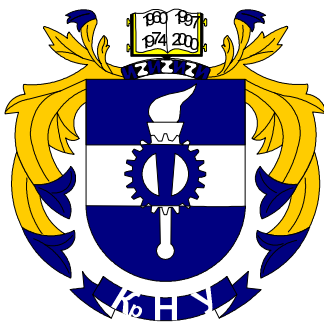


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО
ІНСТИТУТ ЕЛЕКТРОМЕХАНІКИ, ЕНЕРГОЗБЕРЕЖЕННЯ
І СИСТЕМ УПРАВЛІННЯ



МЕТОДИЧНІ ВКАЗІВКИ
ЩОДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ПРОЕКТУВАННЯ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ»
ДЛЯ СТУДЕНТІВ ДЕННОЇ ТА ЗАОЧНОЇ ФОРМ НАВЧАННЯ
ЗІ СПЕЦІАЛЬНОСТІ
141 – «ЕЛЕКТРОЕНЕРГЕТИКА, ЕЛЕКТРОТЕХНІКА ТА
ЕЛЕКТРОМЕХАНІКА»
ЗА ОСВІТНЬО-НАУКОВОЮ ТА ОСВІТНЬО-ПРОФЕСІЙНОЮ
ПРОГРАМАМИ
«ЕЛЕКТРОМЕХАНІЧНЕ ОБЛАДНАННЯ ЕНЕРГОЄМНИХ ВИРОБНИЦТВ»
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

КРЕМЕНЧУК 2018

Методичні вказівки щодо виконання лабораторних робіт з навчальної дисципліни «Проектування електромеханічних систем» для студентів денної та заочної форм навчання зі спеціальності 141 – «Електроенергетика, електротехніка та електромеханіка» за освітньо-науковою та освітньо-професійною програмами «Електромеханічне обладнання енергоємних виробництв» освітнього ступеня «Магістр»

Укладач старш. викл. М. Ю. Юхименко

Рецензент к.т.н., доц. А. П. Калінов

Кафедра систем автоматичного управління та електропривода

Затверджено методичною радою Кременчуцького національного університета імені Михайла Остроградського

Протокол «____» від «____» _____ 2018 р.

Голова методичної ради _____ проф. В. В. Костін

ЗМІСТ

Вступ.....	4
1 Теми та погодинний розклад лабораторних і самостійних робіт ..	6
2 Перелік лабораторних робіт.....	7
Лабораторна робота № 1 Вивчення навчального лабораторного стенда НЛС ПЛІС фірми ALTERA.....	7
Лабораторна робота № 2 Вивчення організації системи автоматизованого проектування MAX+PLUS II.....	13
Лабораторна робота № 3 Створення схеми проекту та симуляція його роботи засобами САПР MAX+PLUS II.....	23
Лабораторна робота № 4 Вивчення опису логічних схем засобами мови програмування AHDL.....	30
Лабораторна робота № 5 Розробка проекту складного цифрового пристрою на базі НЛС ПЛІС	38
3 Критерії оцінювання знань студентів	48
Список літератури.....	49

ВСТУП

Вивчення навчальної дисципліни «Проектування електромеханічних систем» зумовлене метою навчання, що відповідно до кваліфікаційної характеристики спеціальності 141 – «Електроенергетика, електротехніка та електромеханіка», полягає у підготовці фахівців з електромеханіки, здатних самостійно і творчо розв'язувати завдання розробки, використання і експлуатації сучасних електромеханічних систем, що забезпечують ефективне функціонування технологічного обладнання і систем автоматизації промислових установок і технологічних комплексів на їх базі. Здатних виконувати проектування систем керування електроприводами, обґрунтовано вибирати елементну базу і технічні засоби технологічних комплексів; одержання практичних навичок розробки алгоритмів функціонування і оптимізації режимів роботи електромеханічних систем; загальне сприяння закріпленню та поглибленню теоретичних знань з питань системного проектування електромеханічних систем.

Метою навчальної дисципліни «Проектування електромеханічних систем» є поглиблення спеціальної підготовки спеціалістів – електромеханіків, отримання студентами необхідного обсягу теоретичних знань щодо принципів побудови сучасного електромеханічного обладнання, вимог до його складових частин, обґрунтованого вибору принципових технічних рішень під час проектування, тенденцій розвитку сучасних систем автоматизованого електроприводу та його елементної бази.

Навчальна дисципліна «Проектування електромеханічних систем» базується на знанні предметів: «Теорія електроприводу», «Мікроелектронні пристрої електромеханічних систем», «Теорія автоматичного керування» та ін.

У результаті проведення лабораторних робіт з навчальної дисципліни «Проектування електромеханічних систем» студент повинен

знати:

- основні складові процесу проектування електромеханічних систем, декомпозицію процесу проектування;

- зміст етапів проектування автоматизованих електромеханічних систем;
- методи математичного опису складних електромеханічних систем та їх складових;
- правила складання технічного завдання на проектування та оформлення конструкторської документації;
- основні правила вибору оптимальних проектних рішень;
- особливості стандартизації та уніфікації електромеханічних систем та їх складальних одиниць при виконанні вимог технічного завдання;
- склад і призначення технічних засобів автоматизованого проектування;
- перспективи розвитку сучасних електромеханічних систем;

уміти:

- складати і аналізувати моделі динаміки руху промислових електромеханічних систем;
- проводити відбір даних, потрібних для вирішення задач керування електромеханічними системами (у тому числі й задач з неоднозначною постановкою);
- правильно вибирати структуру систем керування електромеханічними системами і алгоритми їх роботи;
- аналізувати залежність вирішення задачі від параметрів, що входять до завдання, і від варіантів її постановки;
- раціонально використовувати сучасну елементу базу силової перетворювальної техніки для реалізації енергетичного електромеханічних систем.

1 ТЕМИ ТА ПОГОДИННИЙ РОЗКЛАД ЛАБОРАТОРНИХ І САМОСТІЙНОЇ РОБОТИ

№ пор.	Тема	Денна форма		Заочна форма	
		К-сть год (лб)	К-сть год СРС	К-сть год (лб)	К-сть год СРС
Змістовий модуль 1					
1	Вивчення навчального лабораторного стенда НЛС ПЛІС фірми ALTERA	2	4	0,5	5
2	Вивчення організації системи автоматизованого проектування MAX+PLUS II	2	4	0,5	5
3	Створення схеми проекту та симуляція його роботи засобами САПР MAX+PLUS II	2	4	–	6
Змістовий модуль 2					
4	Вивчення принципів опису логічних схем за допомогою мови AHDL	2	4	2	6
5	Розробка проекту складного цифрового пристрою на базі НЛС ПЛІС	2	6	2	8
Усього за семестр		10	22	6	30

2 ПЕРЕЛІК ЛАБОРАТОРНИХ РОБІТ

Лабораторна робота № 1

Тема. Вивчення навчального лабораторного стенда НЛС ПЛІС фірми ALTERA

Мета: засвоїти функціональні можливості лабораторного стенда НЛС ПЛІС для практичної реалізації навчальних проектів цифрових пристроїв на базі ПЛІС Altera.

Короткі теоретичні відомості

В даний час фірма ALTERA випускає такі сімейства програмованих інтегральних схем, як MAX 3000, MAX 7000, MAX 9000, FLEX 6000, FLEX 8000, FLEX 10K, APEX 20K, ACEX, Mercury та ряд інших. Основні характеристики найбільш популярних з них наведені в таблиці 1.1.

Таблиця 1.1 Основні технічні характеристики сімейств ПЛІС ALTERA

Характеристики	Сімейства ПЛІС			
	MAX 7000E(S)	MAX 9000	FLEX 8000A	FLEX 10K
Архітектура	матриці І-АБО	матриці І-АБО	таблиці перекодування	таблиці перекодування
Логічна ємність ¹	6000-600-5000	12000	2500-16000	10000-100000
Внутрішня пам'ять ²	немає	немає	немає	6-24 Кбіт
Число користувальницьких виводів	36-164	60-216	68-208	59-406

¹ одиниця виміру – число еквівалентних логічних вентилів (вентилів типу

2 І-НІ);

² при використанні внутрішньої пам'яті доступні користувачеві логічні ресурси НВІС не зменшуються.

Опис лабораторного обладнання

Устрій стенда заснований на мікросхемі сімейства MAX 7000S (тип EPM7032SLC44-5). Лабораторний стенд складається з основного блоку, що включає елементи конфігурування, елементи індикації, нефіксовані кнопкові вимикачі, перемикачі, що задають на виводах мікросхем вхідні логічні рівні «0» та «1», роз'єм розширення для підключення зовнішніх пристроїв, конфігураційного пристрою й стабілізованого джерела живлення напругою 5 В.

Схема розташування елементів лабораторного стенда НЛС ПЛІС наведена на рис. 1.1.

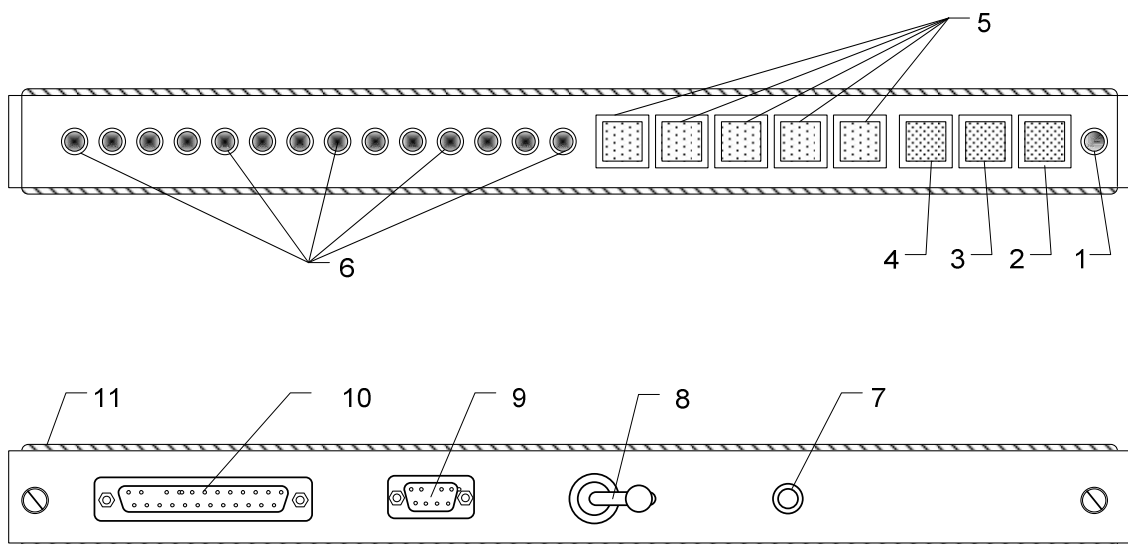


Рисунок 1.1 – Розташування елементів лабораторного стенда НЛС ПЛІС

На рис. 1.1 позначені:

1. Світлодіод індикації наявності напруги живлення.
2. Кнопка «Вперед».
3. Кнопка «Стоп».
4. Кнопка «Назад».
5. Додаткові кнопки управління.
6. Світлодіоди індикації логічних рівнів на виводах ПЛІС.

7. Роз'єм для підключення зовнішнього блока живлення
8. Тумблер відключення індикації.
9. Роз'єм RS-232C для підключення сигнального кабелю.
10. Роз'єм для підключення програматора ByteBlaster (MV).
11. Корпус лабораторного стенда.

На передній бічній панелі розташовані роз'єми X3 і X4 для підключення додаткових зовнішніх пристроїв. На задній бічній панелі розташований роз'єм живлення X0 і роз'єм інтерфейсу JTAG. Роз'єм JTAG (X1) служить для підключення до лабораторного стенда кабелю завантажувального пристрою ByteBlaster. Через роз'єм JTAG на ByteBlaster подається живлення на НЛС ПЛІС.

Лабораторний стенд НЛС ПЛІС складається з трьох функціональних модулів (рис. 1.2): модуля клавіатури та індикації, модуля адаптації ПЛІС, виконавчих елементів (плата буферних регістрів для узгодження логічних рівнів цифрової системи управління з високовольтними колами).

