

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Тернопільський національний економічний університет**  
**Факультет комп'ютерних інформаційних технологій**  
**Кафедра комп'ютерної інженерії**

**Цьома Богдан Іванович**

## **HDL-модель роботи панелі LED освітлення / HDL-model of LED-lightening panel**

спеціальність: 6.050102 - Комп'ютерна інженерія  
освітньо-професійна програма - Комп'ютерні системи та мережі

Випускна кваліфікаційна робота

Виконав: студент групи КСМ-42/1  
Цьома Богдан Іванович

---

Науковий керівник:  
к.т.н., Дубчак Л.О.

---

Випускну кваліфікаційну роботу  
допущено до захисту:

" \_\_\_\_ " \_\_\_\_\_ 20\_\_ р.

Завідувач кафедри  
О. М. Березький

**ТЕРНОПІЛЬ - 2019**

## РЕЗЮМЕ

Робота виконана на 62 сторінках, містить 11 рисунків, 14 таблиць, 3 додатки, 2 аркуші графічних матеріалів.

Сучасні системи освітлення та індикації використовують різноманітні технології їх реалізації. Одним із широко розповсюджених типів освітлення є використання LED компонентів. Проте використання стандартних засобів не завжди забезпечує необхідні параметри обладнання.

Тому розробка сучасних засобів освітлення з різноманітними функціями є важливою та актуальною задачею.

Метою дипломного проекту є розробка HDL-моделі панелі LED-освітлення.

Ключові слова: LED-ОСВІТЛЕННЯ, СВІТЛОДІОД, VHDL, ACTIVE-HDL, FPGA.

## RESUME

Work is executed on 62 pages, contains 11 figures, 14 tables, 3 additions, 2 graphic material.

Modern lighting and display systems use a variety of technologies for their implementation. One of the most common types of lighting is the use of LED components. However, the use of standard means does not always provide the necessary equipment parameters.

Therefore, the development of modern lighting equipment with a variety of functions is an important and topical task.

The purpose of the diploma project is to develop an HDL-model of the LED-lighting panel.

Keywords: LED-LIGHTING, LED, VHDL, ACTIVE-HDL, FPGA.

## ЗМІСТ

Вступ.....	9
1 Стан предметної області.....	10
1.1 Сучасні світлодіоди та їх застосування.....	10
1.2 Основні типи світло діодів.....	13
1.3 Постановка задачі та аналіз дерева рішень.....	20
2 Проектування пристрою.....	21
2.1 Опис середовища Active-HDL.....	21
2.2 Особливості розробленої моделі.....	24
2.3 Розробка структурної схеми.....	26
3 Дослідження роботи розробленої HDL-моделі.....	32
3.1 Розробка моделі індикатора LED-панелі.....	32
3.2 Функціональна симуляція проекту.....	36
3.3 Верифікація проекту.....	39
4 Техніко-економічне обґрунтування розробки.....	42
4.1 Розрахунок витрат на розробку програмного забезпечення.....	42
4.2 Визначення експлуатаційних витрат.....	48
4.3 Розрахунок ціни споживання програмного продукту.....	51
4.4 Визначення показників економічної ефективності.....	52
Висновки.....	54
Список використаних джерел.....	55
Додаток А Програмний опис пристрою.....	57
Додаток Б Світлокопія тез доповіді.....	59
Додаток В Довідка про впровадження.....	62

					БР.КСМ.07127/15.00.00.000 ПЗ
Зм.	Арк	№ докум.	Підпис	Дата	
Розробив		П'юма Б.І.			HDL-модель роботи панелі LED освітлення
Перевірив		Лубчак Л.О.			
Н. Контр.		Гураль І.В.			THEU.FKIT.KCM-42
Затв.		Березький О.М.			

## ВСТУП

Сучасні системи освітлення та індикації використовують різноманітні технології їх реалізації. Одним із широко розповсюджених типів освітлення є використання LED компонентів. Проте використання стандартних засобів не завжди забезпечує необхідні параметри обладнання.

