рівень рідини опускається нижче рівня датчика d_1 , то насос $M1$ вимикається, якщо ж рівень рідини продовжує підвищуватися і спрацьовує датчик d_3 , то крім насоса $M1$ вмикається насос $M2$. Якщо після цього рівень рідини знизиться нижче середнього рівня ($d_2 = 0$), то вимикається насос $M2$, а насос $M1$ продовжує працювати. При подальшому зниженні рівня ($d_1 = 0$) насос $M1$ вимикається.

2.5 Схема керування двигуном $M1$ забезпечує дистанційний та місцевий режими роботи. Для вмикання двигуна в дистанційному режимі повинен бути поданим сигнал a (дозвіл дистанційного керування) і натиснута кнопка “Пуск”

(сигнал b). У режимі місцевого керування повинен бути поданим сигнал c (дозвіл місцевого керування) і натиснута кнопка d (пуск з місця). Робота схеми у будь-якому режимі неможлива при наявності сигналу e (заборона роботи). В режимі місцевого керування зупинка здійснюється кнопкою “Стоп1” (сигнал m), при дистанційному керуванні – кнопкою “Стоп 2” (сигнал n).

2.6 Автомат керується кнопками a і b і має чотири вихідних сигнали X_1, X_2, X_3, X_4 . Схема працює так. Спочатку натискають одну з кнопок, а потім, не відпускаючи її – другу. Коли одну з кнопок буде відпущено, сигнал на одному з виходів залежно від черговості натиснення та відпускання кнопок згідно з таблицею 8.2, повинен набувати значення одиниця. Після відпускання другої кнопки схема набуває вихідний стан (усі вихідні сигнали дорівнюють нулеві).

Таблиця 8.2 – Черговість натискання кнопок

Першою натиснута кнопка	Першою відпущена кнопка	Сигнал 1 на виході
a	a	X_1
a	b	X_2
b	b	X_3
b	a	X_4

2.7 Схема забезпечує керування чотирма лампами за допомогою однієї кнопки. У вихідному стані світяться перша і друга лампи. Якщо натиснути кнопку перший раз, то будуть світитися друга і третя лампи, а перша погасне. Після відпускання кнопки стан схеми не змінюється. Якщо натиснути кнопку другий раз, то будуть світитися третя і четверта лампи, а друга погасне. Якщо натиснути кнопку третій раз, то будуть світитися четверта і перша лампи, а третя погасне. Якщо натиснути кнопку четвертий раз, то схема набуває вихідний стан. Після цього цикл роботи схеми повторюється.

2.8 Схема забезпечує місцевий та дистанційний пуск вперед і назад та зупинку двигуном. При місцевому керуванні необхідно подати сигнал $d=1$ – дозвіл місцевого керування. Керування електроприводом в цьому режимі здійснюється кнопками “Вперед 1”, “Назад 1”, “Стоп 1”. Для керування електроприводом в дистанційному режимі необхідно подати сигнал $e=1$ (дозвіл дистанційного керування). Пуск і зупинка електропривода здійснюється кнопками “Вперед 2”, “Назад 2”, “Стоп 2”. При розробці схеми передбачити

індикацію стану електропривода при дистанційному керуванні, а також блокування, яке унеможливорює одночасно вмикання контакторів “Вперед” і “Назад”.

2.9 Схема забезпечує автоматичне і ручне керування двигуном М1. Вибір режиму здійснюється перемикачем, який має три положення: ручне, нейтральне, автоматичне. При ручному керуванні пуск і зупинка двигуна здійснюється кнопками “Пуск” і “Стоп”, в автоматичному режимі – відповідними командами від пристрою автоматичного керування. При розробці схеми передбачити необхідність повернення перемикача режимів в нейтральне положення для повторного вмикання двигуна після раптового зникнення і наступного поновлення напруги живлення.

2.10 Умови роботи схеми збігаються із завданням 2.9. Різниця полягає у тому, що привод реверсивний і при ручному керуванні команди подаються кнопками “Вперед”, “Назад”, “Стоп”, а при автоматичному – відповідними сигналами від пристрою автоматичного керування.

2.11 Схема має три вхідні сигнали a, b, c і один вихідний X . Сигнал $X=1$, якщо усі три вхідні сигнали дорівнюють одиниці, причому сигнал a набув значення 1 першим. Після того, як сигнал X набув значення 1, воно зберігається доти, поки $a=1$, незалежно від значень сигналів b і c .

2.12 Схема має три вхідні сигнали a, b, c і один вихідний X . Сигнал $X=1$, якщо усі три вхідні сигнали мають значення 1, причому сигнал a набуває значення 1 останнім. Після того, як сигнал X набув значення 1, воно зберігається доти, поки $a=1$, незалежно від значень сигналів b і c .

2.13 Схема має три вхідні сигнали a, b, c і один вихідний X . Сигнал $X=1$, якщо усі три вхідні сигнали мають значення 1, причому сигнал a набуває значення 1 другим. Після того, як сигнал X набув значення 1, воно зберігається доти, поки $a=1$, незалежно від значень сигналів b і c .

2.14 Виконати синтез схеми, що має два вхідні сигнали a і b і два вихідні f_1 і f_2 . Стан входів розглядається як двійкове число. Якщо при зміні входів це число збільшується, то $f_1=1, f_2=0$, якщо зменшується, то $f_1=0, f_2=1$.

2.15 Схема має два вхідні сигнали a і b та один вихідний X . Сигнал $X=1$ тільки в тому разі, коли a спочатку набуває значення 1, а потім змінюється з одиниці на нуль. Після цього значення $X=1$ зберігається доти, поки сигнал b спочатку набуває значення одиниця, а потім зміниться з одиниці на нуль.

2.16 Схема порівнює за величиною два двійкові дворозрядні числа $A = a_2a_1$ і $B = b_2b_1$. Вихідні сигнали схеми X, Y, Z . Якщо $A > B$, то $X=1, Y=0, Z=0$; якщо $A < B$ то $X=0, Y=1, Z=0$; якщо $A=B$, то $X=0, Y=0, Z=1$.

2.17 Схема має три вхідні сигнали a, b, c і один вихідний X . Сигнал $X=1$, якщо усі три вхідні сигнали мають значення 1, причому сигнал a набуває значення 1 першим, сигнал b – другим, а сигнал c – останім. Після того, як сигнал X набув значення 1, воно зберігається доти, поки сигнали не будуть набирати значення 0 з такою ж самою черговістю.

2.18 Схема має три вхідні сигнали a, b, c і один вихідний X . Сигнал $X=1$, якщо усі три вхідні сигнали мають значення 1, причому сигнал a набуває значення 1 останнім. Після того, як сигнал X набув значення 1, воно зберігається доти, поки $a=1$, незалежно від значень сигналів b і c .

2.19 Схема забезпечує керування двома двигунами $M1$ та $M2$. При натисненні кнопки «Старт» обидва двигуни починають обертатися. В залежності від того який кінцевий вимикач SQ_1 чи SQ_2 спрацює перший, двигун $M1$ чи $M2$ зупиниться, а інший продовжить обертатися. Після натиснення SQ_3 інший двигун також вимикається.

2.20 Схема забезпечує керування двома двигунами $M1$ та $M2$. При натисканні кнопки «Старт» двигун $M1$ починає обертатися. При спрацюванні кінцевого датчика SQ_1 , $M1$ зупиняється, а $M2$ обертається доки не надійде сигнал SQ_3 , після чого знову вмикається $M1$, а $M2$ продовжує свою роботу. При спрацюванні датчика SQ_2 двигуни $M1$ та $M2$ зупиняються.

2.21 Схема забезпечує роботу в ручному та автоматичному режимі. При обранні автоматичного режиму двигун виконує наступну послідовність дій: обертається до надходження сигналу SQ_1 , стоїть до надходження сигналу SQ_2 , обертається в протилежну сторону до надходження сигналу SQ_3 . При обранні

ручного режиму, двигун керується кнопкою «Старт». Якщо «Старт» натиснута, то двигун обертається, якщо ні, то двигун стоїть.

2.22 Виконати синтез схеми, що має чотири вхідних сигналів $a_1 \dots a_4$ і чотири вихідних $b_1 \dots b_4$. Система відпрацьовує наступну послідовність вхідних та вихідних сигналів:

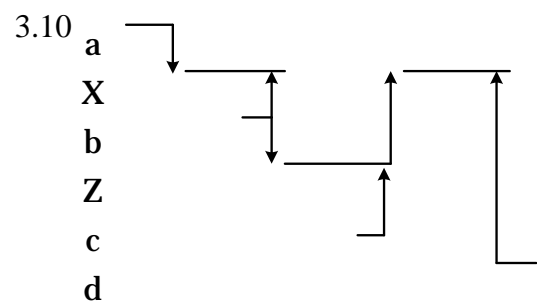
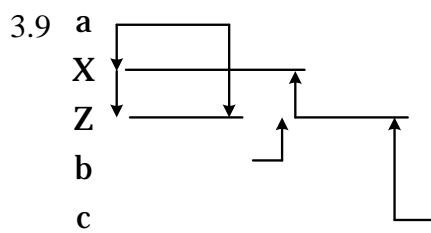
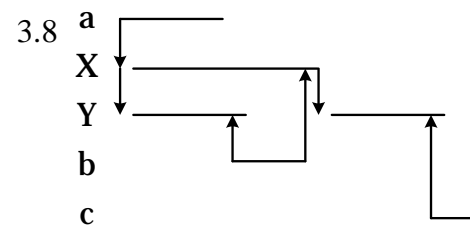
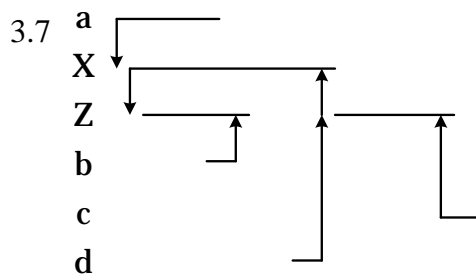
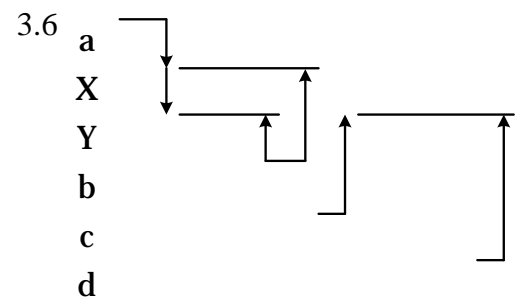
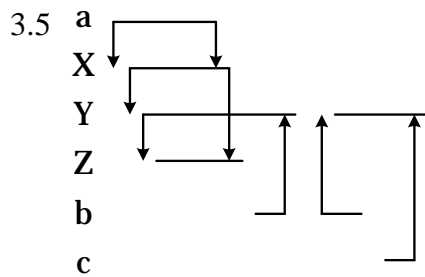
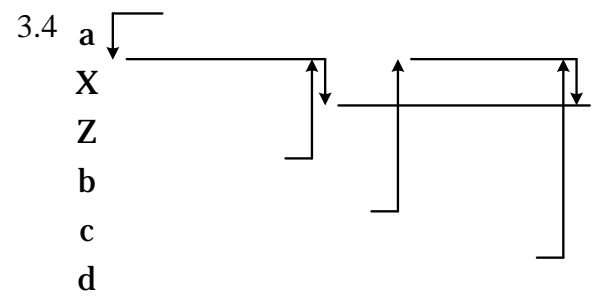
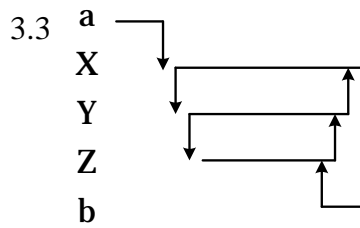
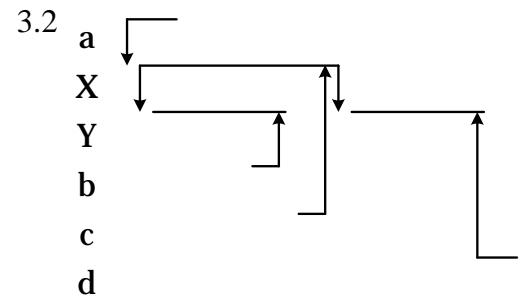
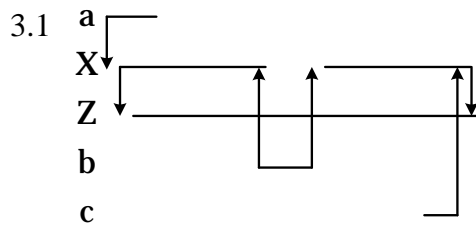
$$a_1 a_2 a_3 a_4 : 0000 \rightarrow 1000 \rightarrow 0000 \rightarrow 0100 \rightarrow 0000 \rightarrow 0010 \rightarrow 0000 \rightarrow 0001 \rightarrow 0000$$

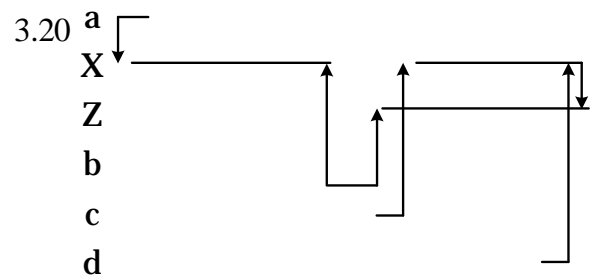
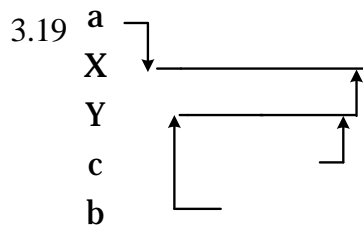
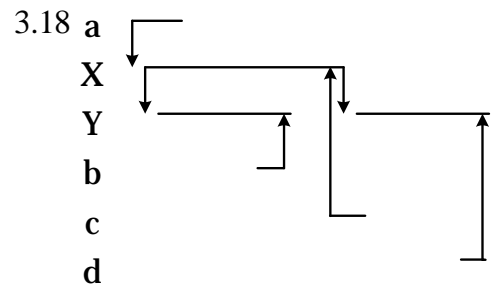
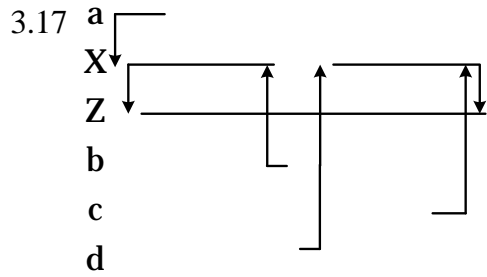
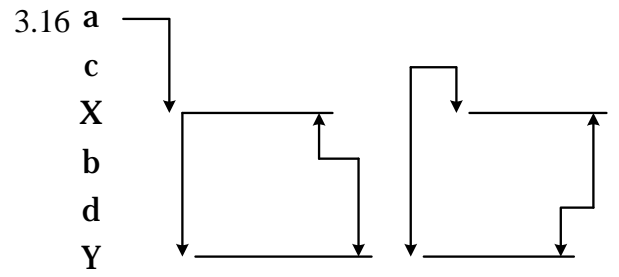
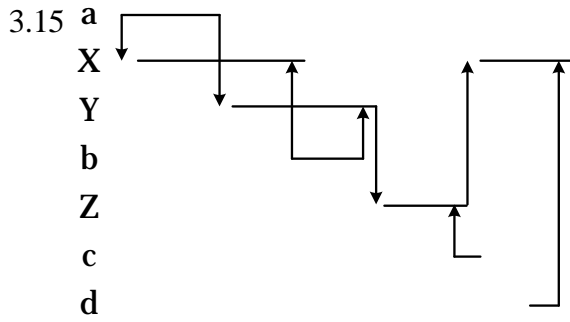
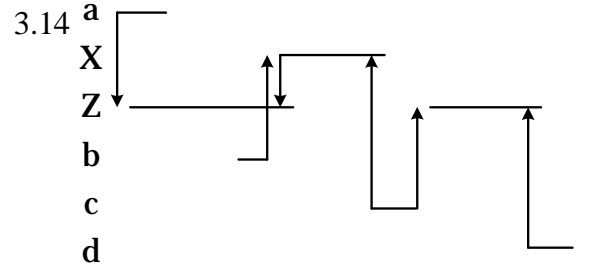
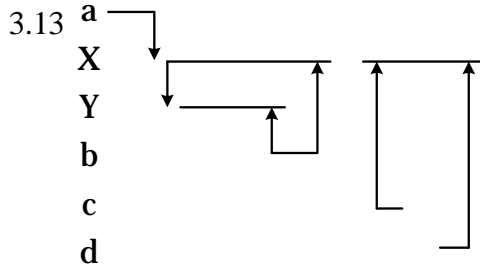
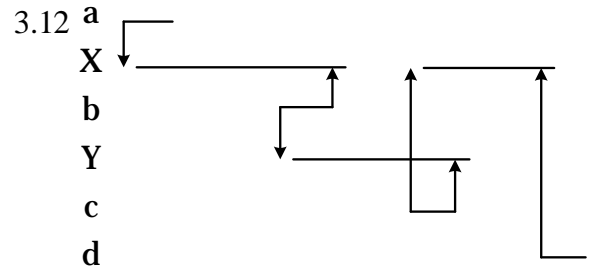
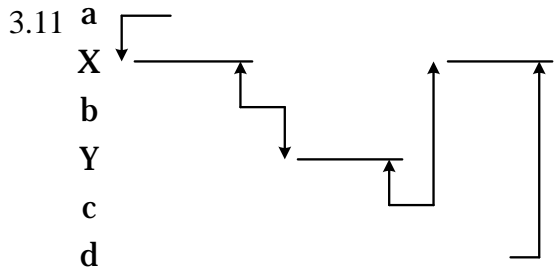
$$b_1 b_2 b_3 b_4 : 0000 \rightarrow 0001 \rightarrow 0011 \rightarrow 0111 \rightarrow 1111 \rightarrow 1110 \rightarrow 1100 \rightarrow 1000 \rightarrow 0000$$