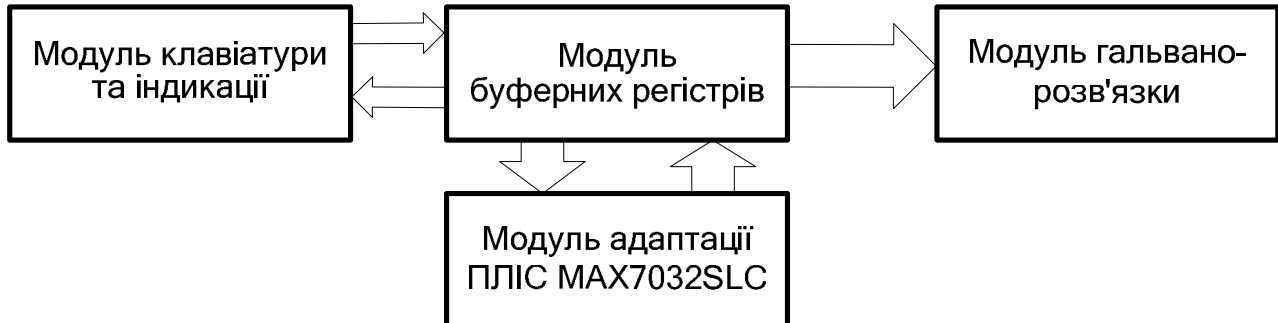


Рисунок 1.2 – Функціональна схема системи управління ЕП на базі ПЛІС Altera

Для реалізації системи управління використовується мікросхема програмувальної логіки сімейства MAX 7000S. Мікросхема EPM7032SLC44-5 являє собою пристрій у пластмасовому корпусі типу J з 44 контактами і розміщується на печатній платі, принципова схема якої зображена на рис. 1.3.

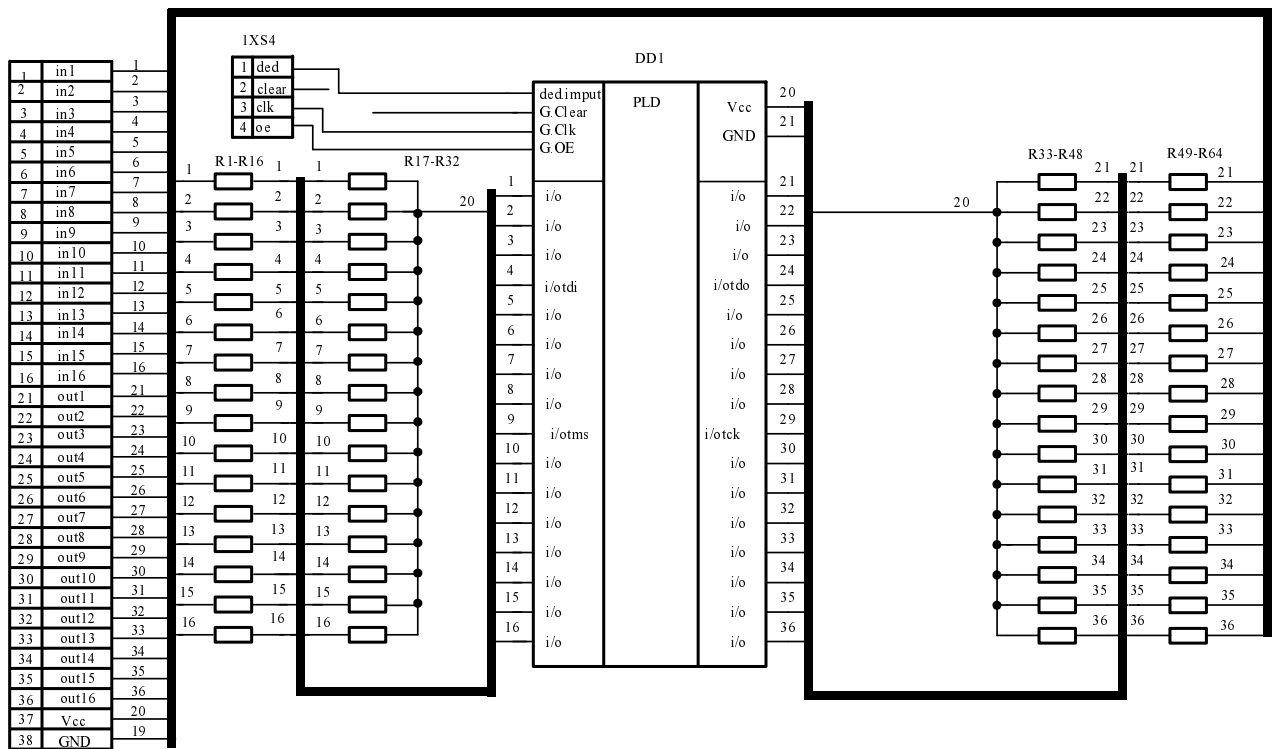


Рисунок 1.3 – Принципова схем плати модуля ПЛІС

Принципова схема блоку буферних регістрів для підключення пристрою керування до блоку узгодження і індикації показана на рис. 1.4. На рис. 1.5 та 1.6 показаний зовнішній вигляд роз'ємів лабораторного стенда для вивчення мікросхем програмованої логіки.

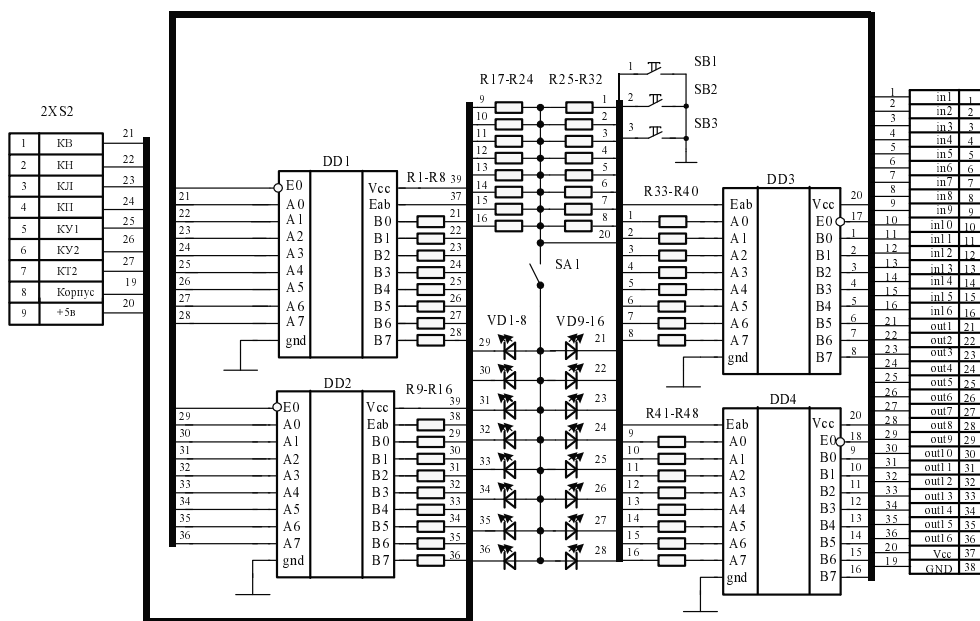


Рисунок 1.4 – Принципова схема блоку буферних регістрів та індикації

Програматор ByteBlaster(MV) (Рис. 1.5) забезпечує наступні режими завантаження конфігурації:

- PS-режим – пасивний послідовний режим (PS – Passive Serial) - використовується для конфігурації мікросхем сімейств APEX 20K, FLEX 10K, FLEX 8000 і FLEX 6000.
- JTAG-режим – використовується для програмування або конфігурування ПЛІС сімейств APEX 20K, FLEX 10K, MAX 9000, MAX 7000S, MAX 7000A, MAX 3000A.

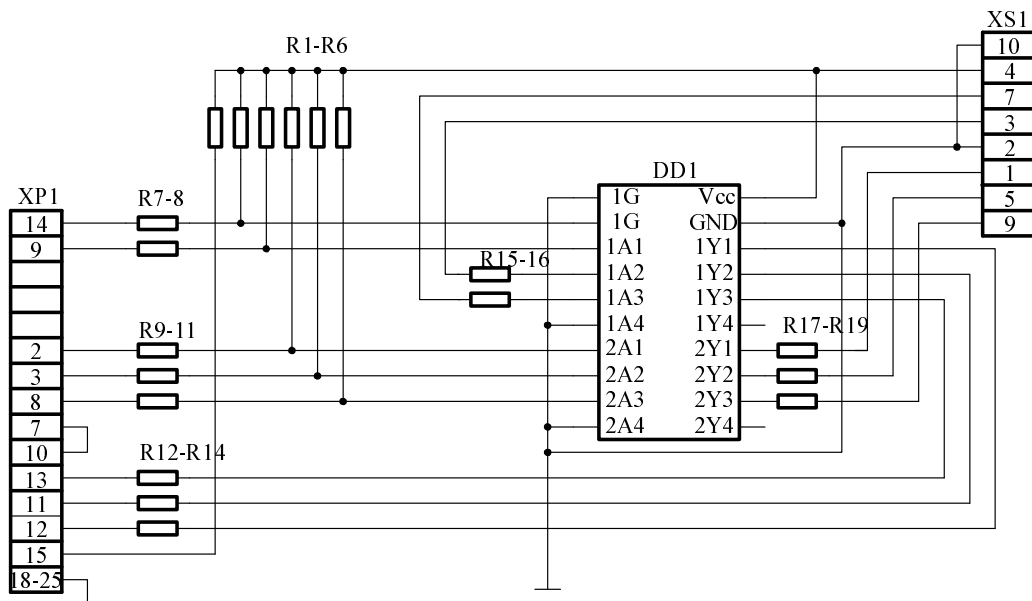


Рисунок 1.5 – Електрична схема пристрою ByteBlaster (MV)

Таблиця 1.3 – Граничні режими експлуатації пристрою ByteBlaster(MV)

Позначення	Параметр	Умови	Min	Max	Одиниці виміру
V_{CC}	Напруга живлення	Відносно землі	-0,5	7,0	Вольт
V_I	Вхідна напруга	Відносно землі	-0,5	7,0	Вольт
V_{CC}	Напруга живлення	5,0	4,5	5,5	Вольт
		3,3	3,0	3,6	Вольт

Порядок виконання роботи

1. Ознайомитися з описом і електричною принциповою схемою лабораторної установки.
2. Вивчити склад органів керування, індикації та інтерфейсних роз'ємів лабораторного стенда НЛС ПЛІС.
3. Ознайомитися із системою внутрішньо-схемного програмування НВІС програмованої логіки фірми Altera.
4. Ознайомитися з логічними схемами систем керування на базі ПЛІС Altera.
5. Використовуючи персональну ЕОМ зі встановленим програмним пакетом ALTERA MAX+PLUS II, забезпечити коректне підключення лабораторного стенда НЛС ПЛІС до ПЕОМ.
6. За допомогою програматора ByteBlaster (MV), приєднаного до паралельного порту ПЕОМ, перевірити працездатність системи внутрішньосхемного програмування НВІС ПЛІС.
7. Досягти коректної роботи лабораторного стенда НЛС ПЛІС і налагодити схему.
8. Оформити звіт.

Зміст звіту

1. Назва та мета лабораторної роботи.
2. Схема лабораторної установки.
3. Короткі відомості щодо призначення елементів лабораторного стенда.
4. Висновки щодо роботи.

Контрольні питання

1. В чому полягає призначення програмованих логічних інтегральних схем?
2. Проаналізуйте основні особливості використання ПЛІС порівняно з системами, побудованими на схемах з «жорсткою» логікою.
3. Які основні характеристики ПЛІС фірми Altera?

4. Опишіть процедуру внутрішньосхемного програмування ВІС програмованої логіки фірми Altera.

5. Охарактеризуйте призначення основних функціональних блоків ПЛІС ПЛІС.

6. Поясніть принципи з'єднання основних елементів лабораторного стенда і підключення його інтерфейсних роз'ємів до портів ПЕОМ.

7. Розкрити особливості програмування НВІС ПЛІС, що вмонтовано в схему.

Література: [1, с. 17 – 46; 2, с. 8 – 33; 3, с. 44 – 56].

Лабораторна робота № 2

Тема. Вивчення системи автоматизованого проектування MAX+PLUS II

Мета: засвоїти навички створення проектів пристроїв керування електромеханічними системами в середовищі MAX+PLUS II фірми ALTERA.

Короткі теоретичні відомості

Система автоматизованого проектування MAX+PLUS II являє собою інтегроване середовище для розробки цифрових пристроїв на базі ПЛІС фірми ALTERA і забезпечує виконання всіх етапів, необхідних для випуску готових виробів:

- створення проектів пристроїв;
- синтез структур і трасування внутрішніх зв'язків ПЛІС;
- підготовку даних для програмування або конфігурування ПЛІС (компіляцію);
- верифікацію проектів (функціональне моделювання й часовий аналіз);
- програмування або конфігурування ПЛІС.

Нижче представлено головне вікно програми (рис. 2.1), воно має стандартний інтерфейс Windows-додатків. У заголовку вікна програми вказується ім'я та шлях до останнього проекту, з яким велася робота.