Тому розробка сучасних засобів освітлення з різноманітними функціями є важливою та актуальною задачею.

Попри яскраве випромінювання, LED-освітлення надзвичайно енергоекономне. Повністю пожегобезпечне, оскільки практично не нагрівається. Не призводить до пошкоджень текстилю, пластику чи будь-якого іншого матеріалу при контакті. Світлодіодна стрічка відмінно тримається на усіх поверхнях, навіть на гладкому кахлі. LED має вигляд яскравого ланцюжка з точкових світильників, які не лише освітлюють кімнату, але й «тонують» її у різні відтінки.

Біле світло LED не втомлює очей та візуально не впливає на колір предметів. Може бути змонтовано у формі у будь-якого візерунку чи конфігурації. Екологічно безпечне для навколишнього середовища. Особливо, якщо порівнювати із люмінесцентними лампами, що містять ртуть та приносять нищівну шкоду природі при неправильній утилізації.

LED-лампи міцні та надійні, витримують жорстку експлуатацію, оскільки корпус виготовлено не зі скла, а з надміцного пластику. Ця характеристика, беззаперечно, також впливає на їхню довговічність.

Метою даної бакалаврської роботи є проектування VHDLмоделі панелі LED-освітлення.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

# 1 СТАН ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Сучасні світлодіоди та їх застосування

LED (англ. light-emitting diode) – світловипромінюючий діод. Світлодіод - це напівпровідникове джерело світла, яке випромінює світло, при проходженні через нього постійного струму. Електрони в напівпровіднику рекомбінуються з електронними дірками, випускаючи енергію у вигляді фотонів. Цей ефект називається електролюмінесценцією [1].

Світло може випромінюватись у видимому спектрі, а також у інфрачервоному та ультрафіолетовому світлі. Колір світла (що відповідає енергії фотонів) визначається енергією, необхідною для перетину електронів у смугі напівпровідника. Біле світло отримують за допомогою декількох напівпровідників або шару світло випромінюючого люмінофора на напівпровідниковому пристрої. Вигляд світлодіода зображено на рисунку 1.1.