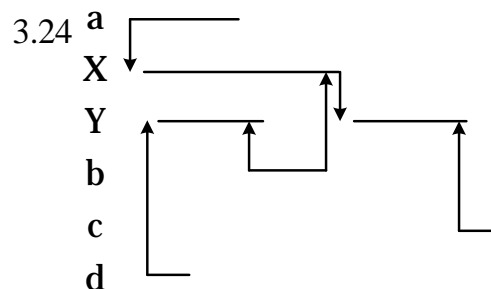
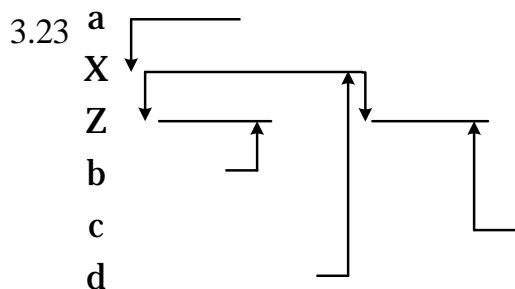
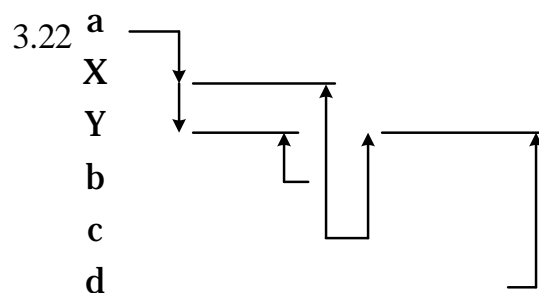
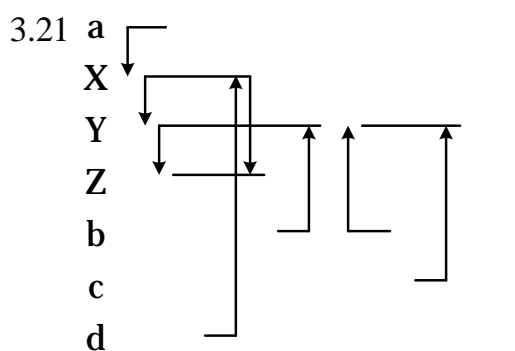
2.23 Схема забезпечує керування чотирма лампами за допомогою однієї кнопки. У вихідному стані світяться третя і четверта лампи. Якщо натиснути кнопку перший раз, то будуть світитися друга і третя лампи, а четверта погасне. Після відпускання кнопки стан схеми не змінюється. Якщо натиснути кнопку другий раз, то будуть світитися перша і друга лампи, а третя погасне. Якщо натиснути кнопку третій раз, то будуть світитися всі чотири лампи. Якщо натиснути кнопку четвертий раз, то схема набуває вихідний стан. Після цього цикл роботи схеми повторюється.

2.24 Схема забезпечує керування чотирма лампами за допомогою двох кнопок. При натисненні першої кнопки вмикається перша лампа, при натисненні другої, не відпускаючи першої, вмикається друга лампа. При відпусканні другої кнопки перша лампа вимикається, а при відпусканні першої – вимикається друга.

3. Виконати синтез схеми за умовами роботи, заданими у вигляді циклограм, та перевірити її працездатність.







4. Виконати синтез генератора десяткових чисел на основі JK-тригерів та перевірити її роботу.

Прямокутні сигнали на вхід схеми краще подавати двома кнопками, використавши на вході RS-тригер, схему якого було розглянуто вище.

Цей прийом також можна використати в завданні 2, де схема реагує як на передній так і на задній фронт імпульсу керуючого сигналу. Оскільки швидкодія мікросхеми досить велика, то на роботу схеми можуть впливати навіть такі швидкі зміни сигналу на вході які виникають через брязкіт контактів при замиканні контакту кнопки.

Задані послідовності двійкових чисел:

4.1 0,1,2,3,4,4,3,2,1,0,1,2...

4.2 0,1,2,3,5,5,3,2,1,0,1,2...

4.3 0,4,2,3,6,6,3,2,4,0,4,2...

4.4 0,1,2,3,7,7,3,2,1,0,1,2...

4.5 0,1,2,4,5,5,4,2,1,0,1,2...

4.6 0,2,4,5,7,7,5,4,2,0,2,4...

4.7 0,1,2,4,7,7,4,2,1,0,1,2...

4.8 0,1,2,5,6,6,5,2,1,0,1,2...

- 4.9 0,3,2,5,7,7,5,2,3,0,1,2...
- 4.10 0,1,2,6,7,7,6,2,1,0,1,2...
- 4.11 0,1,3,4,5,5,4,3,1,0,1,3...
- 4.12 0,2,3,5,6,6,5,3,2,0,2,3...
- 4.13 0,1,3,4,6,6,4,3,1,0,1,3...
- 4.14 0,1,4,5,7,7,5,4,1,0,1,4...
- 4.15 0,1,4,6,7,7,6,4,1,0,1,4...
- 4.16 0,3,5,6,7,7,6,5,3,0,3,5...
- 4.17 0,1,3,4,6,6,4,3,1,0,1,3...
- 4.18 0,7,2,5,3,3,5,2,7,0,7,2...
- 4.19 0,7,2,6,3,3,6,2,7,0,7,2...
- 4.20 0,5,3,4,6,6,4,3,5,0,5,3...
- 4.21 0,7,3,5,6,6,5,3,7,0,7,3...
- 4.22 0,2,3,4,6,6,4,3,2,0,2,3...
- 4.23 0,4,1,5,7,7,5,1,4,0,4,1...
- 4.15 0,3,1,2,7,7,2,1,3,0,3,1...

8.4 Опис лабораторної установки

Функціональна схема (рис. 8.5) стенду відображає одне робоче місце, яке виділяється кожному студенту бригади.

На рисунку представлено наступне обладнання:

ПЛІС Altera – призначена для видачі керуючих сигналів на схеми керування промисловими механізмами. Являє собою програмовану логічну схему, яка реалізовує задані користувачем логічні функції.

Блок живлення – призначений для живлення ПЛІС напругою +5В.

Програматор – призначений для перетворення сигналу що надходить від USB порту персонального комп'ютера в сигнал, що дозволяє програмувати ПЛІС в JTAG-режимі.

Персональний комп'ютер – призначений для роботи з програмним забезпеченням Quartus II, компіляції проектів та конфігурації ПЛІС.

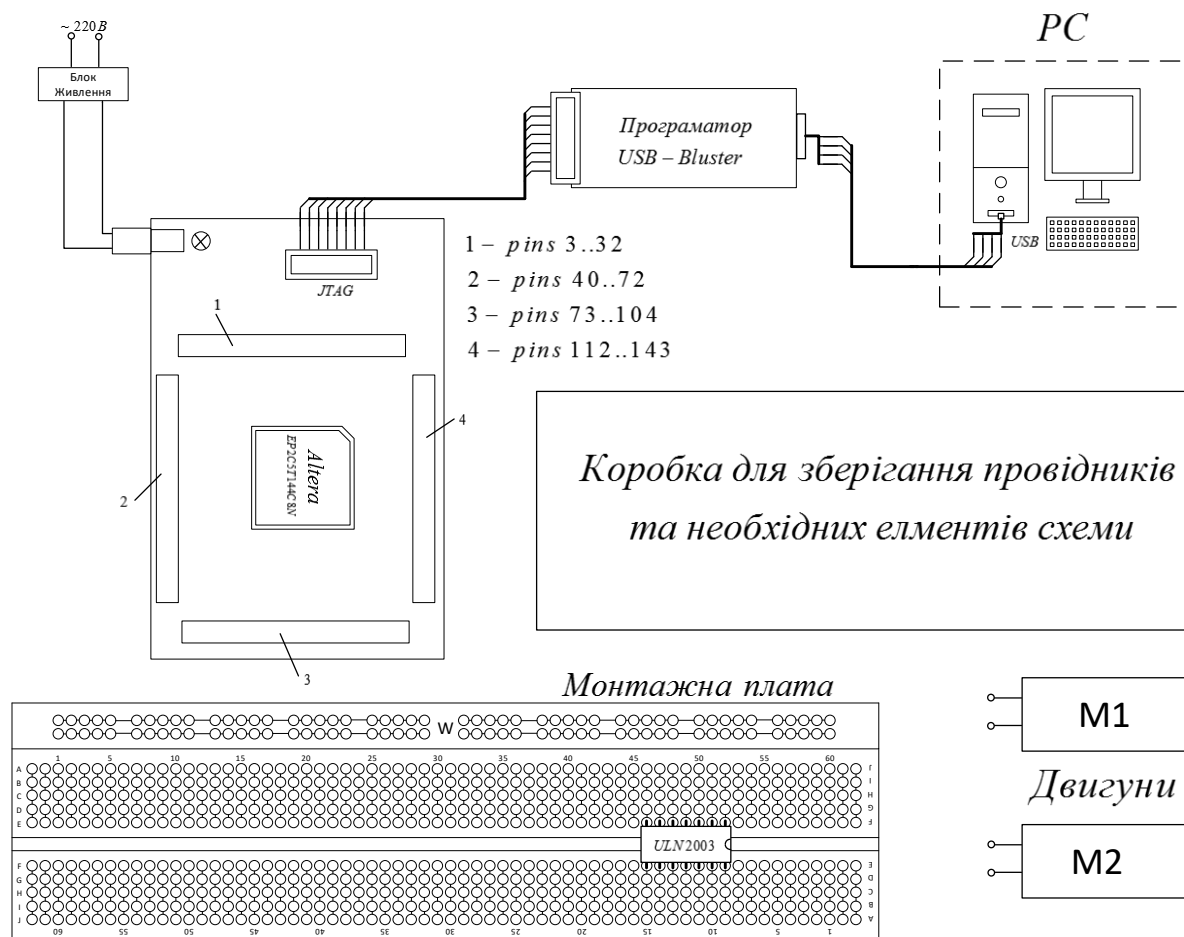


Рисунок 8.5 – Функціональна схема лабораторного стенду

Монтажна плата – призначена для розміщення кнопок, світлодіодів, резисторів та їх з'єднання з входами/виходами ПЛІС.

М1, М2 двигуни напругою живлення 5В. Призначені для більш яскравої індикації вихідних сигналів.

До комплекту, що видається студенту, входять: набір провідників для з'єднання входів/виходів ПЛІС з монтажною платою, кнопки (для емуляції вхідних сигналів) світлодіоди (для емуляції вихідних сигналів), резистори (для «підтяжки» кнопок на GND та обмеження струму через світлодіоди), та мікросхема МК ULN2003 для підсилення вихідного сигналу з ПЛІС (застосовується для керування низьковольтними мікродвигунами М1 та М2).

Позначення виводів мікросхеми Altera Cyclone II EP2C5T144C8N представлена на рис. 8.6.

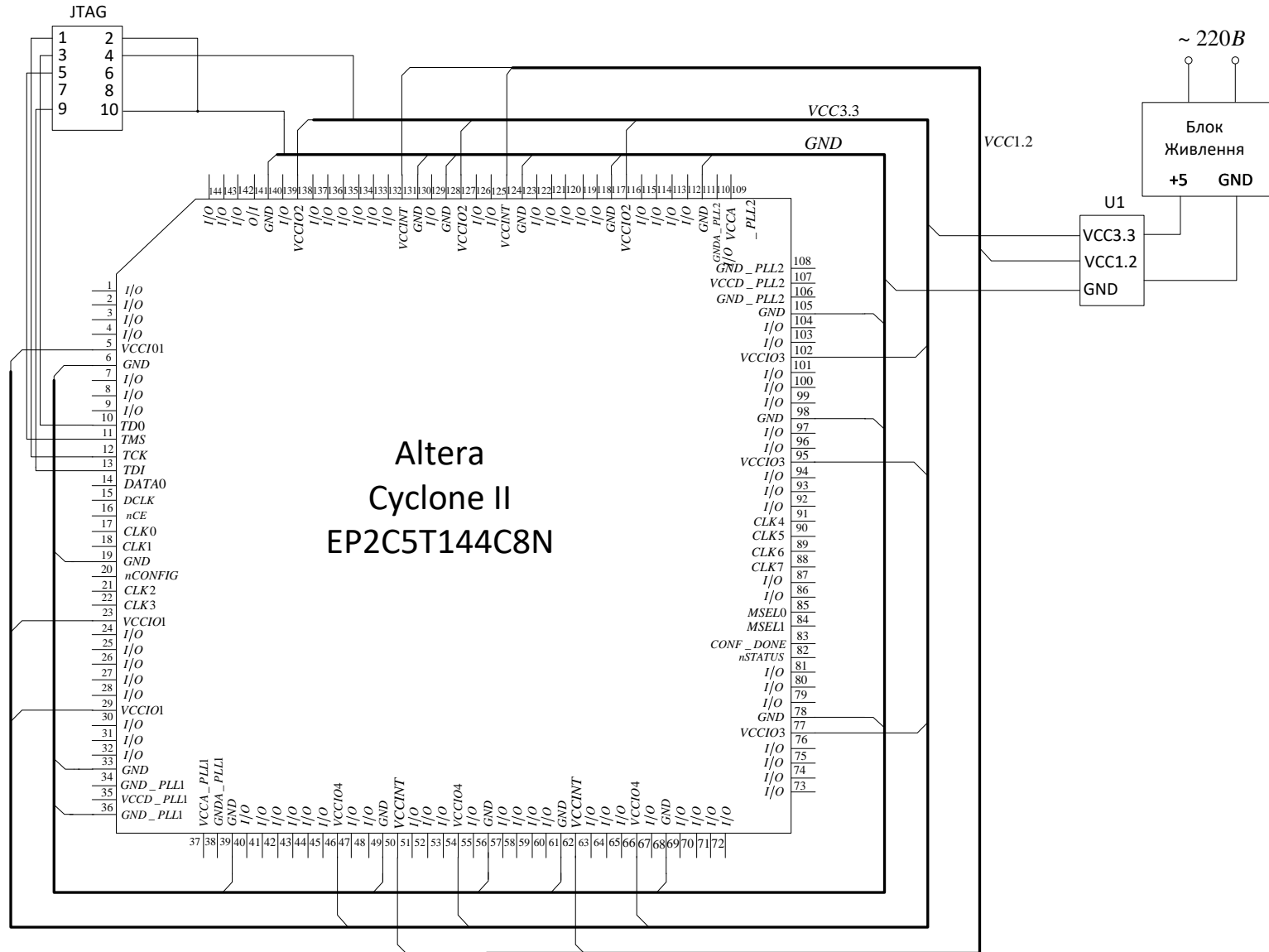


Рисунок 8.6 – Позначення виводів мікросхеми Altera Cyclone II EP2C5T144C8N

Опис програмного пакета QUARTUS II V. 9.1

Створення нового проекту

Відкривши пакет Quartus II вибираємо з меню File пункт New Project Wizard... – майстер створення нових проектів. Потрапляємо у вікно для завдання поточної директорії проекту. Заповнюємо три рядки як показано на рис.8.7 і натискаємо кнопку Next.