Рисунок 2.1 – Головне вікно системи MAX+PLUS II

Додатки системи MAX+PLUS II.

До складу пакета MAX+PLUS II входять наступні зв'язані між собою додатки, що реалізують всі етапи розробки цифрових пристроїв на базі ПЛІС фірми ALTERA.

Додатки для створення проектів (редактори проектів):

Graphic Editor – графічний редактор (рис. 2.2), призначений для введення проекту у вигляді схеми з'єднань символів елементів, що вилучаються зі стандартних бібліотек пакета або з бібліотеки користувача.

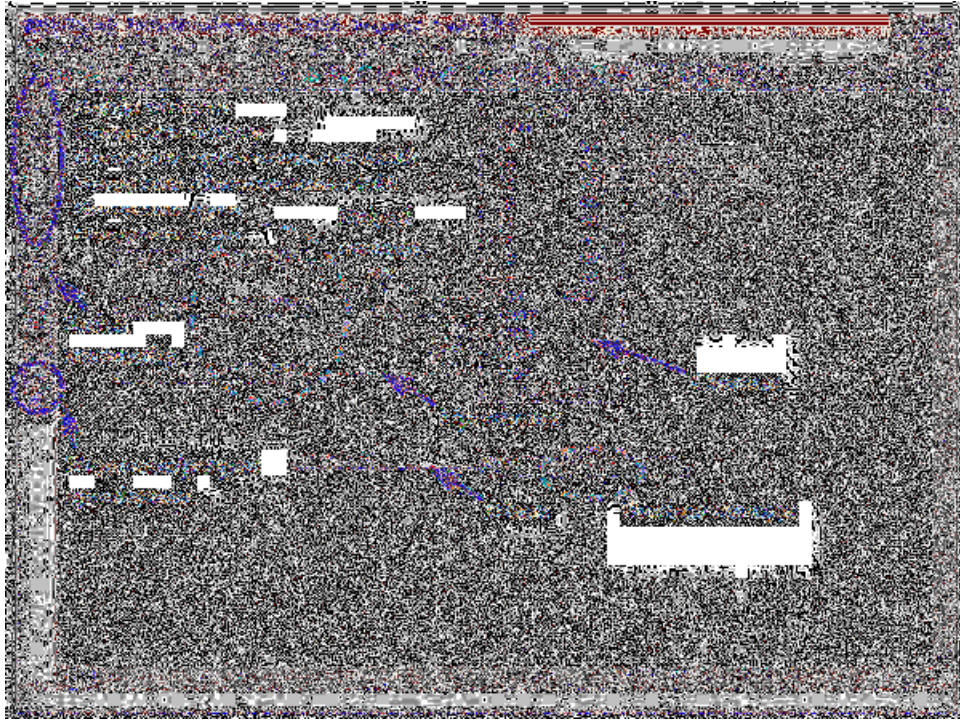


Рисунок 2.2 – Панелі інструментів графічного редактора

Вставлення символу здійснюється подвійним клацанням лівої кнопки мишки на вільному місці вікна графічного редактора. Введені символи й групи символів можна копіювати, видаляти, повертати, пересувати в іншу зону вікна звичайним прийомом «Drag&Drop», а також обмінюватися з іншими вікнами через буфер обміну. Виводи елементів можна з'єднувати сигнальними провідниками або шляхом присвоювання однакових імен провідникам, які повинні бути з'єднані.

Waveform Editor – редактор часових діаграм (деякі розроблювачі називають цей додаток сигнальним редактором), що виконує подвійну функцію: на етапі введення забезпечує введення логіки проекту у вигляді діаграм (епюр) станів входів і виходів, а на етапі моделювання забезпечує введення діаграм тестових (еталонних) вхідних станів пристрою, що моделюється й завдання переліку виходів, що тестуються. Вікно сигнального редактора представлено на рис. 2.3.

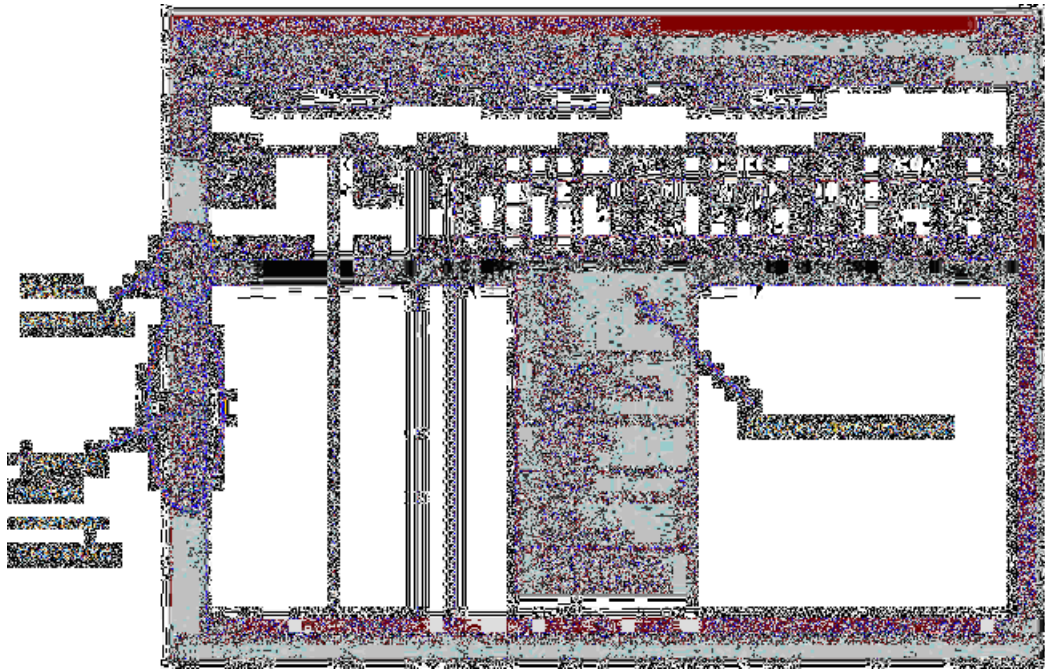


Рисунок 2.3 – Редактор часових діаграм

Час, упродовж якого буде проводитися моделювання, задається меню «File», пункт «End Time», крок часової сітки задається в меню «Options», пункт «Grid Size».

Вікно сигнального редактора має чотири поля, розділених вертикальними лініями. Перше поле ліворуч («Name») призначене для введення імені виводу, у другому полі («Type») відображається тип виводу (INPUT, OUTPUT, BIDIR), у третьому полі «Value» показані стани виводів, що відповідають положенню спеціальної вертикальної візирної лінії. Четверте поле призначене для завдання необхідних станів виводів, при цьому використовуються інструменти з панелі інструментів редактора, що розташована вертикально вздовж лівої сторони вікна. Активізація панелі інструментів відбувається тільки в тому випадку, якщо виділено один з вузлів. Щоб виділити вузол, необхідно клацнути лівою кнопкою миші на імені вузла, можна також виділити будь-яку ділянку вздовж горизонтальної осі, при цьому межі виділюваних ділянок прив'язуються до сітки.

Розміщаються виводи за допомогою спливаючого меню (рис. 2.3), пункт «Insert Node». Введені виводи можна редагувати, переміщати, видаляти,

розмножувати (з обов'язковим редагуванням імені або типу, якщо це необхідно).

Text Editor – текстовий редактор (рис. 2.4) є інструментом для створення текстових файлів проекту мовами опису апаратури: AHDL (*.tdf), VHDL (*.vhd), Verilog HDL (*.v). У цьому текстовому редакторі можна працювати також з довільним файлом формату ASCII.

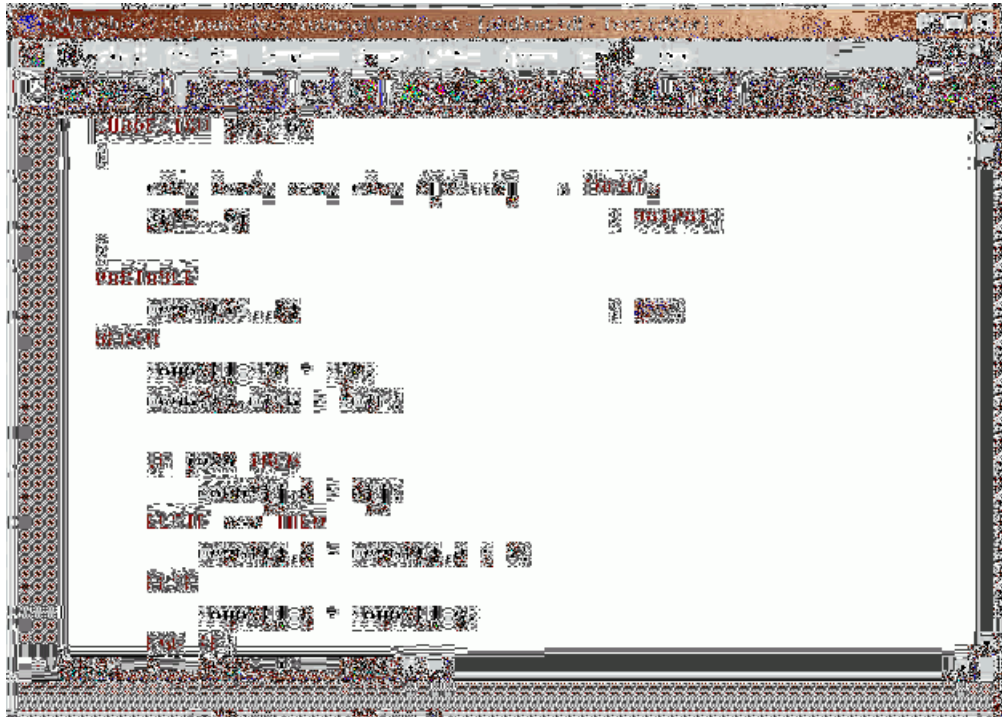


Рисунок 2.4 – Вікно текстового редактора

Усі перелічені файли проекту можна створювати в будь-якому текстовому редакторі, однак даний редактор має вбудовані можливості введення файлів проекту, їх компіляції й налагодження з видачею повідомлень про помилки і їх локалізацію у вихідному тексті або в тексті допоміжних файлів. Крім того, існують шаблони мовних конструкцій для AHDL, VHDL і Verilog HDL, реалізується виділення кольором синтаксичних конструкцій. У текстовому редакторі можна вручну редагувати файли призначень і конфігурації (*.acf), а також робити установки конфігурації для компілятора, симулятора та часового аналізатора.

Symbol Editor – символічний редактор дозволяє редагувати існуючі символи й створювати нові. Будь-який відкомпільований проект може бути

згорнутий до символу, збережений у бібліотеці символів і використаний як елемент у будь-якому іншому проєкті. Вікно символного редактора представлено на рис. 2.5.

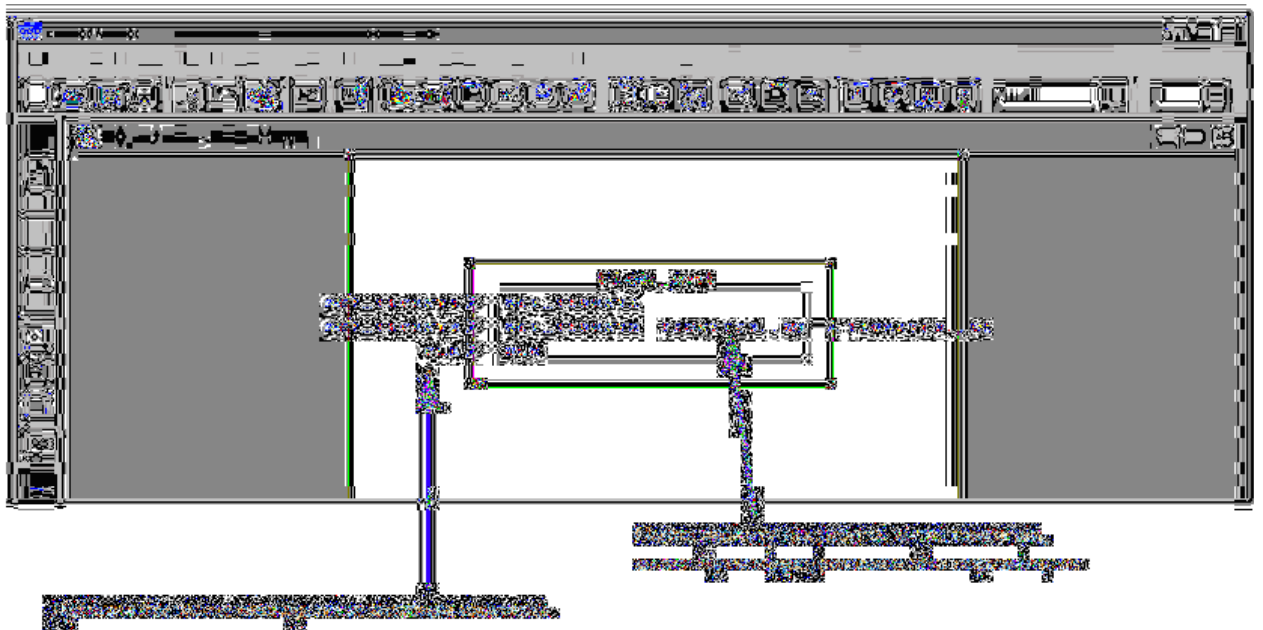


Рисунок 2.5 – Символьний редактор

Floorplan Editor – редактор зв'язків, дозволяє на плані розташування основних логічних елементів вручну розподіляти виводи ПЛІС (закріплювати виводи за конкретними входними й вихідними сигналами) і перерозподіляти внутрішні ресурси ПЛІС. Вікно редактора представлено на рис. 2.6.