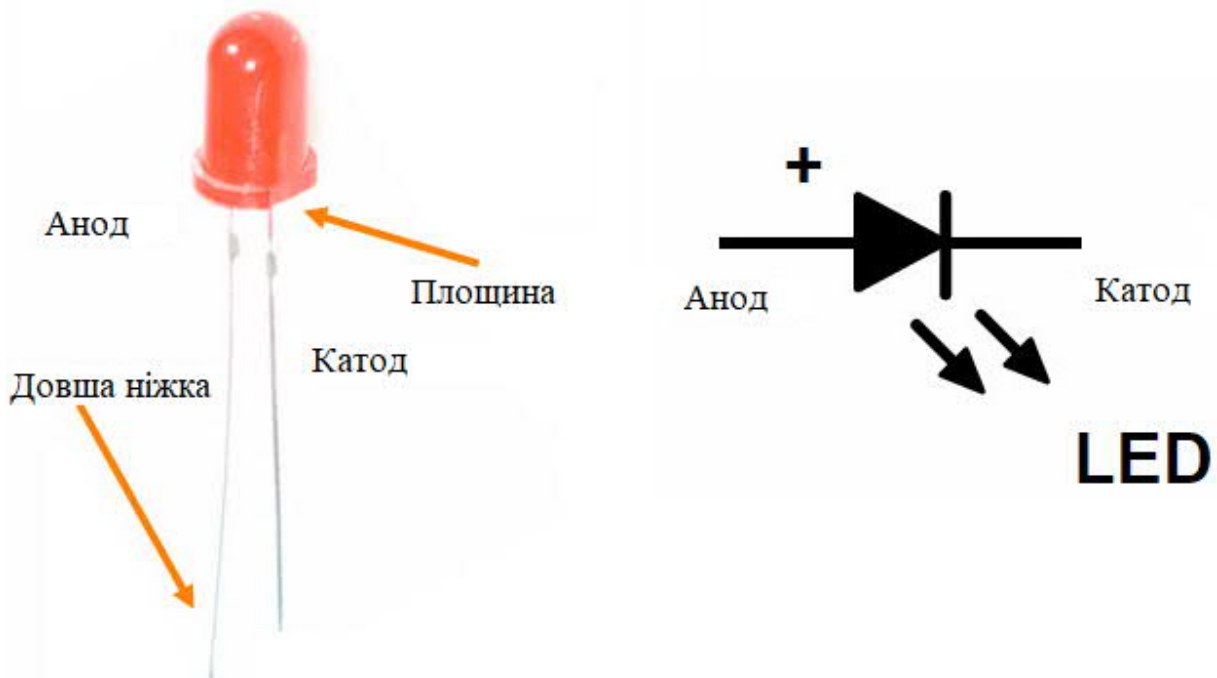


Рисунок 1.1 – Загальний вигляд і позначення світлодіода

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

Ранні світлодіоди часто використовувалися як індикаторні лампи, замінюючи невеликі лампи розжарювання, а також в семисегментних дисплеях. Останні розробки дали світлодіоди білого світла, придатні для освітлення приміщень. Світлодіоди привели до створення нових типів дисплеїв і датчиків, а їх високі коефіцієнти перемикачності корисні в сучасних комунікаційних технологіях.

Світлодіоди мають багато переваг перед джерелами світла з лампами розжарювання, включаючи зниження енергоспоживання, більш тривалий термін служби, поліпшену фізичну надійність, менший розмір і швидше перемикачності.

Світловипромінюючі діоди використовуються в таких різноманітних пристроях, як авіаційне освітлення, автомобільні фари, реклама, загальне освітлення, світлофори, спалахи камери, освітлені шпалери та медичні пристрої.

На відміну від лазера, колір світла, що випромінюється світлодіодом, не є ні когерентним, ні монохроматичним, але спектр вузький по відношенню до людського зору і функціонально монохроматичний. Враховуючи вище зазначені параметри, вони мають широку палітру застосувань.

Вперше, у 1907 році британський експериментатор Генрі Джозеф Раунд в лабораторіях Марконі помітив, що коли потенціал 10 вольт застосовується до кристала карборунду (карбиду кремнію), він випромінює жовтувате світло. Проте першим його дослідив і запропонував робочу теорію Олег Володимирович Лосєв з Росії. У 1927 р. науковець опублікував статтю «Світловий карборундний детектор і ефект виявлення і коливання кристалами».

Протягом десятиліть не було досягнуто прогресу з різних причин. Рубін Браунштейн, який працював в Radio Corporation of America, в 1955 році повідомив, що деякі прості діоди випромінюють інфрачервоне світло при підключенні до струму. У 1961 році Гарі Піттман і Боб Біард з Texas Instruments виявили, що галій-арсенідний діод випромінює інфрачервоне світло кожного разу, коли він підключається до струму. У тому ж році вони отримали патент на інфрачервоний світлодіод [2]. Нік Холоняк-молодший, зайнятий в General Electric, розробив у 1962 році перший світлодіод, який випромінював світло у видимій

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

частині діапазону частот. Це був червоний світлодіод. У 1972 р. М. Джордж Крафорд, який був аспірантом Холоняка, винайшов перший жовтий світлодіод і яскравіший червоний світлодіод. В 1976 році Томас П. Пірсалл розробив світлодіоди високої яскравості для використання у волоконній оптиці в телекомунікаціях. Шуджі Накамура з корпорації Nichia зробив перший синій світлодіод в 1979 році, але він був занадто дорогим для комерційного використання до 1994 року. Тепер світловипромінюючі діоди можуть бути зроблені в одному або декількох кольорах.