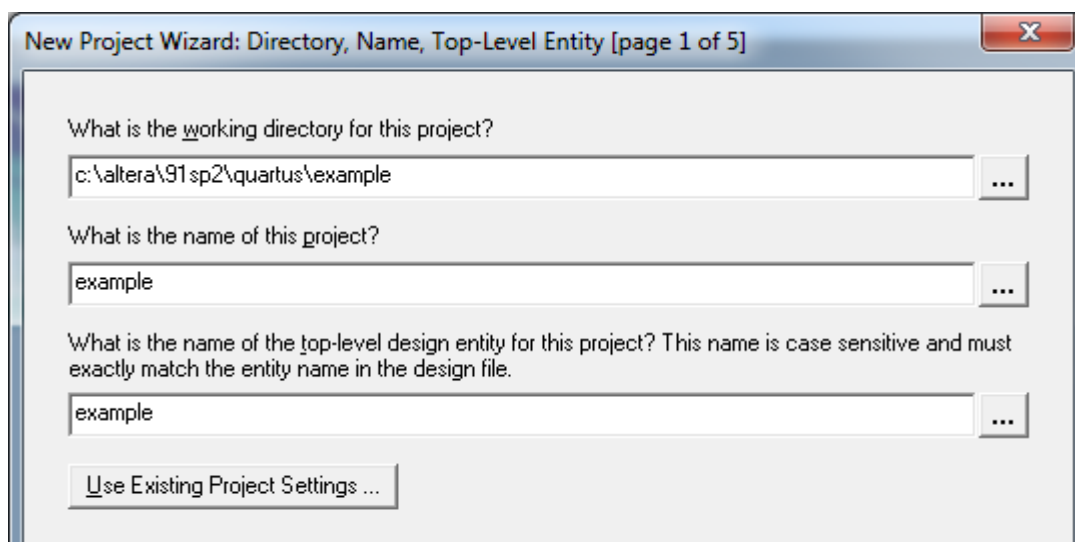


Рисунок 8.7 – Меню завдання поточної директорії проекту

В даному випадку проект буде названо example і буде створена директорія за адресою c:\altera\91sp2\quartus\example.

Після натиснення кнопки Next з'являється вікно де потрібно підтвердити створення нової директорії.

З'являється вікно де можна вибрати вже існуючий файл з програмою. Натискаємо кнопку Next, пропускаючи це вікно.

Далі потрапляємо у вікно де потрібно вибрати сімейство та модель мікросхеми. Модель мікросхеми можна подивитись безпосередньо на стенді, вибрати відповідний пункт у вікні, як це і показано на рис. 8.8 та натиснути кнопку Finish. Після цього новий проект створено.

Для того щоб додати новий файл до проекту вибираємо з меню File пункт New. У вікні що з'явилося обираємо мову на якій буде написана програма. Вибираємо BlockDiagram/SchematicFile та натискаємо ОК, рис.8.9. Новий файл створено.

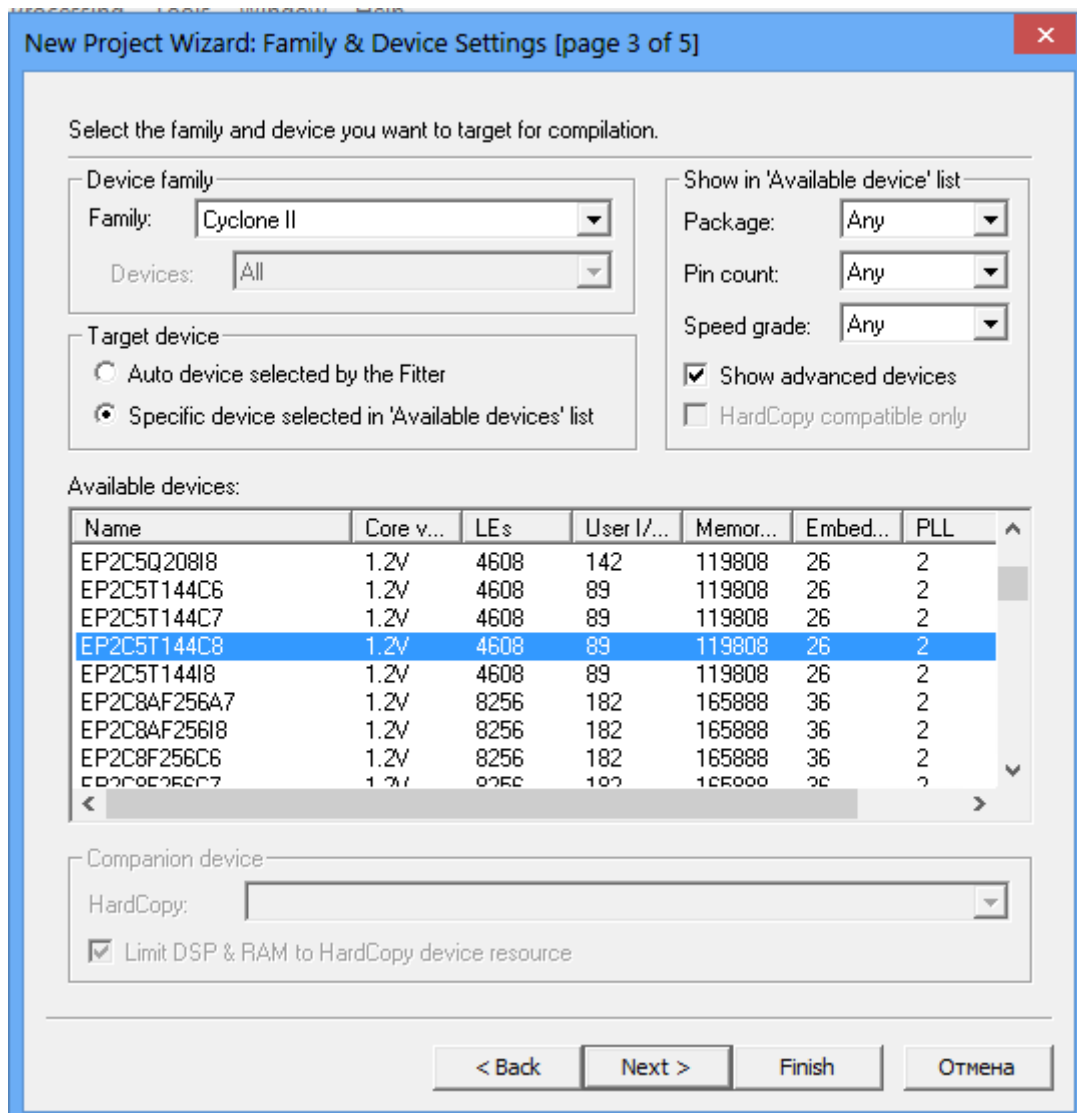


Рисунок 8.8 – Вікно вибору сімейства та моделі мікросхеми

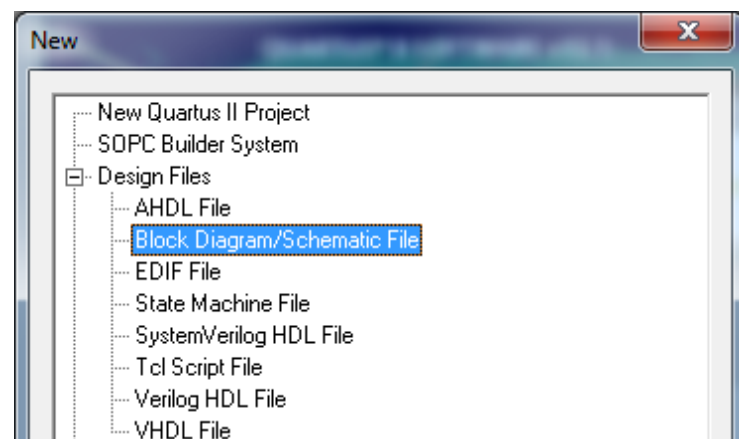


Рисунок 8.9 – Вікно вибору мови програмування та типу файлу

Створення схеми проекту

Після створення файлу проекту стає активною панель інструментів розташована ліворуч від робочої області файлу і представлена на рис. 8.10. В панелі інструментів представлені засоби для створення проекту. Інструмент SymbolTool зображений на рис. 8.10 у верхньому правому куті, натиснемо на його іконці лівою кнопкою миші, в результаті нам відкриється вікно Symbol (рис. 8.11) У лівому верхньому куті представлений список стандартних бібліотек Quartus II, з основними типами елементів, що застосовуються на практиці.

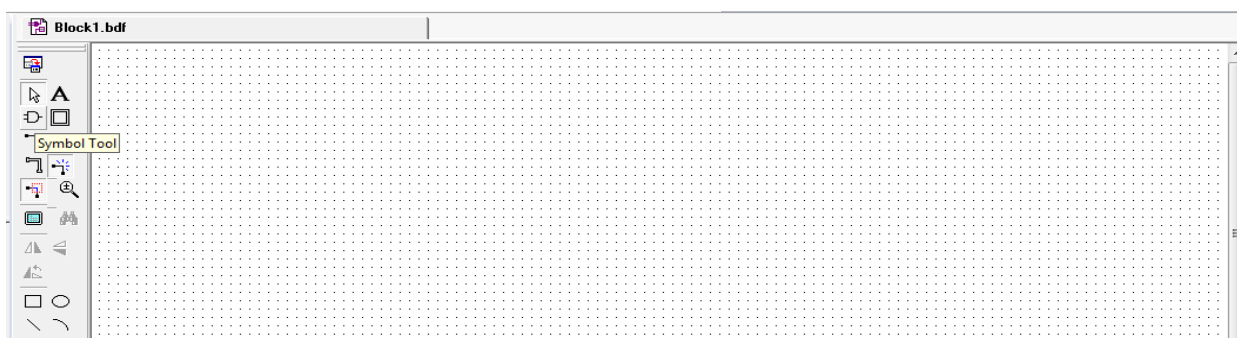


Рисунок 8.10 – Робоча область та панель інструментів

Як показано на рис.8.11 з бібліотеки виберемо primitives / pin / input для завдання вхідних пінів. Зображення компоненти з'являється в правому вікні. Натиснемо кнопку ОК, і приступимо до безпосереднього розташуванню обраного компонента.

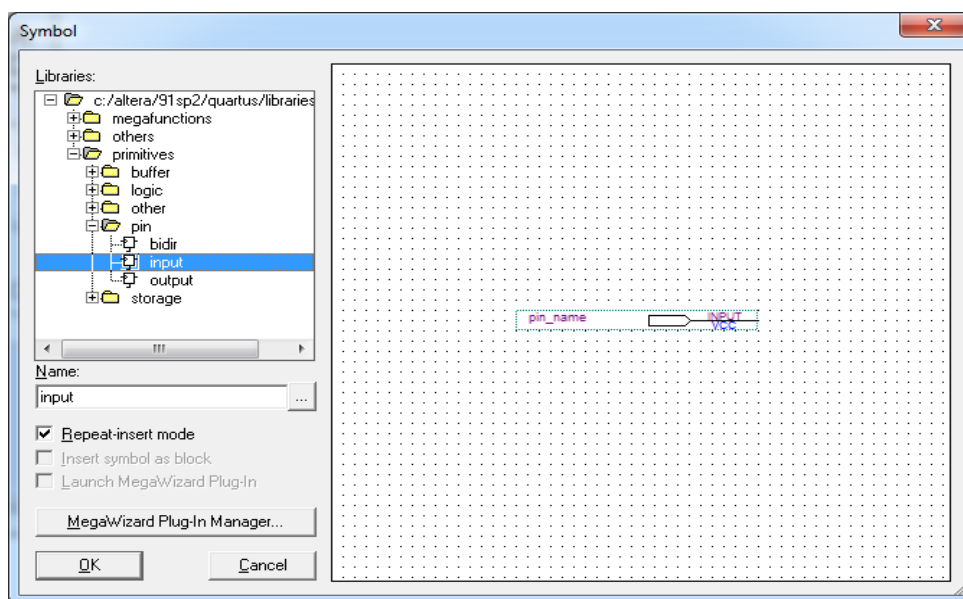


Рисунок 8.11 – Вікно Symbol

Після розташування достатньої кількості обраних компонентів в робочій області файлу Block1.bdf натиснемо праву кнопку миші і з контекстного меню

виберемо пункт Cancel, скасувавши подальше введення компоненти. Повторно скориставшись інструментом SymbolTool встановимо крім компоненти входу input, компоненту виходу output.

В бібліотеці primitives/logic можна знайти елементи, які будуть реалізувати такі логічні функції:

and – логічне «І»

or – логічне «АБО»

not – інверсія

xor – виключаюче «АБО»

Цифра після назви логічної функції означає кількість входів, яку буде мати елемент, наприклад and3 буде мати 3 входи, рис.8.12(а). Наявність букви b перед назвою логічної функції буде означати, що елемент буде мати інверсні входи, рис. 8.12(б), а букви n – інверсний вихід, рис. 8.12(в). Наявність обох букв означатиме, що елемент матиме інверсні як входи так і вихід, рис. 8.12(г).

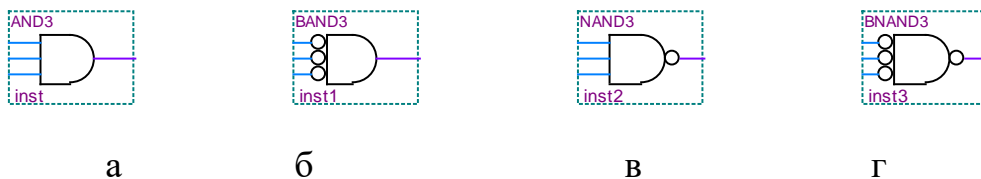


Рисунок 8.12 – Приклади логічних елементів

Побудову схеми розглянемо на прикладі створення асинхронного RS-тригера з двох елементів NOR2. Зробивши установку необхідного числа елементів зробимо з'єднання входів і виходів компонент, помістивши курсор миші на один з входів елемента NOR2, затиснемо ліву кнопку миші і з'єднаємо і вихідним кінцем елемента входу. Проведемо аналогічну операцію для всієї схеми і отримаємо схему наведену на рис.8.13.

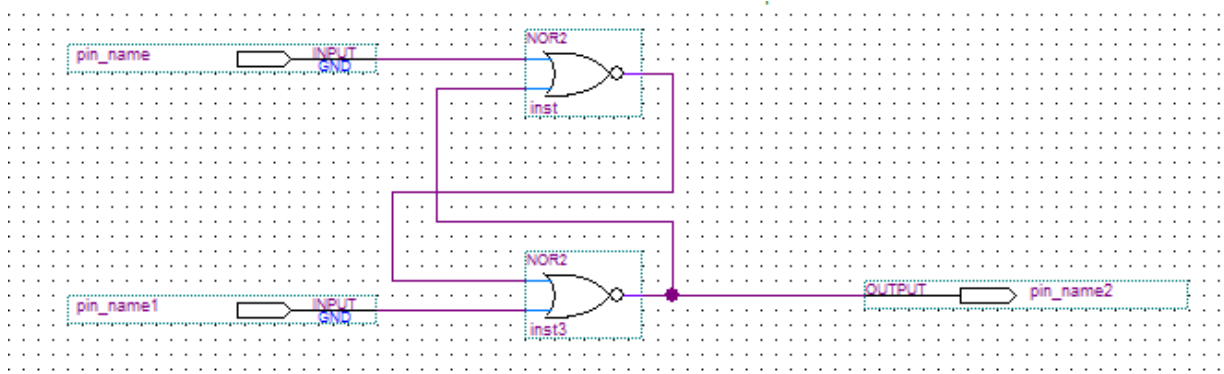


Рисунок 8.13 – Схема поточного проекту

Переіменуємо елементи. Для цього зробимо спочатку подвійне натискання лівою кнопкою миші по одному з елементів input. У результаті відкриється вікно зображене на рис.8.14, що представляє властивості обраного Піна. У закладці General вміст рядка Pinname (s) змінимо на S, таким чином присвоївши ім'я S першому входу. Рядок Defaultvalue являє собою значення логічного сигналу на pins за замовчуванням, в даному випадку VCC тобто пін «підтягнутий» на плюс живлення. Можливо змінити значення за замовчуванням на GND або нуль. Це визначається побудовою схеми, і в нашому випадку необхідно встановити GND. Подібним чином змінимо ім'я другого вхідного піна на R, а вихідного на Q. Зміни імен відразу ж будуть відображатись на схемі.

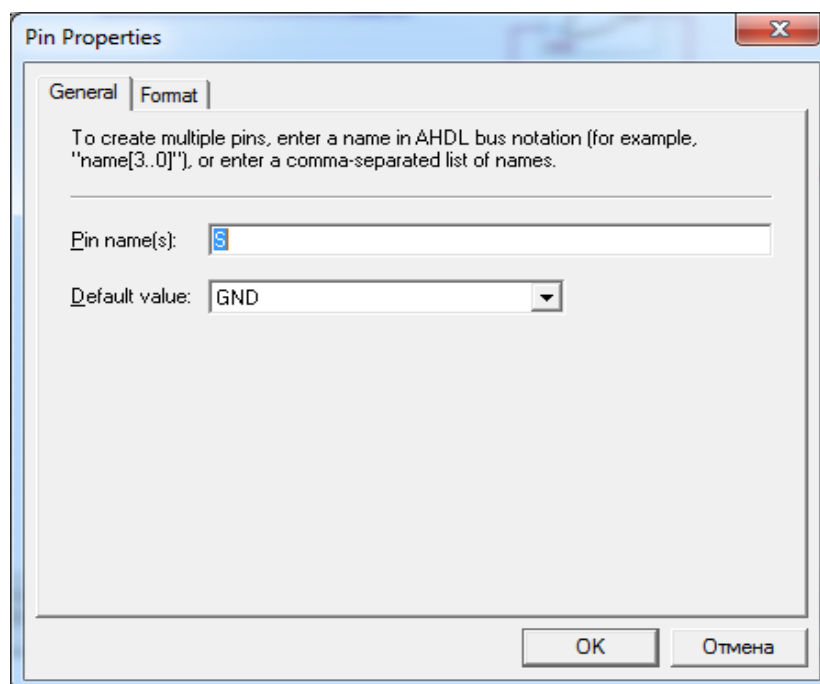




Рисунок 8.14 – Властивості піна

Далі необхідно виконати розпіновку. Перед цим необхідно виконати компіляцію проекту. Для цього необхідно натиснути кнопку StartCompilation .