Рисунок 2.6 – Редактор зв'язків системи MAX+PLUS II

Додатки для верифікації проектів

Simulator – додаток, що разом з редактором часових діаграм призначений для функціонального моделювання проекту (рис. 2.7) з метою перевірки правильності логіки його функціонування.

Timing Analyzer – додаток, що забезпечує розрахунок часових затримок від кожного входу до кожного логічно пов'язаного з ним виходу.

Нарешті, для програмування або конфігурування ПЛІС використовується додаток MAX+PLUS II Programmer (рис. 2.8).

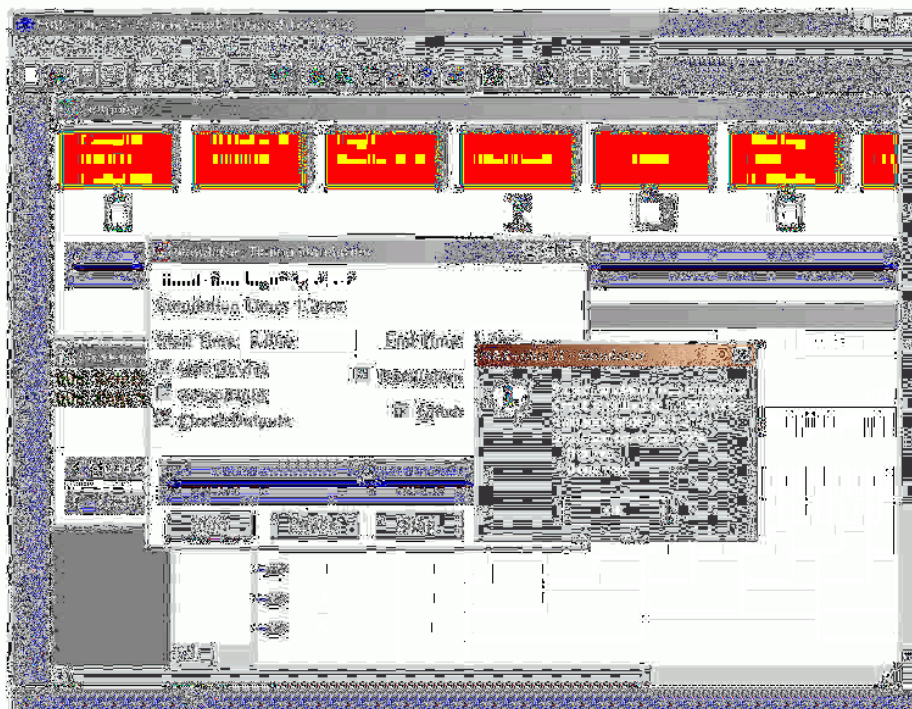


Рисунок 2.7 – Функціональне моделювання в системі MAX+PLUS II

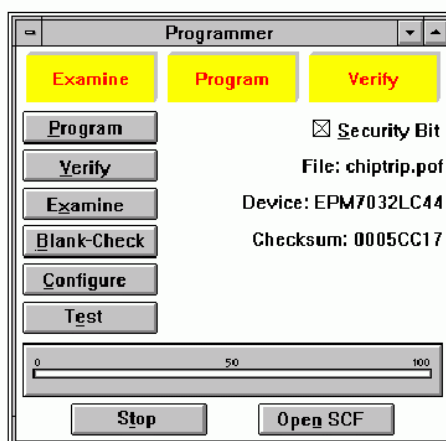


Рисунок 2.8 – Вікно додатка «Programmer» системи MAX+PLUS II

Програмування й перепрограмування мікросхем, що мають вбудовану систему програмування (ISP), може здійснюватися безпосередньо в складі виробу через спеціальний кабель, що підключається до LPT-порту (ByteBlaster) або до COM-порту (BitBlaster) персонального комп'ютера й технологічного 10-контактного з'єднувача JTAG, встановлюваного на платі виробу.

Сервісні додатки

До складу САПР MAX+PLUS II, крім того, входять три сервісних додатки:

Design Doctor – додаток, призначений для перевірки коректності проекту з використанням емпіричних правил.

Message Processor – процесор повідомлень (рис. 2.9), що забезпечує обробку, виведення на відображення й локалізацію (вказування місця в проекті, до якого воно відноситься) повідомлень трьох типів: повідомлень про помилки «Error», попереджень «Warning» та інформаційних повідомлень «Info». Причину виведення того або іншого повідомлення можна з'ясувати за допомоги опції «Help on Message» процесора повідомлень. При наявності повідомлень про помилки компіляція проекту неможлива до їх повного усунення. При наявності попереджень компіляція успішно завершується, однак наявність попередження свідчить про виявлення проблеми, що може призвести до правильного функціонування пристрою. Тому всі попередження повинні бути ретельно проаналізовані з використанням служби «Help on Message», для з'ясування причин їх появи та наступного усунення цих причин (або ігнорування попередження, що іноді буває можливо). Інформаційні повідомлення потрібно тільки брати до відома.

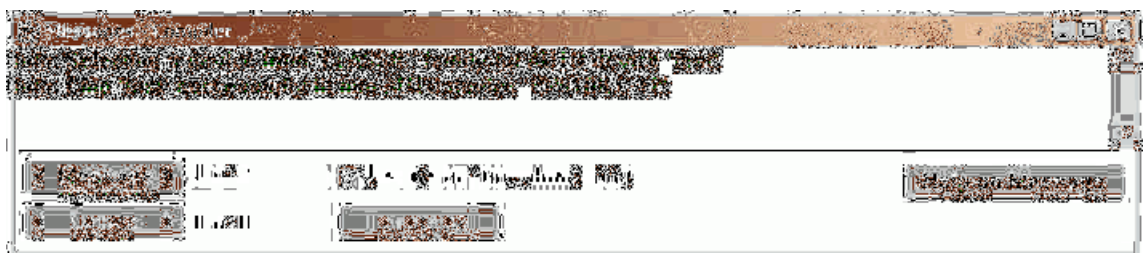


Рисунок 2.9 – Вікно процесора повідомлень системи

Hierarchy Display – додаток, що забезпечує огляд ієрархічної структури проекту, який може складатися з множини створених у різних редакторах і згорнутих у символи проектів більш низьких рівнів, причому кількість рівнів не обмежується.



Рисунок 2.10 – Вікно ієрархії проекту

Основний проект (проект найвищого рівня) повинен бути створений у графічному редакторі, якщо проект має тільки один рівень ієрархії, то він може бути створений у будь-якому редакторі.

Робочі каталоги системи

Під час інсталяції пакета MAX+PLUS II створюються два каталоги: каталог `..\MAXPLUS2`, що містить всі додатки та бібліотеки пакета, і каталог `..\MAX2WORK`, що містить підкаталог `..\CHIPTRIP` з усіма файлами навчального проекту, і ряд підкаталогів, використовуваних електронним довідником MAX+PLUS II Help. У цьому ж каталозі `..\MAX2WORK` варто розміщати й робочі каталоги створюваних проектів пристроїв.

Більша частина допоміжних файлів створюється й автоматично розміщується в каталозі проекту в процесі введення й компіляції проекту. Це насамперед файли призначень і конфігурації (*.ACF), файли звітів (*.RPT), файли даних для функціонального моделювання та часового аналізу (*.SNF), файли даних для програмування (*.POF) і ряд інших. Назви цих файлів завжди збігаються з назвою проекту. Деякі допоміжні файли створюються

користувачем: наприклад, для виконання функціонального моделювання створюється файл (*.SCF), що містить опис початкових і поточних станів вхідних сигналів (входів) і перелік виходів, для яких повинні бути визначені вихідні сигнали.

Порядок виконання роботи

1. Ознайомитися із системою проектування ALTERA MAX+PLUS II .
2. Ознайомитися з засобами створення проектів САПР ALTERA MAX+PLUS II.
3. Використовуючи персональну ЕОМ зі встановленим програмним пакетом ALTERA MAX+PLUS II, за допомогою графічного редактора скласти схему проекту та виконати часову симуляцію її роботи.
4. За допомогою пристрою ByteBlaster (MV), приєднаного до паралельного порту ПЕОМ, запрограмувати НВІС ПЛ.
5. Досягти правильної роботи проекту і налагодити схему.
6. Оформити звіт.

Зміст звіту

1. Титульна сторінка із зазначенням теми лабораторної роботи.
2. Мета, короткі теоретичні відомості.
3. Опис виконання лабораторної роботи.
4. Лістинг програми керування, створеної у редакторі середовища MPLAB IDE.
5. Висновки до роботи.

Контрольні питання

1. Опишіть процедуру створення проекту в середовищі MAX+PLUS II ALTERA.
2. Охарактеризуйте додатки програмного пакета MAX+PLUS II ALTERA, що забезпечують створення проекту.
3. Опишіть процес часової симуляції роботи ПЛІС.
4. Поясніть принцип створення схеми у графічному редакторі MAX+PLUS II ALTERA.

5. Опишіть процес симуляції роботи проектного пристрою.

Література: [1, с. 55–64; 2, с. 112–137; 3, с. 74–77].

Лабораторна робота № 3

Тема. Створення схеми проекту та симуляція його роботи засобами САПР MAX+PLUS II

Мета: засвоїти навички створення проекту логічної схеми за допомогою графічного редактора САПР MAX+PLUS II, дослідити роботу схеми з використанням сигнального редактора САПР MAX+PLUS II.

Короткі теоретичні відомості

Математичною основою цифрової електроніки й обчислювальної техніки є алгебра логіки або бульова алгебра. У бульовій алгебрі незалежні змінні або аргументи (X) приймають тільки два значення: «0» або «1». Залежні змінні або функції (Y) також можуть приймати тільки два значення: «0» або «1». Функція алгебри логіки (ФАЛ) представляється у вигляді:

$$Y = F(X_1, X_2, X_3 \dots X_N) . \quad (3.1)$$

Дана форма завдання ФАЛ називається алгебраїчною.

Основними логічними функціями є:

- логічне заперечення (інверсія):

$$Y = \bar{X} ;$$

- логічне додавання (диз'юнкція):

$$Y = X_1 + X_2 \text{ або } Y = X_1 \vee X_2 ;$$

- логічне множення (кон'юнкція):

$$Y = X_1 \cdot X_2 \text{ або } Y = X_1 \wedge X_2 .$$

До більш складних функцій алгебри логіки відносяться:

- функція рівнозначності (еквівалентності):

$$Y = X_1 \cdot X_2 + \bar{X}_1 \cdot \bar{X}_2 \text{ або } Y = X_1 \sim X_2 ;$$

- функція нерівнозначності (додавання по модулю два):

$$Y = X_1 \cdot \bar{X}_2 + \bar{X}_1 \cdot X_2 \text{ або } Y = X_1 \oplus X_2 ;$$

- функція Пірса (логічне додавання із запереченням):

$$Y = \overline{X_1 + X_2};$$

- функція Шеффера (логічне множення із запереченням):

$$Y = \overline{X_1 \cdot X_2};$$

Для бульової алгебри справедливі наступні закони й правила:

- розподільний закон:

$$X_1(X_2 + X_3) = X_1 \cdot X_2 + X_1 \cdot X_3,$$
$$X_1 + X_2 \cdot X_3 = (X_1 + X_2)(X_1 + X_3);$$

- правило повторення:

$$X \cdot X = X, X + X = X;$$

- правило заперечення:

$$X \cdot \bar{X} = 0, X + \bar{X} = 1;$$

- теорема де Моргана:

$$\overline{X_1 + X_2} = \bar{X}_1 \cdot \bar{X}_2, \overline{X_1 \cdot X_2} = \bar{X}_1 + \bar{X}_2;$$

- тотожності:

$$X_1 \cdot 1 = X, X + 0 = X, X \cdot 0 = 0, X + 1 = 1.$$

Схеми, що реалізують логічні функції, називаються логічними елементами. Основні логічні елементи мають, як правило, один вихід (Y) і декілька входів, число яких дорівнює числу аргументів ($X_1; X_2; X_3 \dots X_N$). На електричних схемах логічні елементи малюють у вигляді прямокутників з выводами для вхідних (ліворуч) і вихідних (праворуч) змінних. У середині прямокутника зображується символ, що позначає функціональне призначення елемента.

Логічні елементи, які реалізують операції кон'юнкції, диз'юнкції, функції Пірса й Шеффера, можуть бути, у загальному випадку, n -входовими. У таблиці істинності такого елемента кількість можливих комбінацій вхідних змінних N , у загальному випадку рівняється: $N = 2^n$, де n – число вхідних змінних.

Логічні елементи використовуються для побудови інтегральних мікросхем, які виконують різноманітні логічні й арифметичні операції.

ФАЛ будь-якої складності можна реалізувати за допомогою вказаних логічних елементів. Як приклад розглянемо ФАЛ, задану в алгебраїчній формі, у вигляді:

$$Y = X_1 + \bar{X}_2 X_3 \quad (3.2)$$

Для реалізації заданої функції на елементах «І-НЕ» її представити в базисі «І-НЕ», використовуючи подвійну інверсію функції та теорему де Моргана

$$Y = \overline{\overline{Y}} = \overline{\overline{X_1 + \bar{X}_2 X_3}} = \overline{\bar{X}_1(\bar{X}_2 \bar{X}_3)} \quad (3.3)$$

Реалізація проекту цифрової схеми в графічному редакторі САПР MAX+PLUS II. Розглянемо роботу із графічним редактором САПР MAX+PLUS II на прикладі схеми, представленої на рис. 3.1.

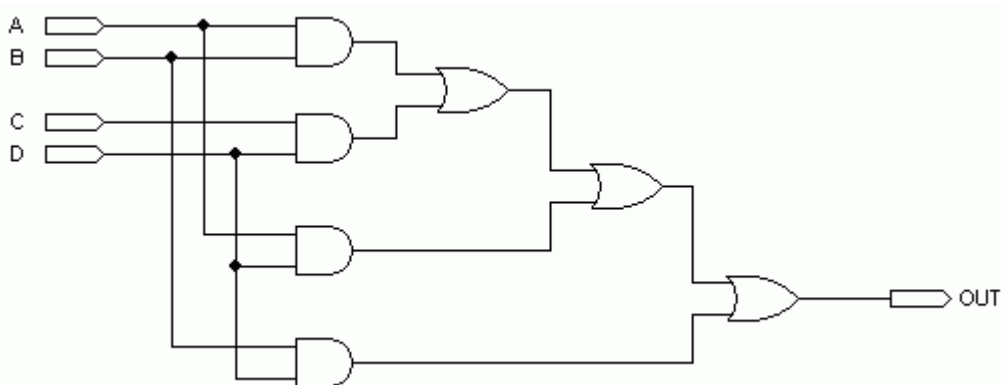


Рисунок 3.1 – Приклад цифрової схеми

Запустивши САПР MAX+PLUS II, за допомогою меню File/New... створюємо файл графічного редактора (рис. 3.2).



Рисунок 3.2 – Вікно пункту меню створення нового файлу «File/New...»

У створений файл вводимо схему логічного пристрою відповідно до завдання до лабораторної роботи. Для введення елементів схеми скористаємося меню «Enter Symbol» (введення символу), що викликається подвійним клацанням лівої кнопки мишки. У вікні, що відкрилося (рис. 3.3) обираємо необхідну бібліотеку примітивів «Symbol Libraries», вибираємо потрібний елемент «Symbol Files».

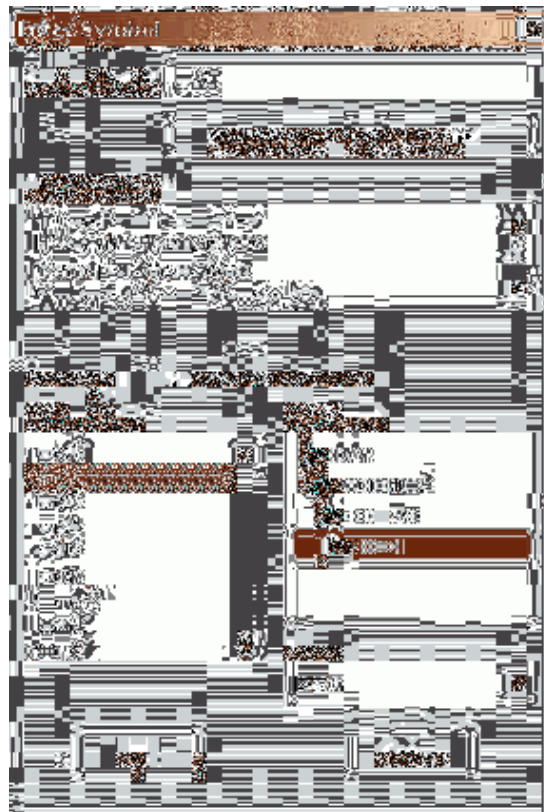


Рисунок 3.3 – Меню Enter Symbol

Після закінчення введення схеми зберігаємо файл у попередньо створену засобами Windows папку розроблюваного проекту (наприклад: ..\Lab_2), у цьому випадку лабораторної роботи, у робочому каталозі MAX+PLUS II: C:\max2work. Через меню «File/Save As...» (рис. 3.4) зберігаємо схему під вибраним ім'ям (наприклад: «lab_2»), при цьому розширення привласнюється автоматично.

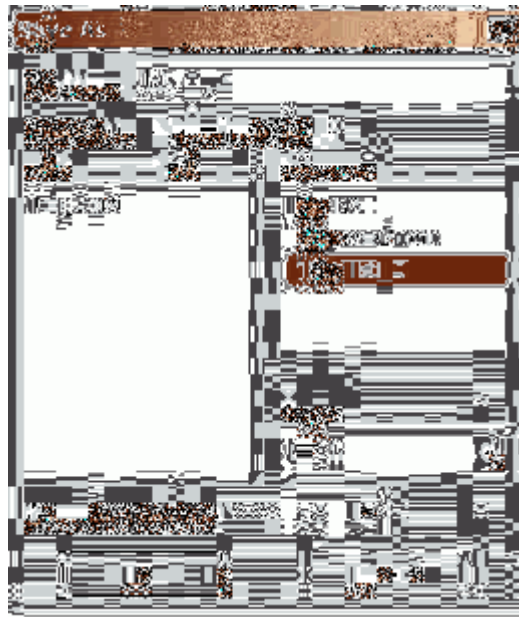


Рисунок 3.4 – Вікно збереження створеного файла меню «Save As...»

Після збереження необхідно ім'я файлу прив'язати до імені проекту – це здійснюється при виборі пункту Set Project to Current File у підменю Project меню File головного меню робочого вікна.

Проводимо перевірку введеної схеми. Для цього натискаємо піктограму Save&Check. При відсутності помилок введення проекту можна виконувати його компіляцію. Для цього відкриваємо вікно компілятора (рис. 3.5) і натискаємо кнопку START. Якщо не був призначений тип ПЛІС для компіляції проекту то система автоматично призначає відповідний тип мікросхеми ПЛІС, і сповіщає користувача у вікні Messages – Compiler (рис. 3.6).

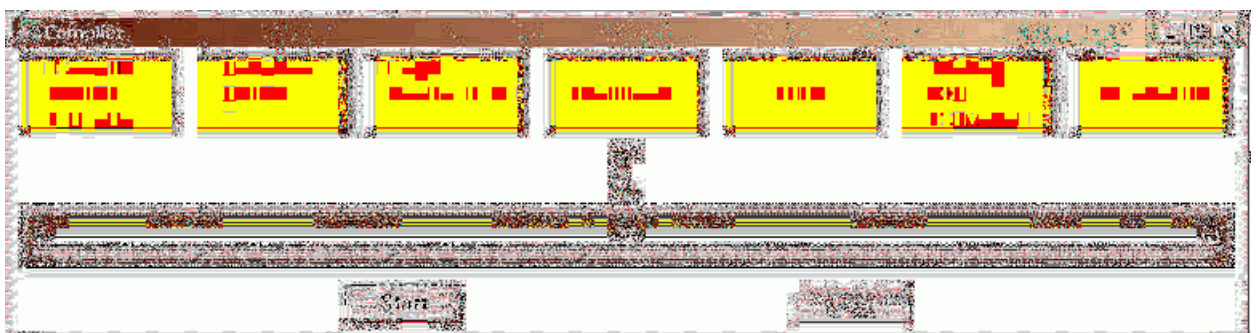


Рисунок 3.5 – Вікно компілятора

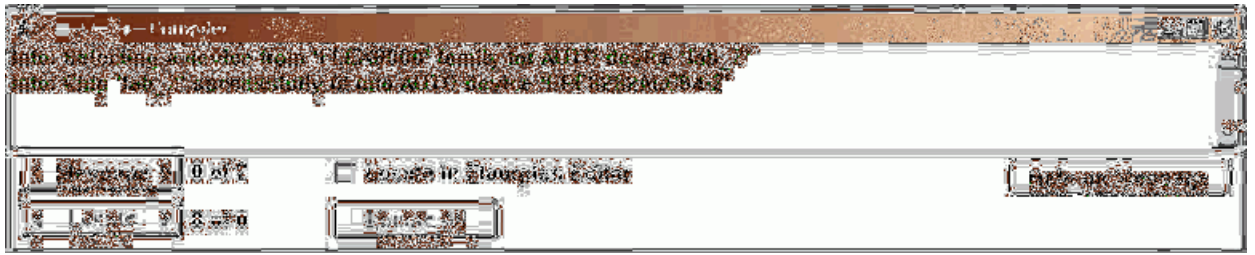


Рисунок 3.6 – Вікно повідомлень компілятора

Перевірка правильності функціонування введеної схеми виконується за допомогою сигнального редактора (Waveform Editor). Для цього відкриваємо сигнальний редактор і створюємо в ньому файл із розширенням *.scf. У створеному файлі за допомогою меню «File/End Time...» задаємо час моделювання, а в меню «Options/Grid Size...» крок сітки моделювання. Далі подвійним клацанням правої кнопки миші на полі Name: викликаємо меню «Insert Node» (рис. 3.7), за допомогою якого вибираємо входи й виходи схеми. Для вхідних виводів задаємо їх значення протягом необхідного часу моделювання. Після того як вхідні значення сигналів задані, відкриваємо вікно симулятора «Simulator Window» (рис. 3.8) та запускаємо його роботу натиснувши кнопку START.

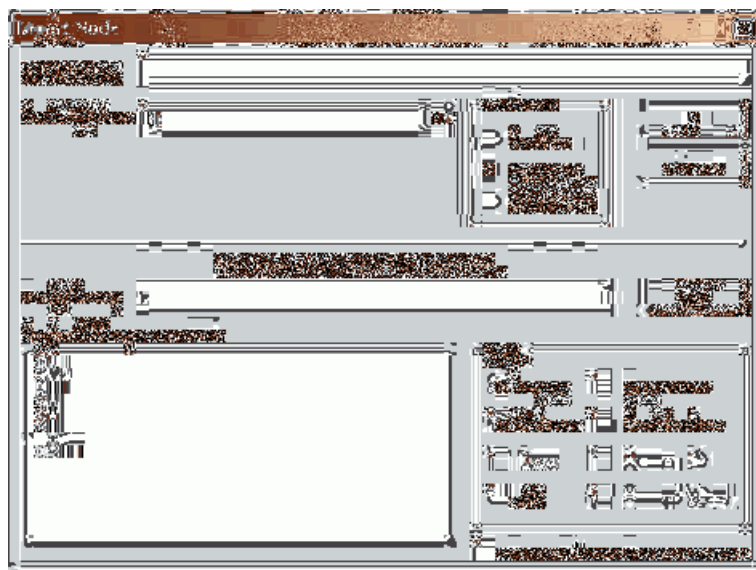


Рисунок 3.7 – Меню Insert Node



Рисунок 3.8 – Вікно симулятора (Simulator Window)

Результати моделювання роботи логічної схеми лабораторної роботи представлені на рис. 3.9.