По закінченні компіляції з'являється вікно з повідомленням про результати компіляції, кількості помилок і попереджень. Далі натискаємо кнопку Pin Planer . Відкривається вікно зображене на рис. 8.15, де необхідно відповідно до схеми приведеної на рис 8.6 встановити вхідні змінні на ті виводи до яких підключено кнопки, а вихідні змінні на ті до яких підключено світлодіодні індикатори. Для цього необхідно два рази клацнути на необхідний вивід та вибрати з переліку назву змінної яку хочемо присвоїти вибраному виводу.

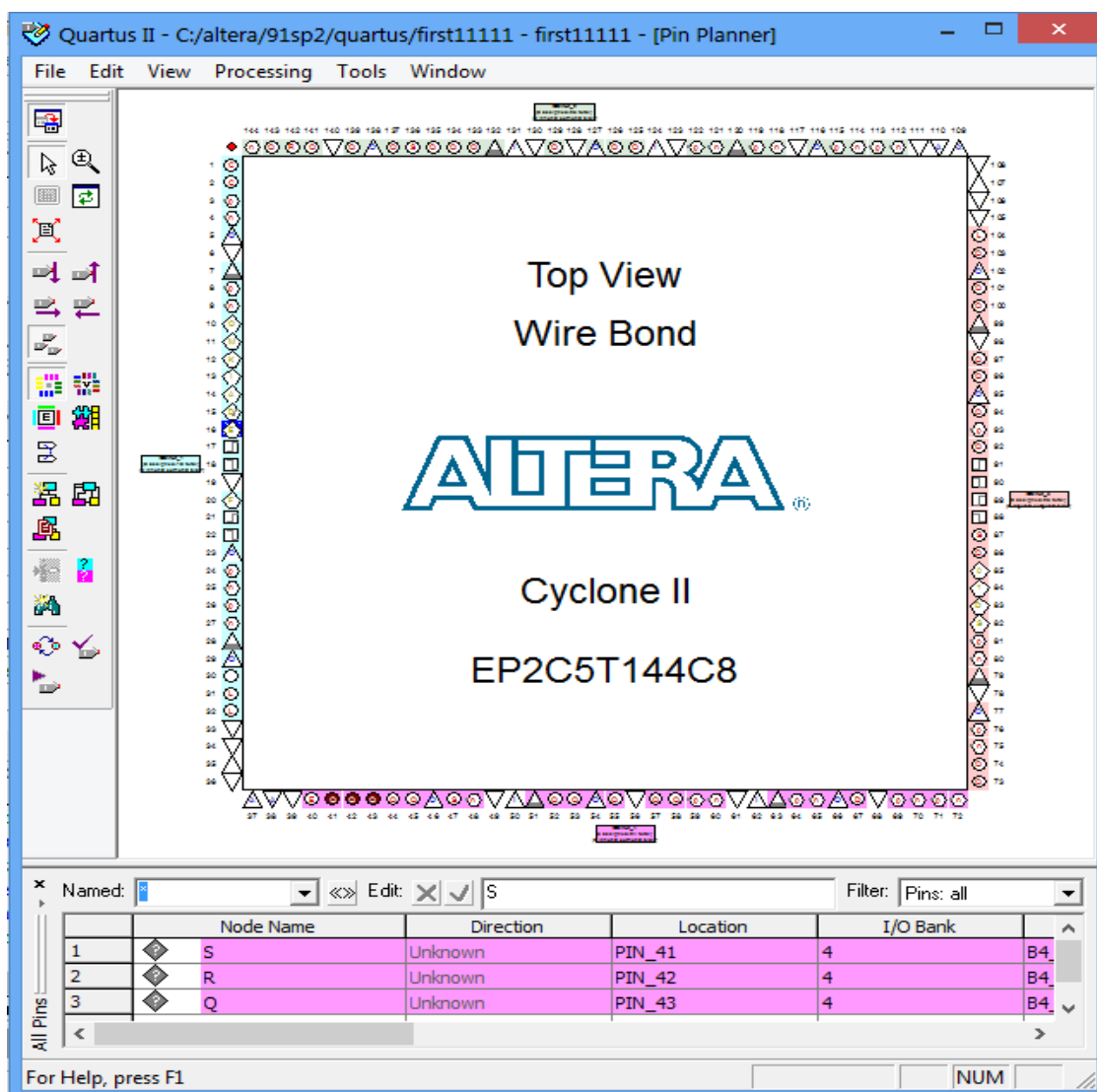


Рисунок 8.15 – Панель призначення виводів мікросхеми

Обирати слід виводи позначені всередині буквою Q

Далі панель розпіновки можна закрити. Результати виконаних операцій відразу будуть відображені на схемі, рис. 8.16.

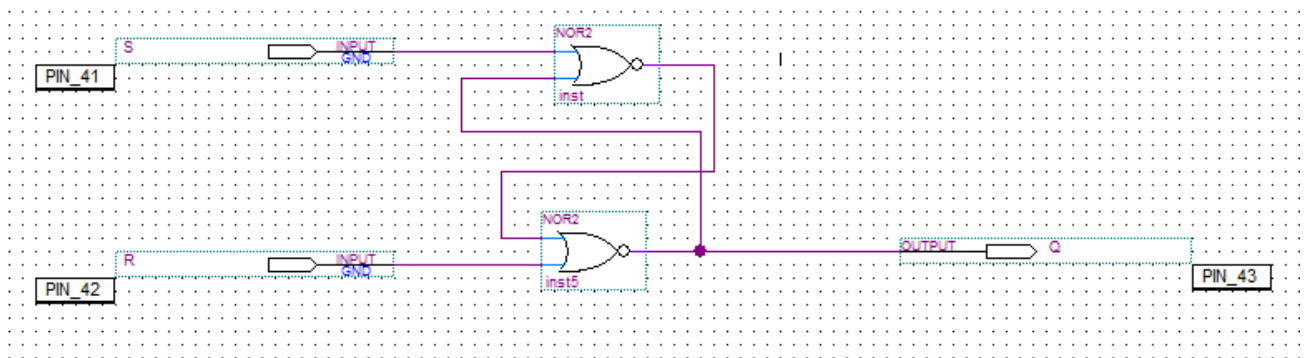


Рисунок 8.16 – Схема поточного проекту

Після цього бажано виконати компіляцію ще раз.

Проведення часової симуляції

Для початку симуляції потрібно задати файл часових діаграм. Для цього вибираємо з меню File пункт New і вибираємо тип файлу VectorWaveformFile. З'являється вікно зображене на рис.8.17.

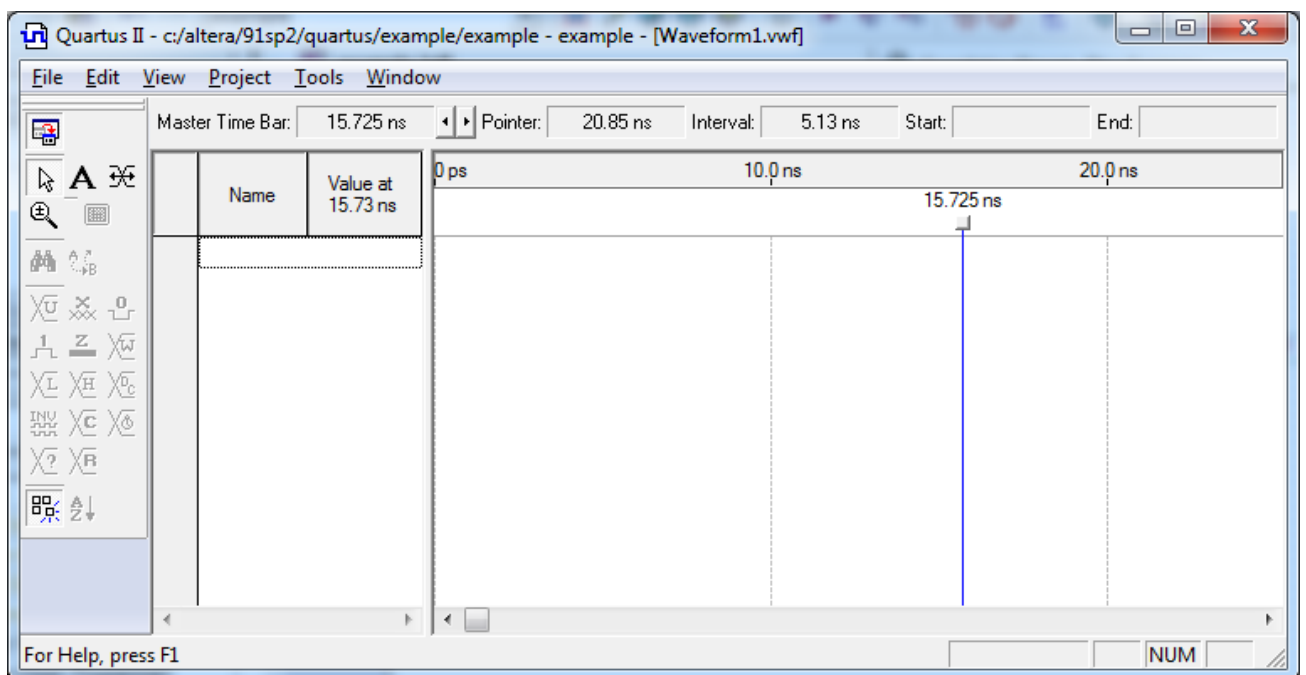


Рисунок 8.17 – Файл часових діаграм

Нам належить завантажити список входів і виходів для яких потім виконаємо симуляцію. Для цього зробимо подвійне натискання в лівому вільному лівому полі. З'являється вікно InsertNodeBus в якому натискаємо кнопку NodeFinder... У вікні NodeFinder натискаємо кнопку List. Бачимо, що у

вікні NodesFound: з'явився список пінів, який ми за допомогою кнопки >> перенавантажуюмо у вікно SelectedNodes:, натискаємо ОК. І підтверджуємо вибір натисканням ОК у вікні InsertNodeBus. В результаті виконаних операцій отримуємо вікно симуляції зображене на рис. 8.18.

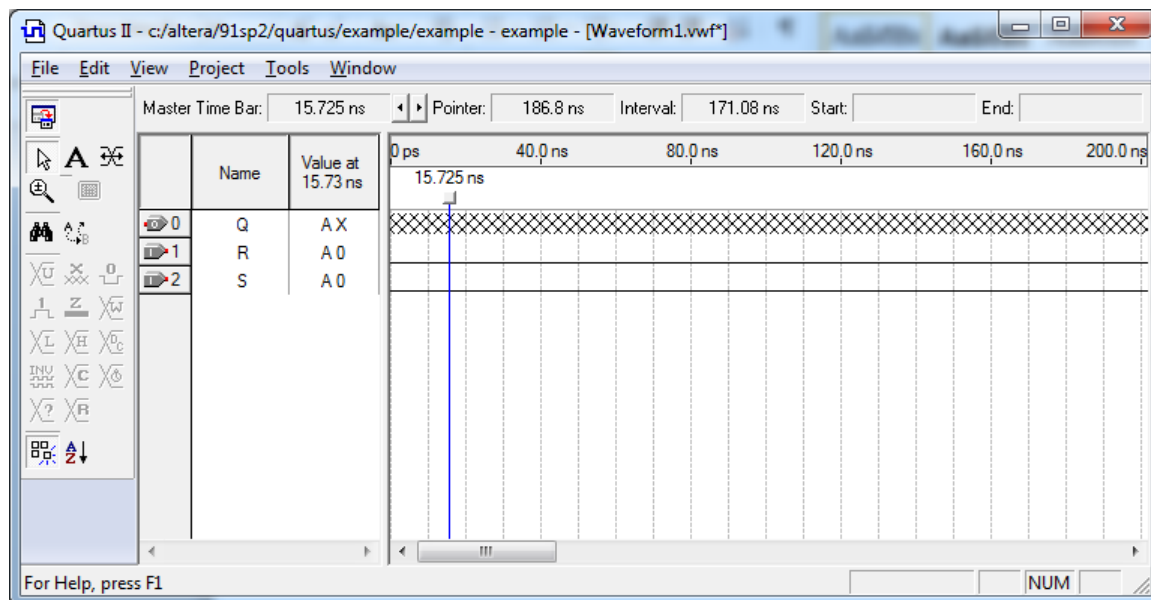



Рисунок 8.18 – Вікно симуляції після введення вектора сигналів

Виділяючи необхідні сигнали на необхідному інтервалі часу, за допомогою натискання й утримання лівої кнопки миші, задаємо на даному інтервалі необхідні логічні рівні з меню сигналів, розташованого безпосередньо зліва віділеної області.

Після задання всіх рівнів сигналів зберігаємо файл вектора сигналів кнопкою  і вибираємо пункт StartSimulation з меню Processing і очікуємо результатів симуляції, представлених, наприклад, на рис. 8.19.

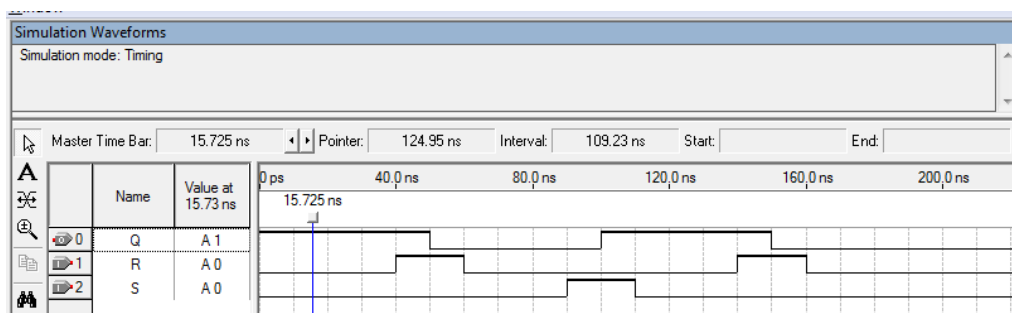



Рисунок 8.19 – Результати часової симуляції

Програмування мікросхеми

Для доступу до меню програмування виберемо Tools-Programmer (або значок  на панелі інструментів). У вікні параметрів програмування

(рис. 8.20), встановимо галочки під пунктами: **Programm/Configure** – запис програми і конфігурації та **Verify**– з перевіркою результату.

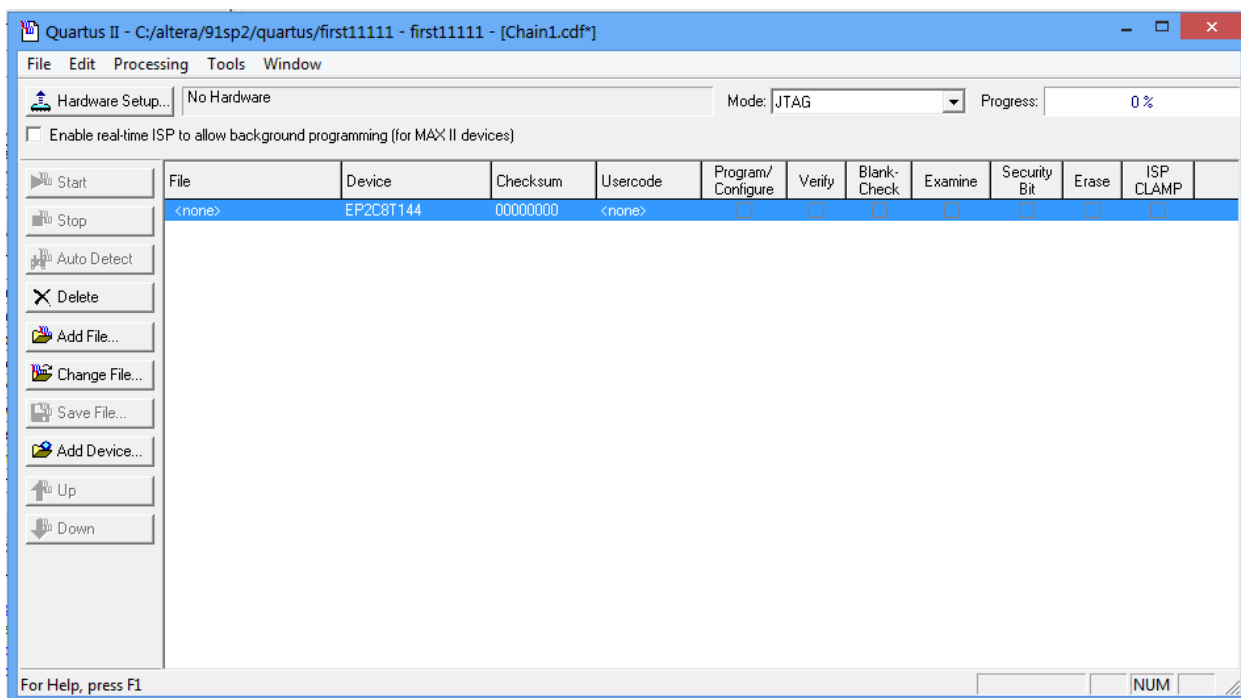


Рисунок 8.20 – Вікно параметрів програмування

В меню HardwareSetup (рис. 8.21) вибирається тип пристрою програмування.