Рисунок 3.9 – Результати моделювання роботи логічної схеми в сигнальному редакторі

Порядок виконання роботи

1. Вивчити правила побудови, принципи роботи логічних схем.
2. Синтезувати електричну принципову схему логічного пристрою, описаного заданим викладачем рівнянням в алгебраїчній формі.
3. Ввести синтезовану схему в графічному редакторі САПР MAX+PLUS II.
4. Виконати симуляцію роботи схеми. Зарисувати діаграми роботи та за

її результатами заповнити таблицю істинності змодельованої схеми.

5. Відповісти на контрольні питання, оформити звіт щодо лабораторної роботи.

Зміст звіту

1. Назва та мета лабораторної роботи.
2. Схема лабораторної установки.
3. Короткі відомості щодо призначення елементів лабораторного стенда.
4. Висновки щодо роботи.

Контрольні питання

1. Назвіть основні логічні (бульові) функції й зобразіть елементи, що їх реалізують. Для кожної з функцій запишіть таблицю істинності.
2. Які логічні елементи доступні в бібліотеці примітивів графічного редактора MAX+PLUS II?
3. Які процеси протікають у системі при компіляції проекту?
4. Пояснить результати моделювання роботи схеми лабораторної роботи.

Література: [1, с. 117 – 133; 2, с. 98 – 112; 3, с. 134 – 156].

Лабораторна робота № 4

Тема. Вивчення опису логічних схем засобами мови програмування AHDL

Мета: надбати основних навичок опису цифрових схем за допомогою мови опису апаратури AHDL, створити опис логічної схеми за допомогою текстового редактора САПР MAX+PLUS II.

Короткі теоретичні відомості

Мова опису апаратури AHDL розроблена фірмою Altera і призначена для опису комбінаційних і послідовностних логічних пристроїв, групових операцій, цифрових автоматів (state machine) і таблиць істинності з урахуванням архітектурних особливостей мікросхем ПЛІС фірми Altera. Вона повністю інтегрується із системою автоматизованого проектування ПЛІС MAX+PLUS II. Файли опису апаратури, написані мовою AHDL, мають розширення *.TDF

(Text design file). Для створення TDF-Файлу можна використовувати як текстовий редактор системи MAX+PLUS II, так і будь-який інший. Проект, виконаний у вигляді TDF-Файлу, компілюється, відлагоджується й використовується для формування файлу програмування або завантаження ПЛІС фірми Altera.

Оператори та елементи мови AHDL є досить потужним і універсальним засобом опису алгоритмів функціонування цифрових пристроїв, зручним у використанні. Мова опису апаратури AHDL дає можливість створювати ієрархічні проекти в рамках однієї цієї мови або ж в ієрархічному проекті використовувати як TDF-Файли, написані мовою AHDL, так й інші типи файлів.

При розподілі ресурсів пристроїв розроблювач може користуватися командами текстового редактора або операторами мови AHDL для того, щоб здійснити призначення ресурсів і пристроїв. Крім того, розроблювач може тільки перевірити синтаксис або виконати повну компіляцію для налагодження та запуску проекту. Будь-які помилки автоматично виявляються оброблювачем повідомлень і висвічуються у вікні текстового редактора.

Елементи мови AHDL. Зарезервовані ключові слова

Зарезервовані ключові слова використовуються для наступних цілей:

- для позначення початку, закінчення та переходів в оголошеннях мови AHDL;
- для позначення визначених констант, тобто GND і VCC.

Ключові слова можна використовувати як символічні імена, тільки якщо вони вкладені в символи одинарних лапок ('). Їх можна також використовувати в коментарях.

Для того, щоб одержати контекстову допомогу за ключовим словом, необхідно переконатися, що файл збережений з розширенням *.tdf, потім натиснути одночасно дві кнопки Shift+F1 у вікні текстового редактора Text Editor і клацнути кнопкою миші на ключовому слові.

Фірма Altera рекомендує всі ключові слова набирати прописними

буквами. Список всіх зарезервованих ключових слів мови AHDL наведений у таблиці 4.1:

Таблиця 4.1 – Зарезервовані ключові слова мови AHDL

FUNCTION	OTHERS	
CASE	TABLE	JKFFE
BITS	SRFFE	NCLUDE
DFF	VCC	NODE
DFFE	WHEN	NOR
ELSE	WITH	NOT
END	XNOR	OPTIONS
EXP	XOR	OR
AND	GLOBAL	OUTPUT
BEGIN	GND	RETURNS
BURIED	INPUT	SOFT
BIDIR	IF	SRFF
CARRY	IS	STATES
CASCADE	JKFF	SUBDESIGN
CLIQUE	LATCH	TFF
CONNECTED_PINS	LCELL	TFFE
CONSTANT	MACHINE	THEN
DEFAULTS	MACRO	TITLE
DESIGN	MCELL	TRI
DEVICE	NAND	VARIABLE
ELSIF	OF	X

Символи

В таблиці 4.2 наведені символи, що мають певне значення в мові AHDL. До цього переліку не включені символи, що використовуються в булевих виразах як оператори та для операцій порівняння.

Таблиця 4.2 – Призначення символів, використовуваних в мові AHDL

Символ	Функція
_ (підкреслення)	Використовувані користувачем ідентифікатори
- (тире)	Символи в символічних іменах
-- (два тире)	Починає коментар у стилі VHDL, який триває до кінця рядка
% (відсоток)	Містить із двох сторін коментар стилю AHDL
() (круглі дужки)	Містять і визначають послідовні імена груп; імена виводів у секції підпроекта (Subdesign Section) і в прототипах функцій; (необов'язково) входи й виходи таблиць в оголошенні Truth Table; стани в оголошенні цифрового автомата State Machine; більш пріоритетні операції в булевих виразах; необов'язкові варіанти в секції проекту «Design Section» (усередині оголошення призначення ресурсів Assignment)
[] (квадратні дужки)	Містять діапазон значень в десятковому ім'ї групи
'...' (одинарні лапки)	Містять символічні імена
"..." (подвійні лапки)	Містять рядок в оголошенні назви Title; цифри в не десяткових номерах; шлях в оголошенні Include. Можуть (необов'язково) містити ім'я проекту й пристрою в секції проекту «Design Section»; ім'я в оголошенні призначення ключі графа Clique Assignment
. (крапка)	Відокремлює символічні імена змінних у макрофункції або примітиви від імен портів. Відокремлює ім'я файлу від розширення
... (три крапки)	Розділяє найменше й найбільше значення в діапазонах

Продовження таблиці 4.2

; (крапка з комою)	Закінчує оголошення й секції в мові AHDL
, (кома)	Розділяє елементи послідовних груп і списків
: (двокрапка)	Відокремлює символічні імена від типів в оголошеннях і призначеннях ресурсів
@ «собака»	Привласнює символічні вузли виводам пристрою й логічних осередків в оголошеннях призначення ресурсів Resource Assignment
= (рівність)	Привласнює значення за замовчуванням GND і VCC входам у секції підпроекта Subdesign Привласнює настановні значення у варіантах Привласнює значення станам у машині станів Привласнює значення в бульових рівняннях
=> (стрілка)	Відокремлює входи від виходів в оголошеннях таблиці істинності Truth Table; пропозиції з WHEN від булевих виразів в операторі Case

Імена в лапках і без лапок

Мова AHDL використовує три типи імен:

1. Символічні імена – це обумовлені користувачем ідентифікатори.

Вони використовуються для позначення наступних частин TDF:

- внутрішніх і зовнішніх вузлів (вершин);
- констант;
- змінних цифрового автомата, бітів станів, імен станів;
- прикладів (Instance).

2. Імена підпроекту – це обумовлені користувачем імена для файлів проекту більше низького рівня. Ім'я підпроекту повинне бути таким самим, як і ім'я файлу TDF.

3. Імена портів – це символічні імена, що ідентифікують вхід або вихід примітива або макрофункції.

У файлі *.fit проекту можуть з'явитися імена виводів, що генеруються компілятором, із символом «тильда» (~). Цей символ зарезервований для імен, що генеруються компілятором, користувачеві забороняється його використовувати для позначення імен виводів, вузлів (вершин), груп (шин).

Існують дві форми запису для всіх трьох типів імен (символічних, підпроєкту й портів): у лапках (‘) і без лапок.

Якщо розроблювач створює символ за замовчуванням для файлу TDF, що містить у собі імена портів у лапках, власне лапки не входять в імена виводів.

Числа в мові AHDL

У мові AHDL можна використовувати десяткові, двійкові, вісімкові та шістнадцяткові числа в будь-якій комбінації. У таблиці 4.3 наведений синтаксис запису чисел мовою AHDL для кожної системи числення.

Таблиця 4.3 – Синтаксис представлення чисел мовою AHDL

Система числення	Значення
Десяткова	<послідовність цифр 0-9>
Двійкова	B"<послідовність із 0, 1, X>", де символ X позначає індиферентне значення
Вісімкова	O"<послідовність цифр 0-7>" або Q"<послідовність цифр 0-7>"
Шістнадцяткова	X"< послідовність цифр 0-9, літер A-F>" або H"< послідовність цифр 0-9, літер A-F>"

Бульові вирази

Бульові вирази складаються з операндів, розділених логічними та арифметичними операторами й компараторами й (необов'язково) згрупованих за допомогою круглих дужок. Вирази використовуються в бульових рівняннях, а також в інших конструкціях мови, таких як оператори Case та If.

Існують наступні застосування бульових виразів:

1. Операнд.

Приклад: a, b[5..1], 7, VCC

2. Вбудоване в текст (in-line) посилання (reference) на примітив або макрофункцію.

3. Префіксний оператор (! або -), застосований до бульового виразу.

Приклад: !c

4. Два бульових вирази, розділені двійковим (не префіксним) оператором.

Приклад: d1 \$ d3

5. Вкладений в круглі дужки бульовий вираз.

Приклад: (!foo & bar)

Результат кожного бульового виразу повинен мати ту ж саму ширину, що й вузол або група (у лівій стороні рівняння), якому він, в остаточному підсумку, привласнюється.

Логічні оператори

У таблиці 4.4 наведені логічні оператори для бульових виразів.

Таблиця 4.4 – Логічні оператори для бульових виразів мовою AHDL

Оператор:	Приклад:	Опис:
!	!tob	Доповнення (префіксне звернення)
NOT	NOT tob	
&	bread & butter	Логічне І
AND	bread AND butter	
!&	a[3..1]! & b[53]	
NAND	a[3..1] NAND b [5..3]	Обернення І
#	ttick # treat	Логічне АБО
OR	trick OR treat	
!#	c[8..5] !# d[7..4]	Обернення логічного АБО
NOR	c [8..5] NOR d [7..4]	

Продовження таблиці 4.4

\$	foo \$ bar	АБО, що виключає
XOR	foo XOR bar	
!\$	x2 !\$ x4	Обернення логічного АБО, що виключає
XNOR	x2 XNOR x4	

Кожний оператор являє собою логічний вентиль із двома входами; виключення становить оператор NOT, що є префіксним інвертором. Для запису логічного оператора можна використовувати його ім'я або символ.

Вирази, у яких використовуються ці оператори, інтерпретуються по-різному, залежно від того, що являють собою операнди: одиночні вузли (вершини), групи або числа. Крім того, вирази з оператором NOT інтерпретуються не так, як інші логічні оператори.

Порядок виконання роботи

1. Вивчити основні елементи мови AHDL і правила опису логічних схем.
2. Виконати опис електричної схеми заданої в попередній роботі за допомогою текстового редактора САПР MAX+PLUS II.
3. Виконати симуляцію роботи схеми. Замалювати діаграми роботи й за її результатами заповнити таблицю істинності змодельованої схеми.
4. Порівняти результати, отримані в ході виконання лабораторної роботи з результатами, отриманими в роботі №2.
5. Відповісти на контрольні питання, оформити звіт про виконану роботу.

Зміст звіту

1. Назва та мета лабораторної роботи.
2. Схема лабораторної установки.
3. Короткі відомості щодо призначення елементів лабораторного стенда.
4. Висновки щодо роботи.

Контрольні питання

1. Що таке мова опису апаратури? Назвіть існуючі мови опису

апаратури, у чому полягає їх відмінність?

2. Назвіть основні елементи мови AHDL, дайте їх коротку характеристику.

3. Як описуються логічні елементи мовою AHDL?

Література: [1, с. 17 – 46; 2, с. 8 – 33; 3, с. 44 – 56, 7 с. 5 – 40].

Лабораторна робота № 5

Тема. Розробка проекту складного цифрового пристрою на базі НЛС ПЛІС

Мета: надбати основних навичок проектування складних цифрових пристроїв за допомогою САПР MAX+PLUS II, дослідити роботу пристрою з використанням редакторів САПР MAX+PLUS II і лабораторного стенда НЛС ПЛІС.

Короткі теоретичні відомості

На першому етапі проектування на підставі аналізу технічного завдання (ТЗ) виявляються специфічні вимоги проекту, що дозволяють зупинити свій вибір на певній фірмі, що випускає великі інтегральні схеми програмованої логіки, і на певному сімействі ПЛІС цієї фірми. Відбір здійснюється на основі аналізу характеристик як самих ВІС – логічних, конструктивних, експлуатаційних, вартісних, так і на аналізі властивостей необхідної або припустимої САПР. Найчастіше вибір визначається вже наявним практичним доробком і досвідом роботи проектувальника із продукцією та САПР певної фірми. У цьому випадку уточнюється сімейство, архітектурні та експлуатаційні характеристики якого задовольняють вимогам ТЗ. Вибір сімейства може істотно залежати від специфічних вимог проекту, наприклад, необхідності відповідності певним інтерфейсним стандартам, вимоги наявності швидкісної вбудованої пам'яті значного обсягу, підвищеної радіаційної стійкості й т. д. На вибір можуть впливати й такі характеристики, як умови поставок, оголошення про розробку перспективних модифікацій сімейства й ряд інших міркувань. Одним із самих неприємних (і затратних за сукупністю наслідків) варіантів є

з'ясування в ході виконання проекту неможливості його реалізації на продукції вибраної фірми.

Аналіз комплексу інтерфейсних вимог до проєктованого пристрою дозволяє конкретизувати кількість зовнішніх контактів, необхідних для реалізації проєкту, тобто типорозмір корпусу ВІС вибраного сімейства ПЛІС. Складність проєкту або певні вимоги до проєкту (наприклад, швидкісні характеристики) можуть приводити до доцільності використання на початкових етапах проєктування групи САПР сторонніх фірм.

Специфікація проєкту

Завдання цього етапу – перехід від технічного завдання до формалізованого опису проєктованого пристрою. Як правило, ТЗ є комбінацією словесного та технічного опису, його формалізація призводить до виявлення основних блоків пристрою (або алгоритму) і визначення їх зв'язків і/або взаємодії. По суті, саме в цей момент реалізуються початкові дії другого етапу. Формально ж зміст робіт цього етапу проєкту – розбиття завдання на окремі функціонально відособлені підзадачі – етап декомпозиції. Спосіб і засоби розбиття найчастіше визначаються саме функціональної завершеністю та відособленістю окремих фрагментів, хоча в значній мірі тут суттєву роль відіграють просто вподобання проєктувальника, і лише іноді розбивка є повністю визначеною. Сама форма ТЗ може провокувати проєктувальника на використання тих або інших засобів, хоча не виключено, що більш ефективним міг би бути інший метод опису проєкту або його фрагментів. Декомпозиція може зводитися до складання схем алгоритмів функціонування фрагментів або до функціональної схеми пристрою і його частин. Можливим варіантом для досить складних систем буде розумне сполучення і поведінкової, і структурної розбивки проєкту. Розбивка здійснюється не тільки в рамках одного рівня ієрархії. Для більшості проєктів здійснюється ще й розбивка на ієрархічно-організовані рівні.

Істотним завданням, розв'язуваним на цьому етапі, є уточнення й узгодження із замовником інтерфейсних функцій проєктованого пристрою. При

цьому уточнюється реалізація протоколів зовнішнього обміну. Саме необхідні часові характеристики та правила взаємодії із зовнішніми приладами визначають припустиму організацію й структуру внутрішніх вузлів проекту.

Використання САПР на цьому етапі проектування поки що явище досить рідке, хоча для реалізації сучасних дуже складних проектів (кілька сотень тисяч вентилів) все частіше використовуються спеціальні блокові редактори, що дозволяють здійснювати декомпозицію проекту без деталізації складових частин. Прикладом може служити САПР Quartus фірми Altera, що має у своєму складі спеціальний засіб для редагування проекту на рівні блоків (block-level editing).

Розробка загальної структури проекту

Основні завдання цього етапу проектування – вибір припустимих для реалізації кожного рівня ієрархії елементів, визначення зв'язків між ними, і якщо параметри елементів є такими, що потребують налаштування, то й їх налаштування. Ряд моментів є для даного етапу визначальним: з одного боку – це джерело набору припустимих елементів, а з іншого боку – засобів опису з'єднань елементів між собою, а при необхідності, і можливість опису нових (специфічних для конкретного проекту) елементів.

Як вже вказувалося, можливий як лише часовий (поведінковий), так і тільки просторовий (архітектурно-структурний) опис проекту. Однак зазвичай доцільно сполучати обидві можливості. При розробці пристроїв з цифровим поданням інформації буває природною розбивка їх на два блоки: операційний блок і пристрій керування. Операційний блок (ОБ) виконує перетворення даних і будується зі стандартних частин (частин з певною поведінкою), а пристрій керування (пристрій керування, ПК) забезпечує необхідну послідовність операцій, виконуваних у ОБ (одному або декількох). Для цього ПК передає на входи ОБ керуючі сигнали. Послідовність дій і відповідність керуючих сигналів залежить від результатів операцій у ОБ та зовнішніх впливів. Звідси видно, що ПК зручно задавати у формі кінцевого автомата з пам'яттю (АП) того чи іншого типу.

У складних проектах можливий поділ ПК на декілька функціонально слабо пов'язаних пар «ОБ-ПК» на одному рівні ієрархії або створення пари, ієрархічно інтегрованої в ОБ (рідше в ПК).

Приклад реалізації проекту з комбінованим описом

Порядок використання основних додатків системи MAX+PLUS II розглянемо на прикладі розробки проекту секундоміра. Як індикатор використовується чотири семисегментних індикатори із загальним катодом, тобто сигнали керування сегментами повинні мати активні високі рівні для підсвічування сегментів.

Основними елементами такого пристрою є:

- схема розподілу тактового сигналу генератора;
- рахунковий каскад секунд і хвилин;
- комбінаційні схеми перетворення двійкових кодів у сигнали керування сегментами індикатора.

У розроблювальний проект доцільно включити два рівні ієрархії:

- рівень структурних елементів, у який входить схема розподілу сигналу генератора, схеми лічильників секунд і хвилин, комбінаційні схеми перетворення двійкових кодів у сигнали керування сегментами індикатора;
- рівень проекту в цілому.

Перед початком роботи в системі MAX+PLUS II у робочому каталозі системи, що був створений при інсталяції САПР за замовчуванням, C:\max2work необхідно створити каталог, у якому будуть перебувати файли проекту. Це пов'язано з тим, що в процесі роботи MAX+PLUS II створює велику кількість допоміжних файлів, що належать тільки до даного проекту. Так само рекомендується видаляти файли, що належать до проекту тільки через меню «File\Delete File...» [2] .

Розробку проекту можна розпочати із проектування двійково-десятькового лічильника в графічному редакторі (Graphic Editor). Двійково-десятьковий лічильник може бути побудований на Т-тригерах за схемою, представленою на

рис. 5.1.

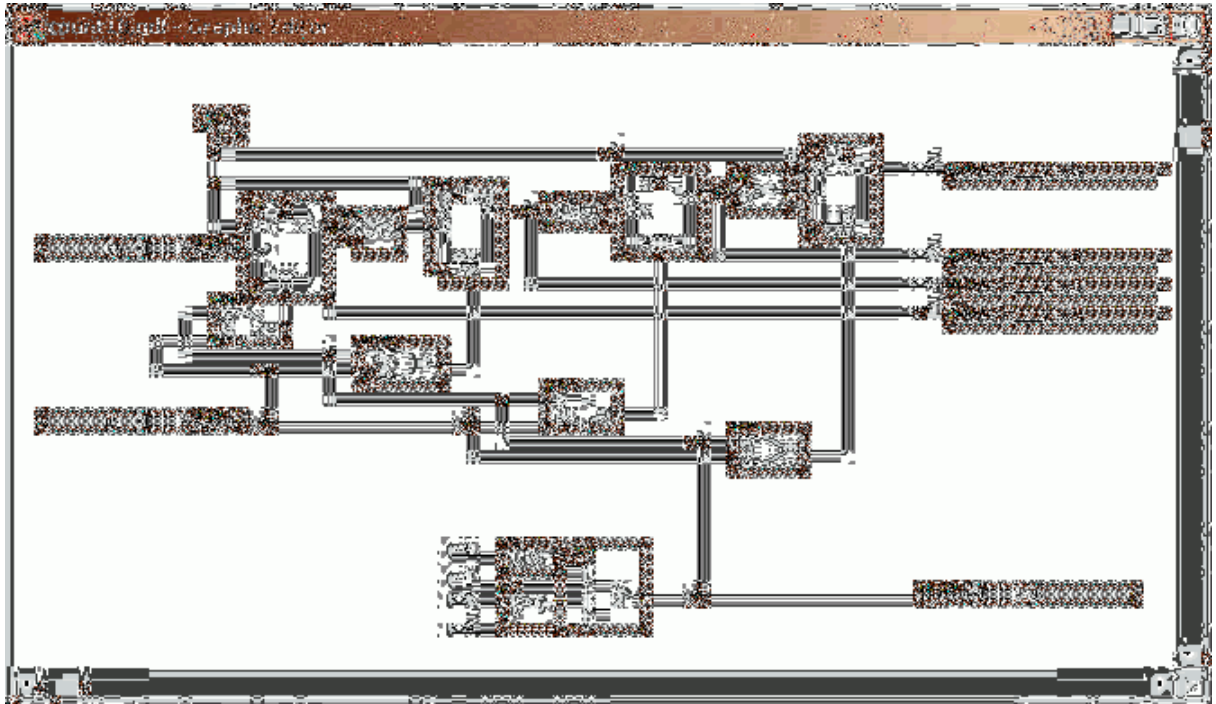



Рисунок 5.1 – Схема двійково-десятькового лічильника, створена в середовищі графічного редактора

У процесі розробки можна користуватися бібліотеками примітивів, макрофункцій, LPM – функцій, які доступні через меню «Symbol\Enter Symbol...», або подвійним клацанням лівої кнопки миші на вільному місці вікна графічного редактора.

Для включення схеми лічильника у файл верхнього рівня ієрархії проекту секундоміра необхідно створити символ проекту лічильника через меню «File\Create Default Symbol», причому перед створенням символу необхідно провести перевірку на предмет наявності можливих помилок через меню «File\Project\Save & Check» або натисканням відповідної кнопки  на панелі інструментів.

Для створення проекту дешифратора вихідного двійкового коду лічильників у сигнали керування елементами семисегментних індикаторів можна скористатися текстовим редактором (Text Editor), у якому алгоритм роботи представлений мовою опису цифрової апаратури, у даному випадку мовою AHDL [3].

Для завдання відповідності двійкових кодів кодам керування семисегментних індикаторів мовою AHDL використовується оголошення TABLE. На рис. 5.2 наведений текст файлу 7seg.tdf, що описує логіку роботи дешифратора.

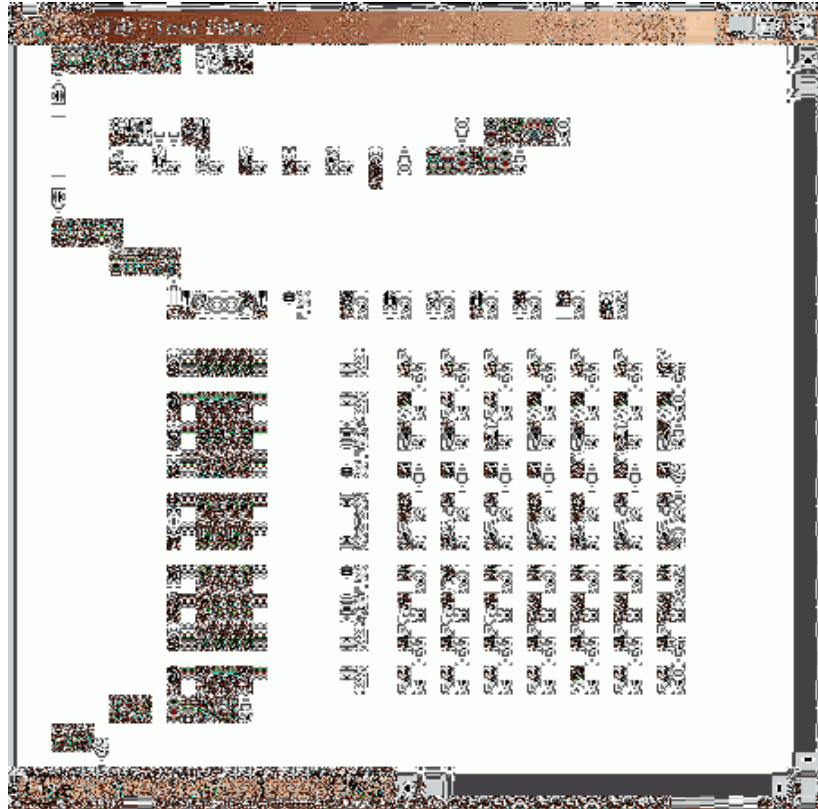


Рисунок 5.2 – Опис логіки дешифратора

Далі, як було описано вище, створюється символ проекту дешифратора, із призначенням файлу 7seg.tdf файлом верхнього рівня ієрархії проекту секундоміра.