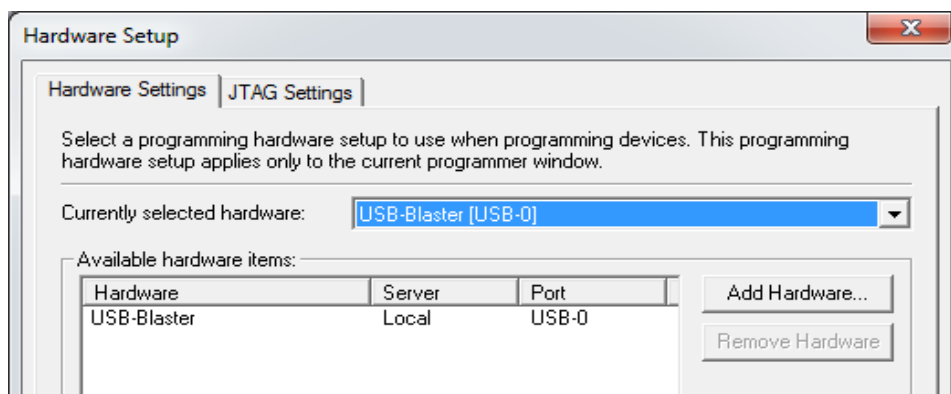


Рисунок 8.21 – Вікно вибору пристрою для програмування

Далі натискаємо кнопку Start і в вікні Progress спостерігаємо за процесом програмування.

Мікросхема ULN2003N

Мікросхема містить сім незалежних транзисторних пар Дарлінгтона в одному корпусі зображено на рис. 8.22. Пара Дарлінгтона – каскад з двох біполярних транзисторів. Чіп дозволяє за допомогою слабкого току ПЛІС,

такого як Altera, керувати потужністю навантаження та току до 500мА та напруги до 50В на канал. Для кожного каскаду встановлений зворотній діод для запобігання пробією або обриві індуктивного навантаження. При необхідності можливо об'єднати у паралель виходи для отримання максимального струму. В даній роботі мікросхема використовується для живлення двох двигунів постійного струму.

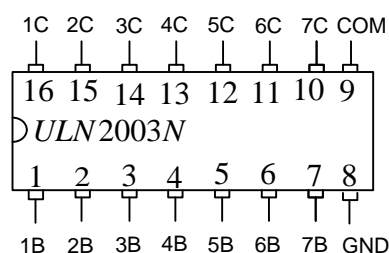


Рисунок 8.22 – Мікросхема ULN2003N

GND – «земля» живлення;

COM – спільний вихід усіх каналів;

1B-7B – входи мікросхеми;

1C-7C – виходи мікросхеми;

Кожен каскад має вигляд зображений на рисунку 4.23. Суть полягає в тому, що один транзистор відкриває інший. А разом вони працюють як один транзистор з коефіцієнтом підсилення струму рівним добутку коефіцієнтів першого і другого транзистора.

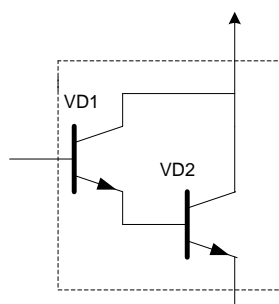


Рисунок 8.23 – Транзистор Дарлінгтона

Характеристики

- Керуюче навантаження: 5 В;
- Максимальне вихідне навантаження: 50 В;
- Максимальний вихідний струм: 500 мА на канал;

8.5 Методичні вказівки до виконання роботи

1. Для подачі живлення +5В на монтажну плату необхідно провідниками з'єднати будь-які виводи на ПЛІС, що позначені GND (земля) та V_{CC} (+5В) із відповідними контактами «-» та «+» на монтажній платі.

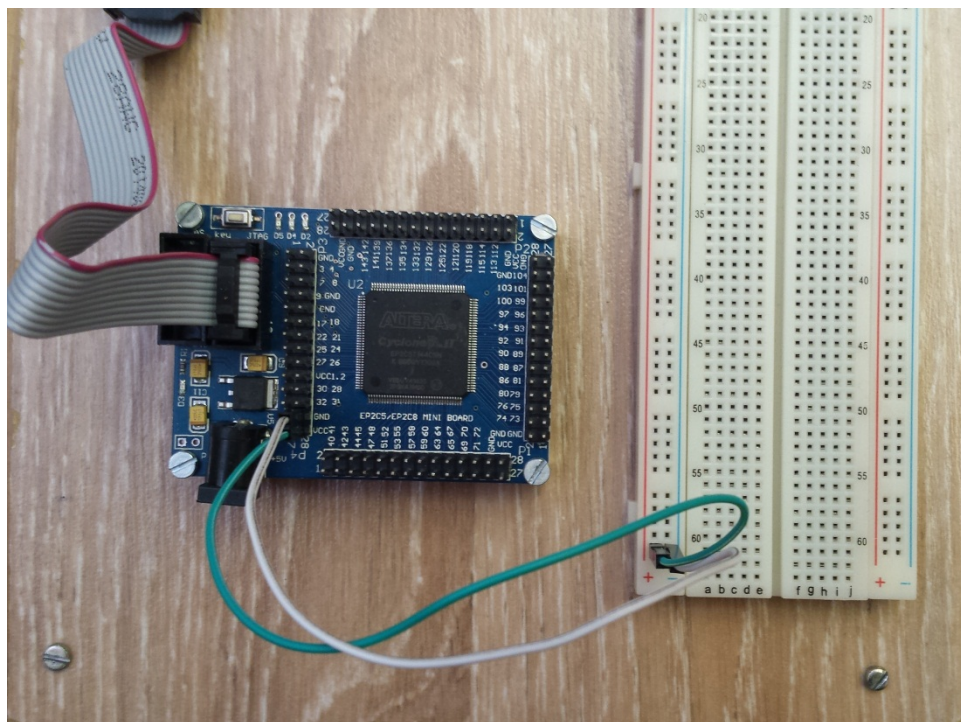


Рисунок 8.24 – Підключення живлення плати

2. Всі входні сигнали необхідно «підтягнути» на контакт GND.

3. Двигуни M1 та M2 споживають більший струм, ніж на виході ПЛІС, тому їх необхідно підключати до виходів через мікросхему ULN2003N. Підключення мікросхеми та двигунів зображено на рис. 8.25. Де a, b – керуючі сигнали з виходів ПЛІС на включення двигунів M1 та M2. Двигуни мають два виводи один з яких підключається на землю, а другий на вихід мікросхеми ULN2003N.

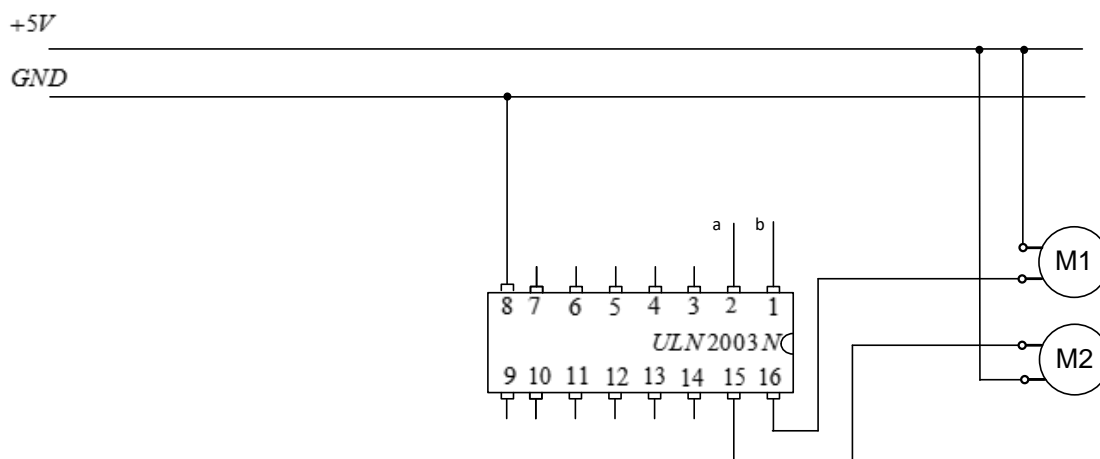


Рисунок 8.25 – Приклад підключення мікросхеми ULN2003N

Приклад 1. Скласти схему керування, яка буде забезпечувати роботу механізму, принцип роботи якого описано нижче. Під час роботи поздовжньо-стругального верстата його стіл періодично переходить з крайнього лівого в крайнє праве положення. Переміщення стола обмежується кінцевими вимикачами SQ1 (ліворуч) і SQ2 (праворуч). Рух починається після натиснення кнопки “Пуск” (рух праворуч). Робоча зупинка здійснюється кнопкою “Стоп 1”. Після натиснення цієї кнопки стіл, перед тим як зупинитися, повинен дістатися крайнього лівого положення. Аварійна зупинка здійснюється кнопкою “Стоп 2” (стіл зупиняється у будь-якому положенні).

Вводимо такі позначення: X – сигнал на рух праворуч, Y – сигнал на рух ліворуч, а–“Пуск”, б–“Стоп 1”, с – “Стоп 2”. Графоперехід за умовами роботи схеми представлений на рис. 8.26.

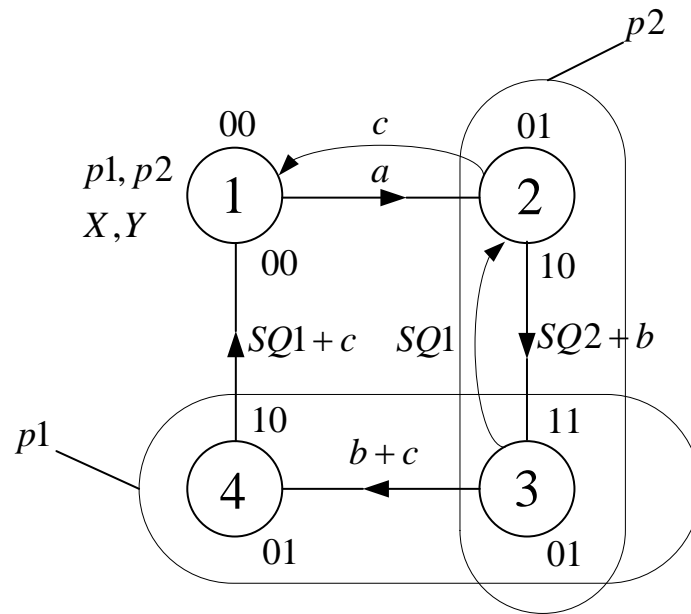


Рисунок 8.26 – Графоперехід за умовами роботи схеми

Рівняння для встановлення тригерів:

$$S_{p_1} = (SQ2 + b)p_2$$

$$S_{p_2} = a\bar{p}_1$$

Рівняння для скидання тригерів:

$$R_{p_1} = SQ1p_2 + (c + SQ1)\bar{p}_2$$

$$R_{p_2} = c\bar{p}_1 + (c + b)p_1$$

Рівняння для вихідних змінних:

$$X = \bar{p}_1p_2$$

$$Y = p_1p_2 + p_1\bar{p}_2$$

За допомогою САПР за рівняннями зберемо схему та виконаємо адресацію універсальних входів/виходів (рис. 8.27) відповідно до схеми підключень (рис. 8.6).

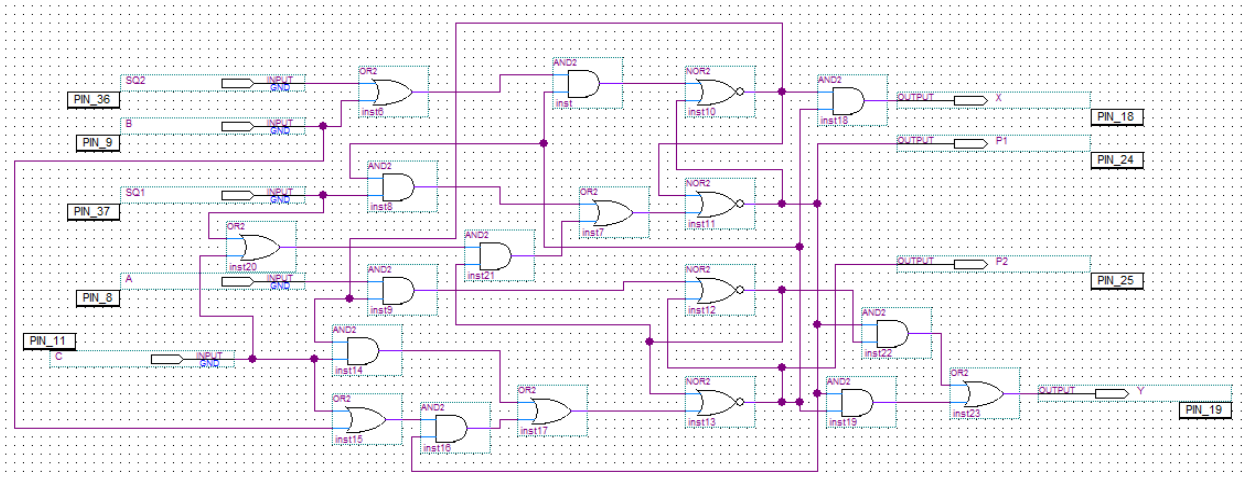


Рисунок 8.27 – Схема зібрана за рівняннями отриманими вище

Далі проведемо часову симуляцію, змодельовавши нормальну роботу схеми.

Задавши вектор часових діаграм отримаємо:

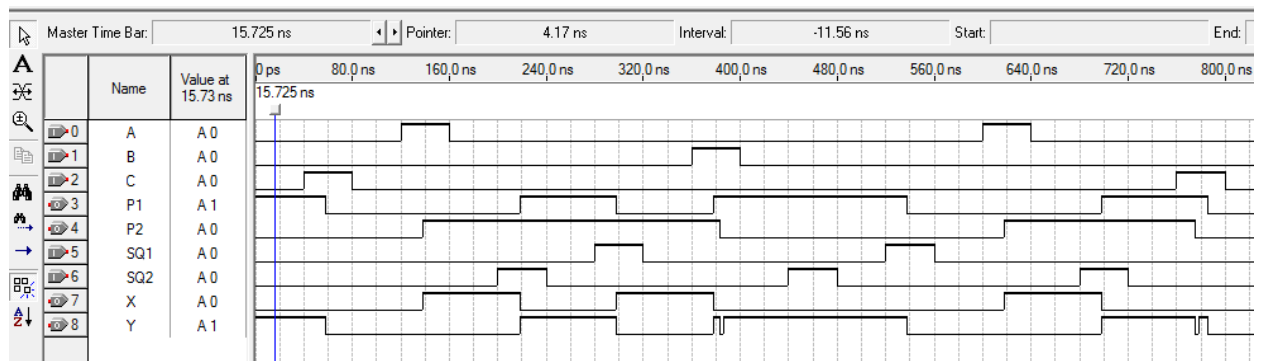


Рисунок 8.28 – Часова симуляція роботи схеми

Приклад 2. Виконати синтез методом циклограм за умовою роботи схеми заданими у вигляді циклограми, що зображена нижче.

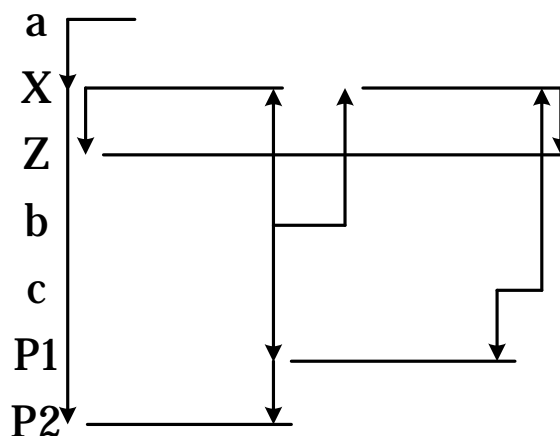


Рисунок 8.29 – Циклограма, що описує роботу схеми

Для вирішення циклограми необхідно ввести дві проміжні змінні. P1 для забезпечення умови не спрацювання на всьому вмикаючому періоді, та P2 для запобігання впливу самотронування першого періоду роботи X на другий.

Рівняння для проміжних змінних отримані з циклограми:

$$P_1 = (b + p_1)\bar{c}$$

$$P_2 = (a + p_2)\overline{(p_1 + c)}$$

Рівняння для вихідних змінних отримані з циклограми:

$$X_1 = (a + p_2)\bar{b}$$

$$X_2 = \bar{b}(c + p_1)$$

$$X = X_1 + X_2 = \bar{b}(a + p_2 + c + p_1)$$

$$Z = (x + z)(x + b)$$

За допомогою САПР за рівняннями зберемо схему та виконаємо адресацію універсальних входів/виходів (рис. 8.30) відповідно до схеми підключень (рис. 8.6).

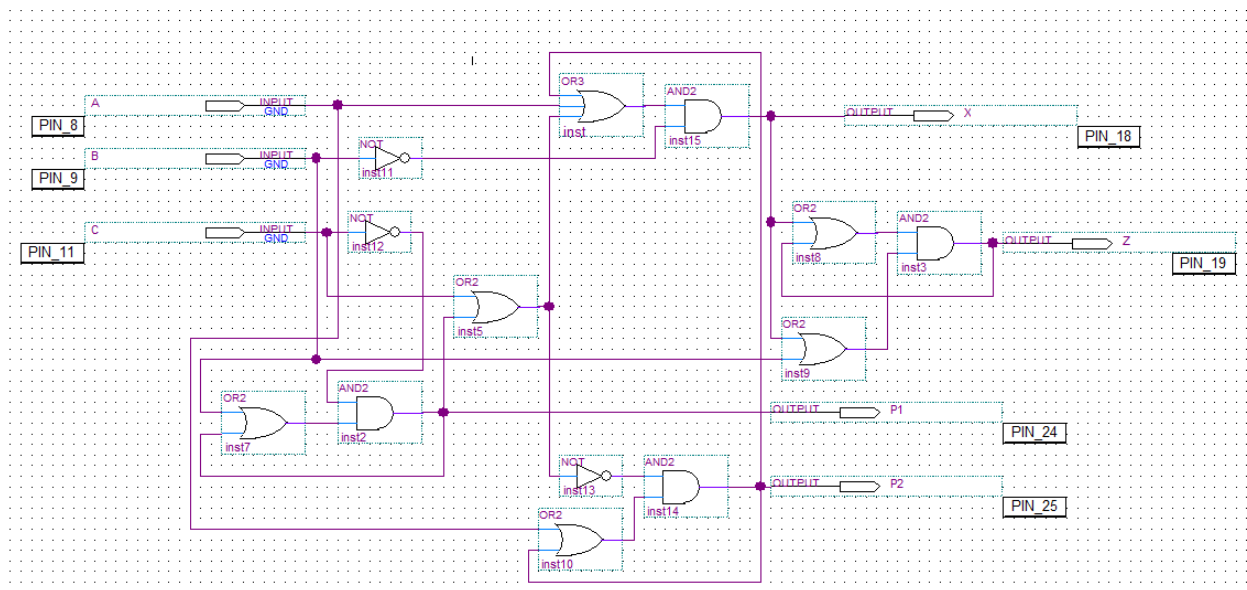


Рисунок 8.30 – Схема зібрана за рівняннями отриманими вище

Далі проведемо часову симуляцію, змодельовавши нормальну роботу схеми.

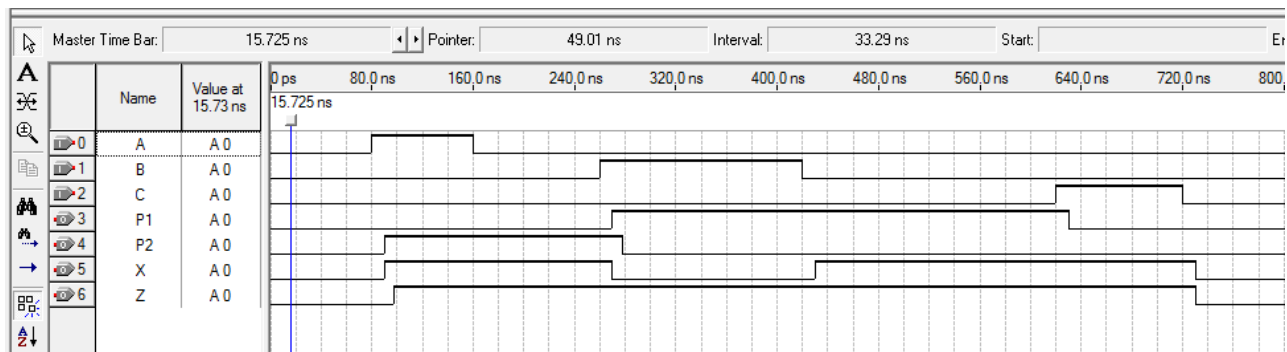


Рисунок 8.31 – Часова симуляція роботи схеми

Приклад 3. Виконати синтез схеми, що генерує послідовність двійкових чисел: 0,3,5,6,6,5,3,0,3...при надходженні на вхід послідовності прямокутних імпульсів.

Хоча найбільше число в послідовності три розрядне і в такому випадку достатньо було б трьох тригерів, числа в заданій послідовності повторюються, тому забезпечення однозначності роботи схеми необхідно взяти ще один тригер, так щоб для одного з чисел послідовності, що повторюються цей тригер набував значення 0, а для іншого – 1 (С – синхронізуючий імпульс).

Побудуємо таблицю для вихідних станів і станів переходу.

	P1	P2	P3	P4	C			
0	0	0	0	0	0	1	1	1
3	0	1	1	1	1	0	1	1
5	1	0	1	1	1	1	0	1
6	1	1	0	1	1	1	0	0
6	1	1	0	0	1	0	1	0
5	1	0	1	0	0	1	1	0
3	0	1	1	0	0	0	0	0

Рисунок 8.32 – Таблиця для вихідних станів і станів переходу з урахуванням додаткового тригера

Побудуємо карти Карно для вхідних сигналів J_i K тригерів P1, P2, P3, P4 та C=1.

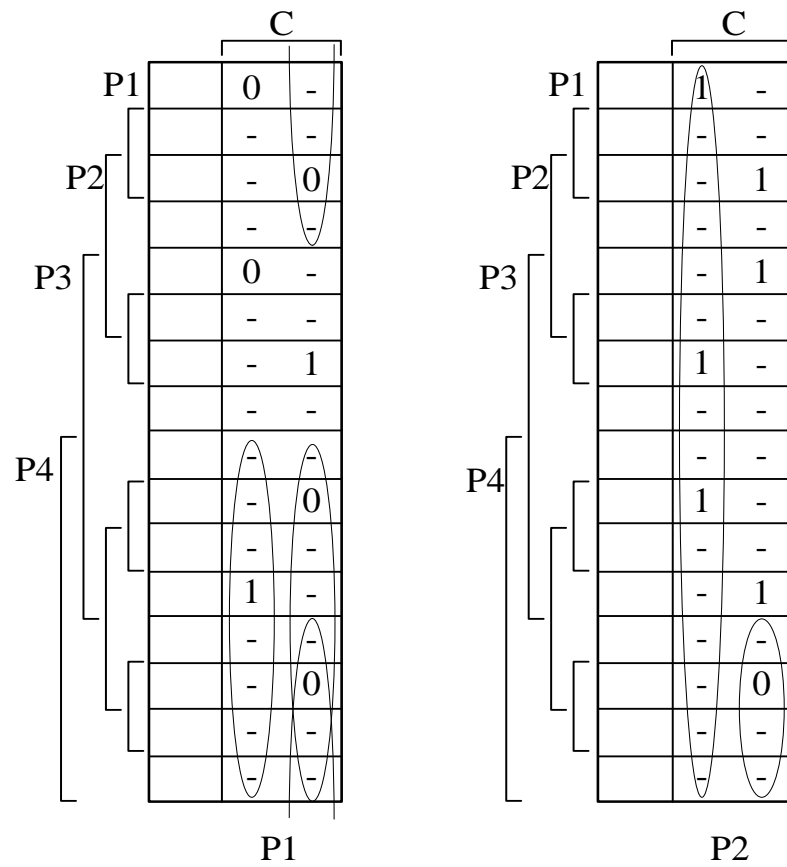


Рисунок 8.33 – Карти Карно для першого і другого тригерів

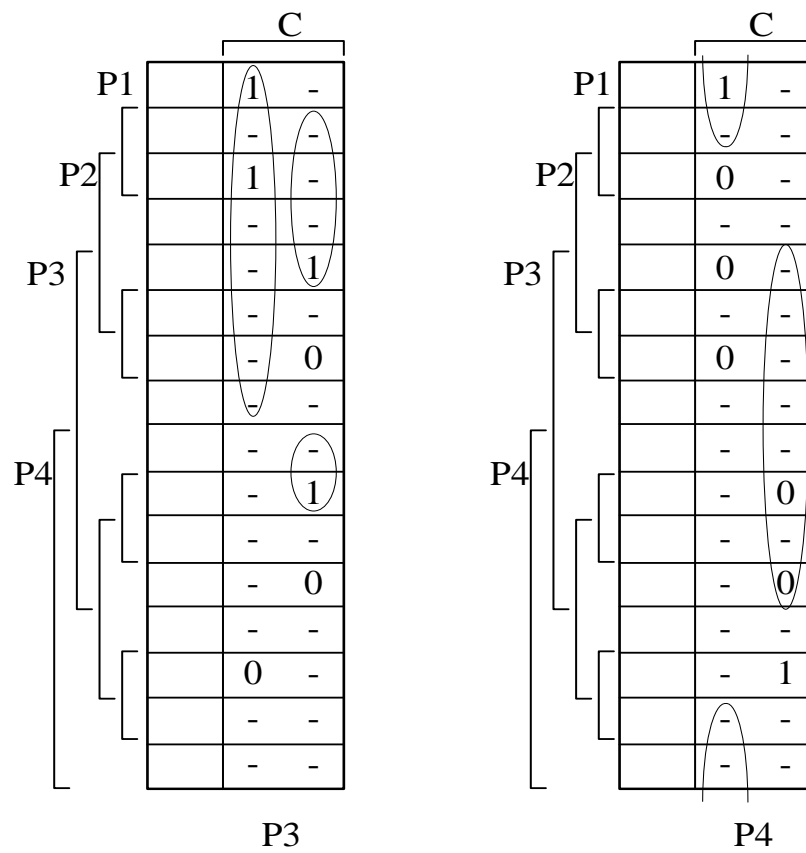


Рисунок 8.34– Карти Карно для третього і четвертого тригерів

Рівняння для встановлення тригерів:

$$J_{p1} = p_4;$$

$$J_{p2} = 1;$$

$$J_{p3} = \bar{p}_4;$$

$$J_{p4} = \bar{p}_2 \bar{p}_3;$$

Рівняння для встановлення тригерів:

$$K_{p1} = p_3 \bar{p}_4;$$

$$K_{p2} = p_3 + \bar{p}_4;$$

$$K_{p1} = p_2 \bar{p}_4 + p_1 p_3 p_4;$$

$$K_{p4} = \bar{p}_3;$$

За допомогою САПР за рівняннями зберемо схему та виконаємо адресацію універсальних входів/виходів (рис. 8.36) відповідно до схеми підключень (рис. 8.6).

Далі проведемо часову симуляцію, змодельювавши нормальну роботу схеми.

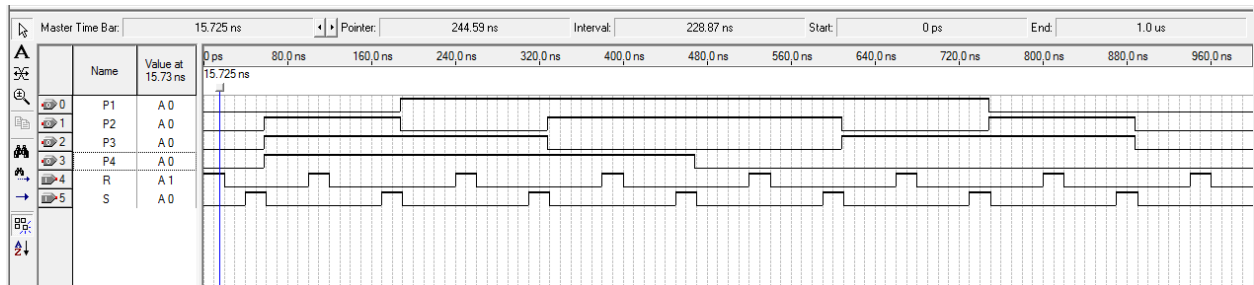


Рисунок 8.35 – Часова симуляція роботи схеми

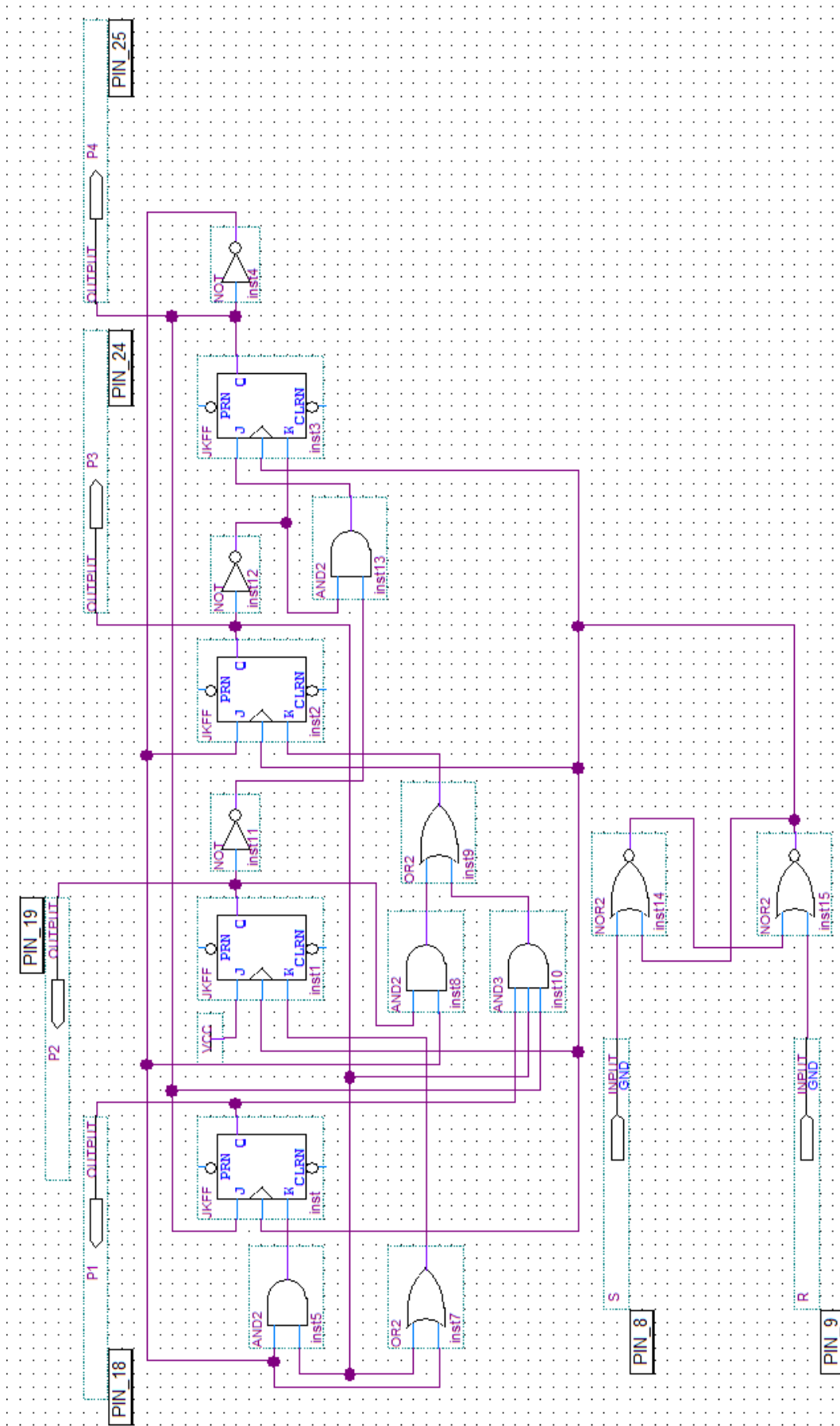


Рисунок 8.36 – Схема зібрана за рівняннями отриманими вище

8.6 Зміст звіту

Звіт з виконання лабораторної роботи робить кожен студент окремо за своїм варіантом. Він повинен містити наступні складові:

- 1) титульний аркуш із зазначенням назви дисципліни та лабораторної роботи (див. Додаток А);
- 2) мету роботи, програму роботи та умови роботи схем, відповідно до свого варіанту;
- 3) логічний синтез схем;
- 4) логічні вирази для вихідних та проміжних змінних, що отримані в результаті синтезу;
- 5) схема на мові Block Diagram у середовищі QUARTUS II з підписами викладача;
- 6) вибір виводів ПЛІС для вхідних та вихідних змінних;
- 7) часова симуляція роботи схеми;
- 8) Висновки по лабораторній роботі.