Для реалізації секундоміра необхідний подільник вхідних тактових імпульсів генератора з відповідним коефіцієнтом поділу. Наприклад, для поділу частоти генератора лабораторного стенда необхідний подільник з коефіцієнтом поділу $8 \cdot 10^6$, його символ також можна створити в кожному з редакторів САПР MAX+PLUS II.

Після того як всі елементи проєктованого пристрою створені, переходимо до створення файлу верхнього рівня. На рис. 5.3 наведена загальна схема проєкту секундоміра, де елемент DIV являє собою дільник вхідних імпульсів з

коефіцієнтом поділу $8 \cdot 10^6$, елемент COUNT6 – двійково-десятковий лічильник з коефіцієнтом рахунку, що дорівнює 6, COUNTER – двійково-десятковий лічильник з коефіцієнтом рахунку 10, 7SEG – дешифратор.

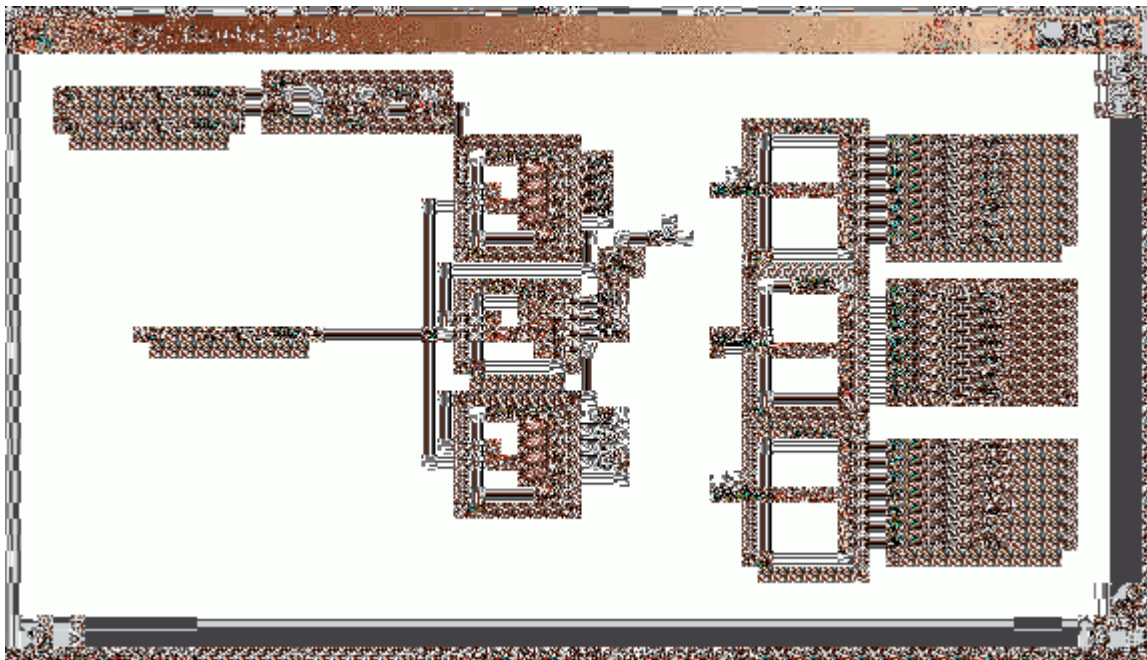


Рисунок 5.3 – Схема проекту секундоміра

Далі відповідно до послідовності розробки проектів на базі ПЛІС виконується призначення типу мікросхеми для реалізації розробленого проекту (меню «Assign\Device...») і призначення виводів ПЛІС (меню «Assign\PinLocationChip...») або за допомогою редактора зв'язків (Floorplan Editor)). Після того як всі призначення проведені, проект необхідно відкомпілювати та виконати симуляцію. На завершення процесу компіляції створюється файл для подальшого конфігурування ПЛІС.

Перевірка правильності функціонування введеної схеми виконується за допомогою сигнального редактора (Waveform Editor). Для цього відкриваємо сигнальний редактор і створюємо в ньому файл із розширенням *.scf. У створеному файлі за допомогою меню File/End Time... задаємо час моделювання, а в меню Options/Grid Size... крок сітки моделювання. Далі подвійним клацанням правої кнопки миші на полі Name викликаємо меню Insert Node (рис. 5.4), за допомогою якого вибираємо входи й виходи схеми.

Для вхідних виводів задаємо їх значення протягом необхідного часу моделювання. Після того, як вхідні значення задані, відкриваємо вікно симулятора (Simulator Window) (рис. 5.5) й запускаємо його, натиснувши кнопку START.

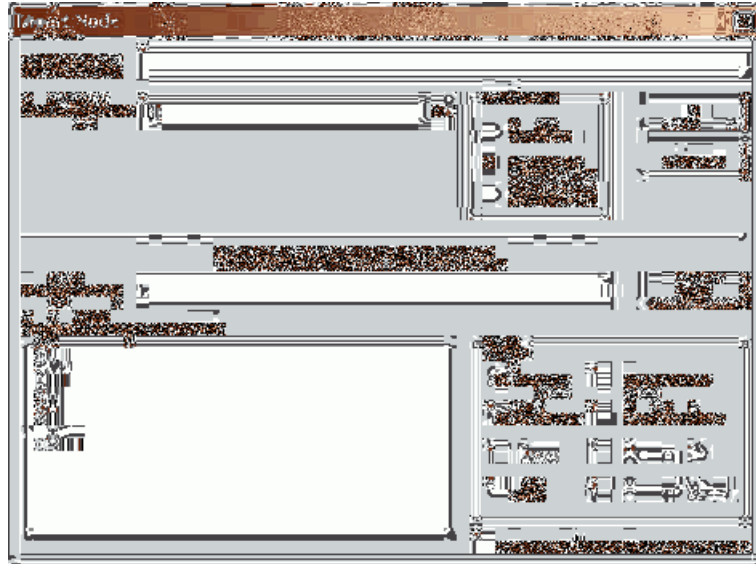


Рисунок 5.4 – Меню Insert Node



Рисунок 5.5 – Вікно симулятора (Simulator Window)

Результати моделювання роботи схеми лабораторної роботи представлені на рис. 5.6.

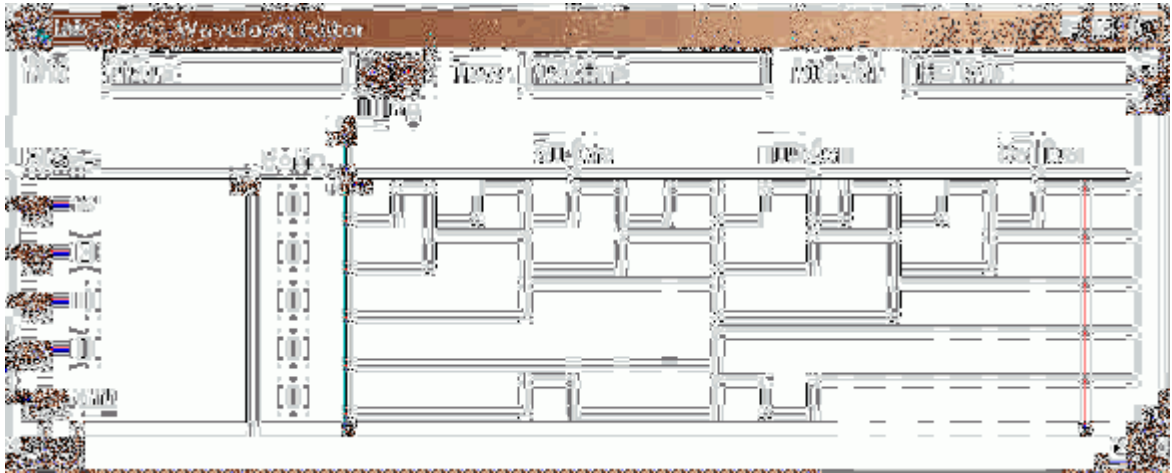


Рисунок 5.6 – Результати моделювання роботи схеми в сигнальному редакторі

Процес конфігурування ПЛІС починається натисканням на кнопку «Configure» вікна програматора САПР MAX+PLUS II.

Після успішного конфігурування НЛС ПЛІС відразу ж готовий до виконання функцій секундоміра. На семисегментних індикаторах відображується відлік часу. Керування роботою пристрою здійснюється за допомогою кнопки пуску/зупинки рахунку та кнопки скидання секундоміра, визначених в процесі призначення виводів ПЛІС.

Порядок виконання роботи

1. Ознайомитися із принципами розробки складних цифрових пристроїв засобами САПР MAX+PLUS II.
2. Розробити структурну схему проєктованого пристрою.
3. Кожний функціональний блок проєктованого пристрою розробити окремо й об'єднати їх у загальний файл верхнього рівня.
4. Промоделювати роботу схеми в сигнальному редакторі, оцінити часові затримки схеми.
5. Дослідити роботу розробленого пристрою за допомогою лабораторного стенда.
6. Відповісти на контрольні питання, оформити звіт про виконану роботу.

Зміст звіту

1. Назва та мета лабораторної роботи.
2. Структурна схему проєктованого пристрою.
3. Результати моделювання роботи схеми засобами сигнального редактора, часові діаграми роботи схеми.
4. Висновки з роботи.

Контрольні питання

1. Поясніть принципи роботи розробленого пристрою.
2. Як реалізується лічильник із заданим числом рахунку?
3. Що таке дешифратор? Як реалізовані дешифратори у проєкті?
4. Як призначити виводи ПЛІС в MAX+PLUS II?
5. Чи завжди необхідно прибігати до призначення виводів ПЛІС вручну, обґрунтуйте свою відповідь.

Література: [1, с. 220-232; 2, с. 181-192; 3, с. 201-216; 6, с. 40-49].

3 КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАНЬ СТУДЕНТІВ

Кількість лабораторних занять – 10 годин (5 лабораторних занять).

Поточний контроль на лабораторних заняттях протягом змістових модулів:

– відвідування лабораторних занять – 0,5 бала за заняття (максимум 5 балів за семестр);

– виконання лабораторного завдання – 0,5 бала за виконане лабораторне завдання (максимум 5 балів за семестр);

– захист лабораторної роботи – захист на оцінку «відмінно» – 2 бали за лабораторну роботу, захист на оцінку «добре» – 1 бал за лабораторну роботу, захист на оцінку «задовільно» – 0,5 бала за лабораторну роботу (максимум 10 балів, 5 балів та 2,5 бали за семестр);

– своєчасність захисту усіх лабораторних робіт – 0,25 бала за лабораторну роботу (максимум 1,25 бали за семестр).

СПИСОК ЛИТЕРАТУРИ

1. Антонов А.П. Язык описания цифровых устройств AlteraHDL: Практический курс. – М.: ИП «Радиософт», 2001. – 224 с.
2. Соловьёв В. В. Проектирование цифровых систем на основе программируемых логических интегральных схем. – М.: Горячая линия – Телеком, 2001. – 636 с.
3. Стешенко В.С. ПЛИС фирмы ALTERA: проектирование устройств обработки сигналов – М.: Додека, 2000. – 128 с.
4. ALTERA® ACEX 1K Programmable Logic Family Data Sheet.
5. ALTERA® APEX 20K Programmable Logic Device Family Data Sheet.
6. ALTERA® APEX 20KC Programmable Logic Device Family Data Sheet.
7. ALTERA® Altera Programming Hardware Data Sheet.

Методичні вказівки щодо виконання лабораторних робіт з навчальної дисципліни «Проектування електромеханічних систем» для студентів денної та заочної форм навчання зі спеціальності 141 – «Електроенергетика, електротехніка та електромеханіка» за освітньо-науковою та освітньо-професійною програмами «Електромеханічне обладнання енергоємних виробництв» освітнього ступеня «Магістр»

Укладач старш. викл. М. Ю. Юхименко

Відповідальний за випуск зав. кафедри проф. САУЕ Д. Й. Родькін

Підп. до др. _____. Формат 60x84 1/16. Папір тип. Друк ризографія.
Ум. друк. арк. _____. Наклад _____ прим. Зам. № _____. Безкоштовно.

Видавничий відділ КрНУ імені Михайла Остроградського
39600, м. Кременчук, вул. Першотравнева, 20