8.7 Контрольні запитання

1. Як виконується синтез схем на тригерах?
2. Як складаються рівняння для вихідних змінних на основі графопереходу?
3. Як визначається кількість тригерів та мультиплексорів при синтезі багатотактних схем?
4. Наведіть процедуру синтезу лічильників на основі JK-
5. Яким чином створюється проєкт у середовищі Quartus II?
6. Яким чином виконується симуляція методом часових діаграм у середовищі Quartus II?
7. Наведіть процедуру програмування мікросхеми ПЛІС сімейства Cyclone II.

Лабораторна робота № 9

ДОСЛІДЖЕННЯ РОБОТИ ІНТЕГРАЛЬНОГО ТАЙМЕРУ NE555

Тривалість лабораторної роботи – 4 год.

Тривалість домашньої роботи – 4 год.

Мета роботи – здобути вміння розробляти та налаштовувати схеми з використанням мікросхеми інтегрального таймера NE555, навчатися розробляти схеми генератора прямокутних імпульсів та схеми з часовими затримками, практично перевірити працездатність розроблених схем за допомогою інтегральних мікросхем, монтажної плати, кнопок, світлодіодів та резисторів.

9.1. Основні теоретичні відомості

Інтегральний таймер

Промисловістю серійно випускається таймер NE555. Функціональну схему таймера наведено на рисунку 9.1

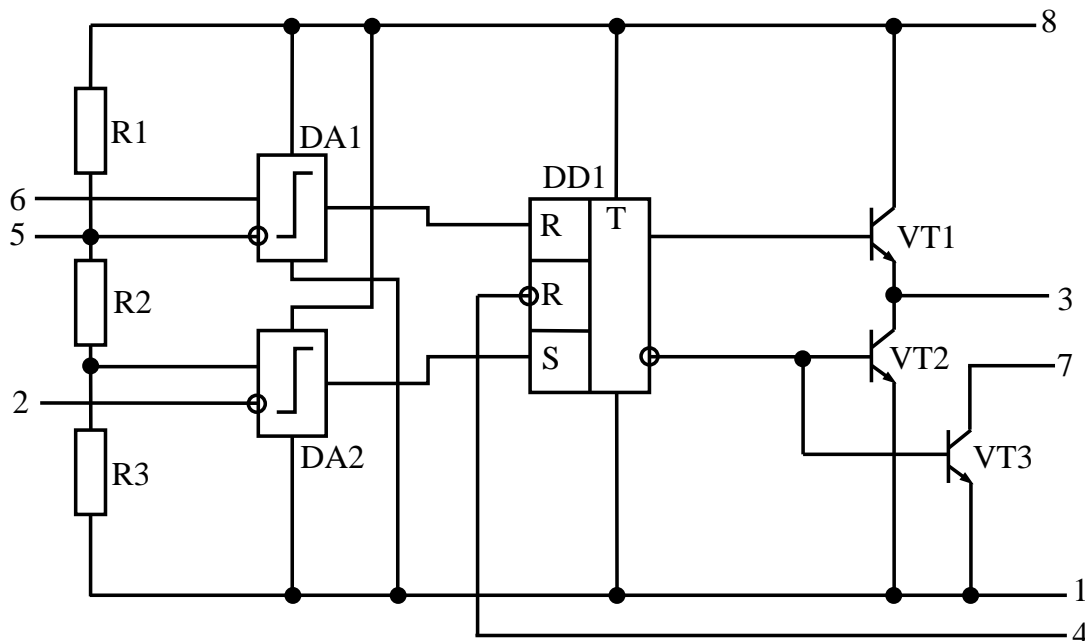


Рисунок 9.1 – Функціональна схема таймера NE555

Таймер містить два аналогових компаратори DA1 і DA2. Вихідний сигнал компаратора дорівнює 1, якщо потенціал на прямому вході вищий, ніж на інверсному. На вході мікросхеми є ланцюжок з трьох резисторів $R_1=R_2=R_3=5$ кОм. Цей ланцюжок (дільник напруги) фіксує пороги спрацьовування компараторів. Оскільки виконується умова $R_1=R_2=R_3$, то пороги спрацьовування компараторів верхнього DA1 і нижнього DA2 рівнів визначаються виразами $2u_{жс}/3$ та $u_{жс}/3$.

Виходи компараторів керують станом асинхронного RS-тригера DD1, який формує керуючі напруги на вході двотактного підсилювача потужності на транзисторах VT1 і VT2. Крім цього, RS-тригер має додатковий інверсний асинхронний вхід скидання. Сигнал з інверсного виходу тригера використовується для керування розрядним транзистором VT3.

Призначення виводів мікросхеми:

- 1 – вивід, який підключається до «нуля» джерела живлення.
- 2 – запуск. Це інверсний вхід компаратора DA2. При подачі на цей вхід імпульсу низького рівня (не більше, ніж $u_{жс}/3$) таймер запускається і на виході (вивід 3) встановлюється сигнал 1 на час, який визначається величиною зовнішнього резистора R і ємністю зовнішнього конденсатора C. Вхідний імпульс може бути як прямокутним, так і синусоїдальним, головне, щоб за тривалістю він був коротший, ніж час заряду конденсатора C. Якщо вхідний імпульс за тривалістю перевищить цей час, то на виході мікросхеми буде залишатися сигнал 1, доти, поки на вході знов не встановиться високий рівень сигналу.
- 3 – вихід. Рівень вихідної напруги (сигнал 1) залежить від напруги живлення і становить $u_{ce}-1,7$ В. При вихідному сигналі 0 вихідна напруга становить приблизно 0,25 В при $u_{ce}=5$ В.
- 4 – скидання. При подачі на цей вивід сигналу 0 (не більше 0.7 В) на виході встановлюється сигнал 0 незалежно від того, в якому режимі перебуває таймер.

5 – контроль. Цей вивід дозволяє отримати доступ до опорної напруги компаратора DA1, яка становить $2u_{жс}/3$. Звичайно цей вивід не використовується. Його рекомендується підключати до «нуля» джерела живлення через конденсатор. 0.01 мкФ для зменшення рівня перешкод.

6 – зупинка. Цей вивід є одним з входів компаратора DA1. Він використовується для зупинки таймера, тобто встановлення на виході сигналу 0. При подачі на цей вивід імпульсу високого рівня (не менше $2u_{жс}/3$) таймер зупиняється і на виході встановлюється сигнал 0. На цей вивід (як і на вивід 2) можна подавати як прямокутні так і синусоїдальні імпульси.

7 – розряд. Цей вивід приєднаний до колектора транзистора VT3, емітер якого заземлений. Через цей транзистор розряджається зовнішній конденсатор С, коли транзистор відкритий, і залишається в розрядженому стані доти, поки транзистор не закриється. Транзистор відкритий, коли на виході мікросхеми сигнал 0, і закритий, коли на виході сигнал 1.

8 – плюс джерела живлення. Напруга живлення таймера може знаходитись в межах 4,5-16 В.

Схему включення мікросхеми, яка виробляє сигнал із заданою тривалістю, подано на рисунку 9.2.

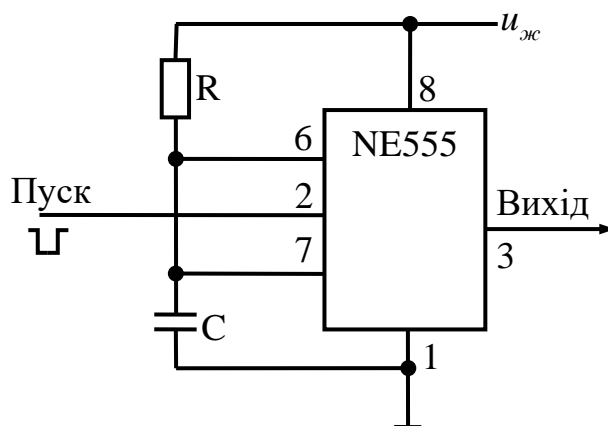


Рисунок 9.2 – Підключення мікросхеми NE555 для отримання сигналу із заданою тривалістю

У вихідному стані на вході 2 сигнал 1 ($>u_{жс}/3$), тригер DD1 скинуто, на його інверсному виході сигнал 1, тому транзистори VT2 і VT3 відкриті і

вихідний сигнал дорівнює 0 (рис. 4.40). Конденсатор С розряджений через відкритий транзистор VT3. При подачі на вхід нульового імпульсу спрацьовує компаратор DA2 і його вихідний сигнал встановлює тригер DD1 в стан 1. Сигнал на інверсному виході тригера стає рівним 0 і транзистори VT2 і VT3 закриваються. На виході встановлюється сигнал 1, а конденсатор С починає заряджатися від джерела живлення через резистор R. Увесь час, поки конденсатор заряджається на виході зберігається сигнал 1. Коли конденсатор заряджається до рівня $2/3 u_{жс}$, спрацьовує компаратор DA1 і скидає тригер DD1. На інверсному виході тригера з'являється сигнал 1, транзистори VT2 і VT3 відкриваються, сигнал на виході стає рівним 0, а конденсатор С розряджається через відкритий транзистор VT3. Схема повертається у вихідний стан.

Часову діаграму імпульсів, що ілюструє роботу схеми, подано на рисунку 9.3.

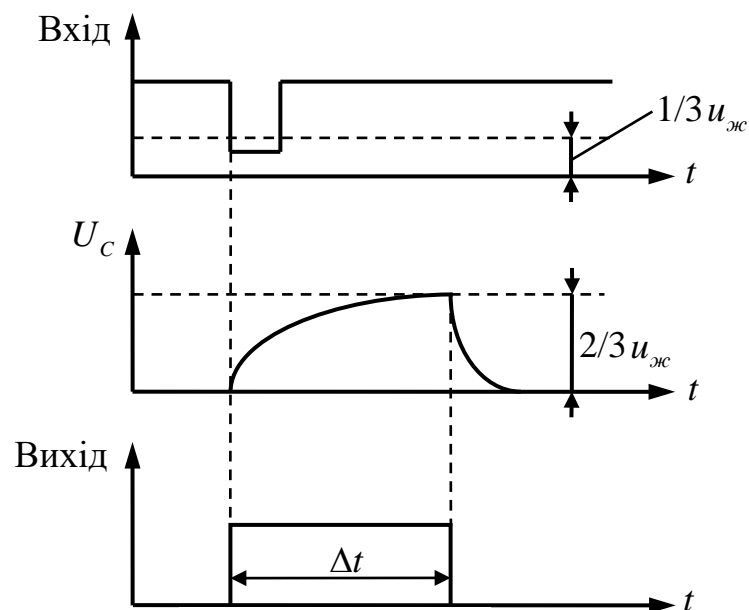


Рисунок 9.3 – Діаграма імпульсів, що ілюструє роботу схеми на рис. 9.2

Час затримки Δt визначається за формулою

$$\Delta t = 1.1RC .$$

Мінімальні значення для R становлять 10 КОм, а для С – 95 нФ. Ці значення встановлені практично із досвіду застосування мікросхеми NE555.

Максимальне значення резистора становить приблизно 15 МОм. Щодо максимальної ємності конденсатора, то справа не в самій ємності, а в струмі

витіку. Чим більша ємність конденсатора, тим більший струм витіку і гірша точність таймера. Тому, якщо таймер буде використовуватись для великих часових інтервалів (десятки та сотні секунд), то рекомендується використовувати конденсатори з малими струмами витіку, наприклад, танталові.

Описана схема виробляє імпульс заданої тривалості, а таймери в схемах автоматики і програмно-реалізовані таймери логічних програмованих контролерів затримують сигнал 1 на заданий час. Вихідний сигнал таймера зникає одночасно із зникненням вхідного сигналу, тобто таймер працює згідно з діаграмою сигналів на рис. 9.4.

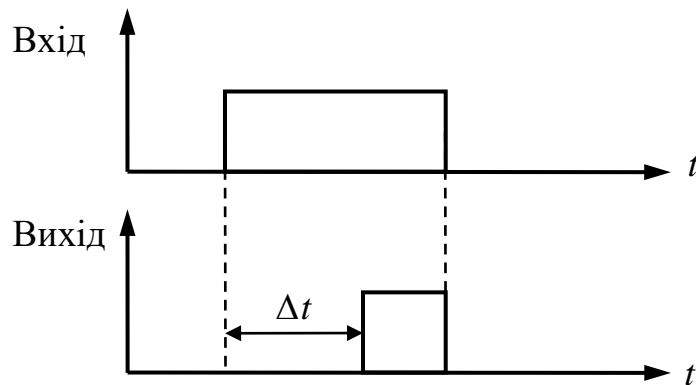


Рисунок 9.4 – Діаграма сигналів на вході та виході таймера

Щоб таймер працював згідно наведеної діаграми, можна застосувати схему на рисунку 9.5.

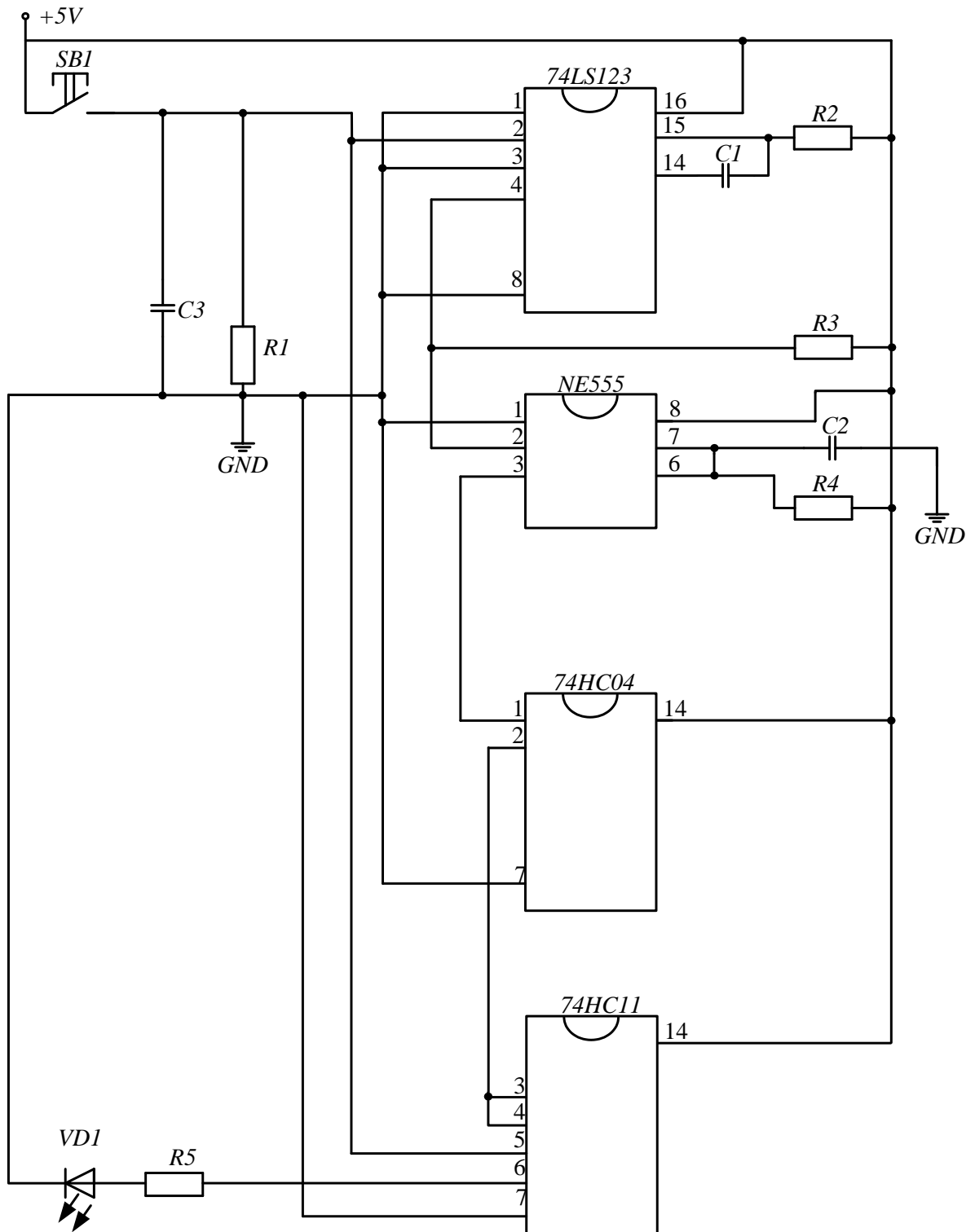


Рисунок 9.5 – Схема таймера, який дає затримку вхідного сигналу 1

R1	15 кОм	R5	800-900 Ом
R2	>15 кОм	C1	>47 мкФ
R3	2.7 кОм	C2	>47 мкФ
R4	>100 кОм	C3	20-100 нФ

На основі мікросхеми NE555 можна побудувати *генератор прямокутних імпульсів*. Підключення зовнішніх елементів до мікросхеми показано на рис. 9.6.

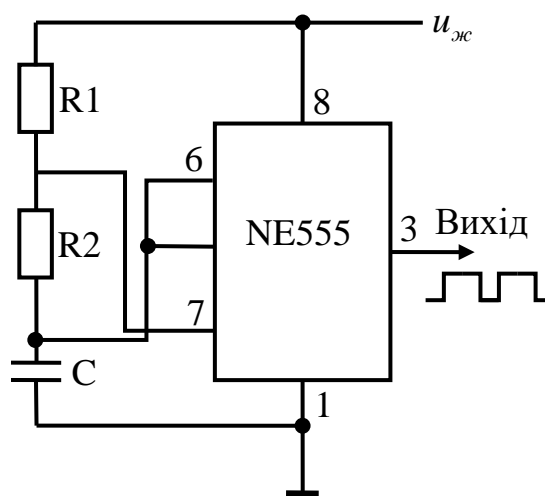


Рисунок 9.6 – Схема генератора прямокутних імпульсів

В цій схемі входи обох компараторів об'єднані та підключені до з'єднання резистора R2 і конденсатора C. Вивід 7 підключено між резисторами R1 і R2.

У вихідному стані конденсатор розряджений і на прямому вході компаратора DA1 (рис. 9.1) та інверсному компаратора DA2 буде рівень напруги, близький до нуля. На прямому вході компаратора DA2 напруга дорівнює $1/3 u_{жс}$, тому на виході цього компаратора сигнал 1. Цей сигнал встановлює тригер DD1 у стан 1. Напруга на базі транзистора VT2 близька до нуля, він закритий, тому сигнал на виході мікросхеми дорівнює 1. Транзистор VT3 також закритий, тому при підключенні напруги живлення $u_{жс}$ конденсатор C починає заряджатись через резистори R1 і R2. Коли напруга на конденсаторі зросте до $2/3 u_{жс}$ спрацює компаратор DA1, який перемикає тригер у стан 0. Сигнал з інверсного виходу тригера відкриває транзистори VT2 і VT3. Вихідна напруга стає близькою до нуля, а конденсатор починає розряджатись через відкритий транзистор VT3 та резистор R2. Як тільки напруга на конденсаторі знизиться до $1/3 u_{жс}$, компаратор DA2 знов перемикає тригер і на виході

мікросхеми знов з'явиться сигнал 0, конденсатор знов починає заряджатись і т.д.

Діаграму сигналів, що пояснюють роботу схеми, подано на рис. 4.46, де u_C – напруга на конденсаторі, $u_{\text{вих}}$ – вихідна напруга.

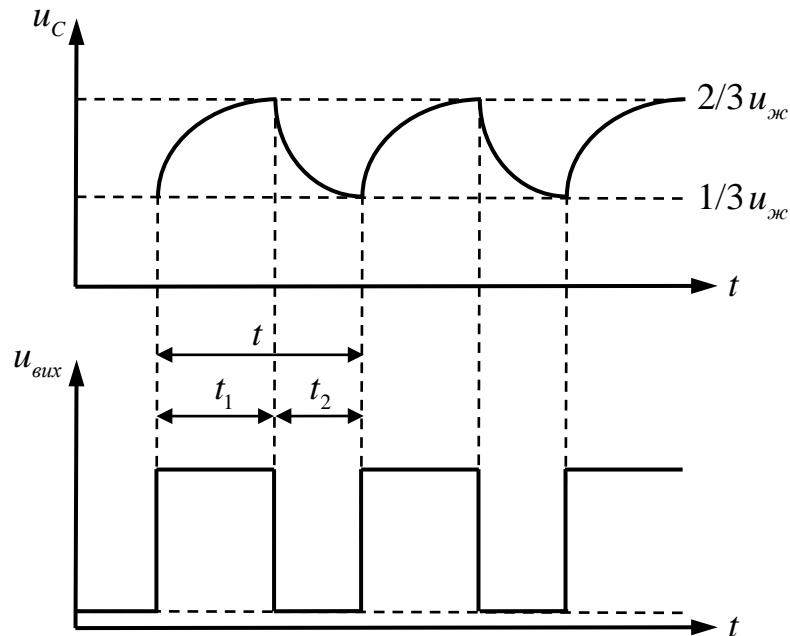


Рисунок 9.7 – Діаграма імпульсів, що пояснює принцип дії генератора прямокутних імпульсів

Частота імпульсів визначається за формулою

$$f = \frac{1}{t} = \frac{1}{0.693 C (R_1 + 2R_2)},$$

тривалість імпульсу

$$t_1 = 0.693 (R_1 + R_2) C,$$

тривалість паузи

$$t_2 = 0.693 R_2 \cdot C.$$

Увага! Реалізація таймерів та генератора імпульсів в середовищі Multisim показана в відео-уроці до Лекції №12 за посиланням <https://youtu.be/Ms0yx8WJRhk>.

9.2 Програма роботи

Перед початком лабораторної роботи кожному студенту необхідно вибрати один із варіантів відповідно до номеру у загальному списку групи. Для кожного з трьох завдань необхідно:

- 7) виконати логічний синтез схеми керування та провести попередні розрахунки за необхідністю (**виконується вдома під час СРС**);
- 8) за отриманими алгебраїчними виразами скласти принципову схему у програмному середовищі MultiSim та виконати симуляцію її роботи (**виконується безпосередньо під час лабораторних робіт**);
- 9) оформити звіт по лабораторній роботі та зробити висновки.

9.3 Завдання до лабораторної роботи

Завдання 1. Виконати синтез системи керування двигунами М1, М2, М3. Затримку часу реалізувати використовуючи схему інтегрального таймера, що показана на рис. 9.2 та дослідити роботу схеми. Підібрати варіанти опорів та конденсаторів для реалізації необхідних затримок часу. Варіанти 1-10 затримки часу відповідно до умов роботи схем, варіанти 11-20 – затримки часу вдвічі більші ніж за умовами роботи схеми.

1. Керування трьома двигунами М1, М2, М3 здійснюється за допомогою кнопок «Пуск» і «Стоп». При натисненні кнопки «Пуск» вмикається із затримкою часу 2 с вмикається двигун М1, потім через 4 с вмикається двигун М2 та через 5 с – двигун М3. При натисненні кнопки «Стоп» двигуни М1, М2, М3 вимикаються без затримки часу.

2. Керування трьома двигунами М1, М2, М3 здійснюється за допомогою кнопок «Пуск» і «Стоп». При натисненні кнопки «Пуск» без затримок часу вмикаються двигуни М1 та М2, потім через 3 с вмикається

двигун М3. При натисненні кнопки «Стоп» двигуни М1 та М2 вимикаються через 2 с, потім через 4 с – двигун М3.

3. Керування трьома двигунами М1, М2, М3 здійснюється за допомогою кнопок «Пуск» і «Стоп». При натисненні кнопки «Пуск» без затримок часу вмикаються двигуни М1 та М3, потім через 1 с вмикається двигун М2. При натисненні кнопки «Стоп» двигуни М2 та М3 вимикаються через 4 с, потім через 2,5 с – двигун М1.

4. Керування трьома двигунами М1, М2, М3 здійснюється за допомогою кнопок «Пуск» і «Стоп». При натисненні кнопки «Пуск» без затримок часу вмикаються двигуни М1, М2, М3. При натисненні кнопки «Стоп» двигун М1 вимикається через 1 с, через 3 с вимикається двигун М3, потім через 4 с – двигун М2.

5. Керування трьома двигунами М1, М2, М3 здійснюється за допомогою кнопок «Пуск» і «Стоп». При натисненні кнопки «Пуск» без затримок часу вмикаються двигуни М1, М2, М3. При натисненні кнопки «Стоп» двигун М3 вимикається через 3 с, через 2 с вимикається двигун М2, потім через 1 с – двигун М1.

6. Керування трьома двигунами М1, М2, М3 здійснюється за допомогою кнопок «Пуск» і «Стоп». При натисненні кнопки «Пуск» вмикається із затримкою часу 2,5 с вмикається двигун М3, потім через 3 с вмикається двигун М1 та через 5 с – двигун М2. При натисненні кнопки «Стоп» двигуни М1, М2, М3 вимикаються без затримки часу.

7. Керування трьома двигунами М1, М2, М3 здійснюється за допомогою кнопок «Пуск» і «Стоп». При натисненні кнопки «Пуск» вмикається із затримкою часу 4 с вмикаються двигун М2 та М3, потім через 2 с вмикається двигун М1. При натисненні кнопки «Стоп» двигуни М1, М2, вимикаються без затримки часу, через 3,5 с вимикається двигун М3.

8. Керування трьома двигунами М1, М2, М3 здійснюється за допомогою кнопок «Пуск» і «Стоп». При натисненні кнопки «Пуск» без затримок часу вмикаються двигуни М1 та М3, потім через 1 с вмикається

двигун М2. При натисненні кнопки «Стоп» двигуни М2 та М3 вимикаються через 4 с, потім через 2,5 с – двигун М1.

9. Керування трьома двигунами М1, М2, М3 здійснюється за допомогою кнопок «Пуск» і «Стоп». При натисненні кнопки «Пуск» без затримок часу вмикаються двигуни М1, М2, М3. При натисненні кнопки «Стоп» двигун М2 вимикається через 4,5 с, через 1,5 с вимикається двигун М1, потім через 2 с – двигун М3.

10. Керування трьома двигунами М1, М2, М3 здійснюється за допомогою кнопок «Пуск» і «Стоп». При натисненні кнопки «Пуск» вмикається із затримкою часу 3,5 с вмикається двигун М3, потім через 1 с вмикається двигун М1 та через 2,5 с – двигун М1. При натисненні кнопки «Стоп» двигуни М1, М2, М3 вимикаються без затримки часу.

Завдання 2. Скласти схему генератора тактових імпульсів в середовищі Multisim, на базі мікросхеми інтегрального таймера, що подана на рис. 9.6. Навести три варіанти роботи генератора при різних значеннях опорів резисторів та ємностей конденсаторів.

Завдання 3. Скласти схему таймера, що робить затримку вхідного сигналу. Схему представлено на рис. 9.5. Вхідним сигналом вважати вихідний сигнал рівняння відповідно до варіанту у загальному списку групи. Час затримки взяти у секундах відповідно до номеру бригади, помноженого на 3.

$$1. f = abc + a\bar{c} + ab$$

$$2. f = a + \bar{b}c \cdot (a + b)$$

$$3. f = \overline{a\bar{b}} + \overline{(ac + abc)}$$

$$4. f = (a + \bar{b} + c) \cdot (b + \bar{c})$$

$$5. f = \overline{(\bar{a} + \bar{b} + c)} \cdot (b + \bar{c})$$

$$6. f = (a + c) \cdot (\bar{b} + a) \cdot (b + \bar{c})$$

$$7. f = \overline{abc} + (ab + c) \cdot cb$$

$$8. f = bc + (\bar{a} + b) \cdot (ab + \bar{c})$$

$$9. f = (a\bar{c} + bc) \cdot (\bar{c} + \bar{b})$$

$$10. f = (b + c) \cdot (abc + a\bar{b})$$

$$11. f = (\bar{a} + c) \cdot (b + \bar{a}) \cdot (b + \bar{c})$$

$$12. f = bc \cdot (\bar{a} + b) + \overline{(ab + c)}$$

$$13. f = (\bar{a} + \bar{c} + bc) \cdot (\bar{c} + \bar{b})$$

$$14. f = \bar{a}b \cdot \overline{(ac + \bar{a}bc)}$$

$$15. f = \bar{b}c \cdot (\bar{a} + b) \cdot \overline{(ab + c)}$$

$$16. f = (\bar{a} + b) \cdot \overline{(ac + (\bar{a} + \bar{b}) \cdot c)}$$

$$17. f = abc \cdot \overline{(a\bar{c} + ab)}$$

$$18. f = (b + c) \cdot \overline{((\bar{a} + b) + (ab + \bar{c}))}$$

$$19. f = \overline{(\bar{a} + c) \cdot (b + \bar{a}) \cdot (b + \bar{c})}$$

$$20. f = (ab + bc) \cdot \overline{(a + bc)}$$

9.4 Зміст звіту

- 1) Титульний аркуш.
- 2) Мета роботи, програма роботи.
- 3) Умова роботи схем згідно варіанту.
- 4) Логічний синтез схеми керування та розрахунки елементів.
- 5) Схеми до завдань 1-3 та результати їх роботи у програмному середовищі MultiSim.
- 6) Висновок про відповідність роботи заданим умовам.

9.5. Контрольні питання

1. Які основні компоненти входять до схеми таймера NE555?
2. Поясніть призначення виводів мікросхеми NE555.
3. Наведіть часову діаграму імпульсів, що ілюструє роботу схеми таймера NE555.
4. Які мінімальні значення для R та C мікросхеми NE555?
5. Як визначити частота імпульсів, тривалість імпульсу та тривалість паузи генератора прямокутних імпульсів?

Заходи безпеки під час виконання лабораторних робіт в лабораторії «Автоматизації технологічних процесів, установок і комплексів»

Перед початком лабораторних робіт проводиться інструктаж з правил безпеки згідно з інструкцією з охорони праці, затвердженою розпорядженням декана ФЕА. Після вивчення і перевірки знань цієї інструкції прізвище кожного студента заноситься до контрольного листа, де він ставить свій особистий підпис.

Під час виконання лабораторної роботи студентам забороняється:

- заходити за огороження робочих місць, торкатися неізольованих струмопровідних частин обладнання, а також конструкцій обладнання, які обертаються;
- збирати схеми або робити перемикання в них, усувати недоліки без відключення установки;
- підніматися на фундаменти машин;
- стояти поряд з незахищеними кожухами, муфтами, дисками та частинами машин, що обертаються;
- підключати та знімати з'єднувальні провідники під напругою;
- розбирати схеми не викручуючи клеми, виривати з'єднувальні провідники із клем;
- приєднувати провідники до приладів, машин та апаратів без напаяних наконечників;
- робити переключення на головних (розподільних) щитах;
- вмикати вимикачі для включення напруги до перевірки викладачем правильності з'єднань у схемі;
- виконувати лабораторну роботу одному.

Студенти зобов'язані:

- перед включенням схеми перевірити, в якому положенні знаходиться решта членів бригади (чи торкається хто-небудь до

струмопровідних частин або до частин машин, які можуть обертатися) та попередити голосом: «Обережно, вмикаю!»;

– не залишати лабораторну установку без нагляду, постійно спостерігати за її справною роботою.

Після закінчення роботи необхідно відключити лабораторну установку, розібрати схему, навести порядок на робочому місці, повідомити викладача про всі неполадки (якщо вони є), виявлені під час роботи.

Список рекомендованої літератури

1. Ковальчук О.В. Логічний синтез дискретних схем автоматики: навчальний посібник – К.: НТУУ «КПІ», 2008. – 168 с. ISBN 978-966-622-294-0.
2. Bruno, F. (2021). FPGA Programming for Beginners. Packt Publishing Ltd, Birmingham-Mumbai. ISBN 978-1-78980-541-3.
3. Zeidman, B. (2002). Designing with FPGAs and CPLDs. Elsevier, CMP Books Lawrence, Kansas. ISBN: 1-57820-112-8.
4. Grout, I. (2008). Digital Systems Design with FPGA and CPLD. Elsevier. ISBN-13: 978-0-7506-8397-5.
5. Семенюк В. Я. Класифікація сучасних програмованих логічних інтегральних схем / В. Я. Семенюк, М. В. Воскресенський, О. І. Міскевич. // Науковий журнал "Комп'ютерно-інтегровані технології: освіта, наука, виробництво". – 2013. – №12. – С. 180–183. (доступ за посиланням <https://cutt.ly/OOoJhE3>).
6. Intel DE10-Lite Board. Documentation (доступ за посиланням <https://cutt.ly/WOoZUqH>).
7. Intel DE1-SoC Board. Documentation (доступ за посиланням <https://cutt.ly/JOoC6Pl>).
8. Altera Cyclone II. Device Family Overview (доступ за посиланням <https://cutt.ly/1OoJbm4>).
9. Altera MAX300A. Programmable Logic Device Family (доступ за посиланням <https://cutt.ly/KJUTy6e>).
10. Intel Max 10 GPGA Device Overview (доступ за посиланням <https://cutt.ly/AOoLKxT>).
11. Ковальчук, О. В., Бур'ян, С. О. (2010). Застосування різних методів в синтезу для складних програм для логічних програмованих контролерів. Інформаційний збірник Промелектро. «Промислова електроніка та електротехніка». (4). 51–53 (доступ за посиланням <https://ela.kpi.ua/bitstream/123456789/38235/1/09.pdf>).

ДОДАТОК А

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Лабораторна робота №1

з дисципліни: «Системи автоматизації-1»

на тему:

**«ЛОГІЧНИЙ СИНТЕЗ ТА ДОСЛІДЖЕННЯ СХЕМ АВТОМАТИКИ НА
ДИСКРЕТНИХ ЛОГІЧНИХ ЕЛЕМЕНТАХ»**

Бригада №1, Варіант №1

Виконав:

студент групи ЕП-32

Шевченко Т.Г.

Перевірив:

ас. Землянухіна Г.Ю.