Перші комерційні світлодіоди зазвичай використовувалися як заміна ламп розжарювання та неонових індикаторів, а також в семисегментних дисплеях спочатку у дорогому обладнанні, наприклад лабораторному та електронному випробувальному обладнанні, пізніше в таких пристроях, як калькулятори, телевізори, радіоприймачі, телефонів, а також годинників [3]. До 1968 року видимі та інфрачервоні світлодіоди були надзвичайно дорогими, близько 200 доларів США за одиницю, і тому не мали практичного використання. Компанія Monsanto була першою організацією, яка масово виробляла видимі світлодіоди, використавши в 1968 році фосфід арсеніду галію (GaAsP) для виробництва червоних світлодіодів, придатних для індикаторів. Hewlett-Packard (HP) представив світлодіоди в 1968 році, спочатку використовуючи GaAsP, що поставлявся компанією Monsanto. Ці червоні світлодіоди були досить яскравими тільки для використання в якості індикаторів, оскільки світлового випромінювання було недостатньо для освітлення області. Покази в калькуляторах були настільки малі, що пластикові лінзи були побудовані над кожною цифрою, щоб зробити їх розбірливими. Пізніше інші кольори стали широко доступними і з'явилися в приладах та обладнанні. У 1970-х роках компанія Fairchild Optoelectronics випустила комерційно успішні світлодіодні пристрої вартістю менш ніж п'ять центів кожен [4]. Ці пристрої використовували з'єднані напівпровідникові мікросхеми, виготовлені за планарною технологією, винайденою доктором Жаном Хорні з Fairchild Semiconductor. Поєднання планарної обробки для виготовлення чіпів та інноваційних методів упаковки

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		



дозволило команді Fairchild на чолі з першопрохідцем оптоелектроніки Томасом Брандтом досягти необхідного скорочення витрат. Виробники світлодіодів продовжують використовувати ці методи.

Ранні світлодіоди були запаковані в металевих корпусах, аналогічних транзисторам, зі скляним вікном або лінзою, що випромінювали світло. Сучасні індикаторні світлодіоди упаковані в прозорі формовані пластикові корпуси, трубчасті або прямокутні, і часто підфарбовані відповідно до кольору пристрою. Інфрачервоні пристрої можуть бути пофарбовані, щоб блокувати видиме світло. Більш складні пакети були адаптовані для ефективного розсіювання тепла у потужних світлодіодах. Поверхневі світлодіоди ще більше зменшують розмір приладу. Світлодіоди, призначені для використання з волоконно-оптичними кабелями, можуть бути оснащені оптичним роз'ємом.

## 1.2 Основні типи світлодіодів

LED з видимим світлом використовується в якості заміни для ламп розжарювання і неонових освітлень, як елементи в семисегментних дисплеях, на великих дисплеях RGB, в семафорах та інших візуальних сигналах, в калькуляторах, годинниках і в ліхтарях. Інфрачервоні світлодіоди використовуються в пристроях дистанційного керування в телевізорах, DVD-дисках та інших місцях, які потребують бездротового керування.

Переваг світлодіодів багато, але вони також мають свої недоліки. Переваги полягають у тому, що вони випромінюють більше світла на одиницю потужності ніж лампи розжарювання, вони набагато менші, час їх вмикання / вимикання набагато коротший, ніж у інших типів електричних джерел світла (вони швидкі), термін служби набагато довший, і їх важче пошкодити. Їх недоліки - висока ціна за люмен, висока залежність від зовнішньої температури і легкий перегрів, якщо зовнішня температура занадто висока і немає тепловідведення. Незважаючи на

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

свої недоліки, світлодіоди знаходять своє місце в людському користуванні і закріплюють свої позиції, витісняючи ті ж лампи розжарювання.

Світлодіоди розраховані на пряму напругу, тобто на величину напруги, необхідну для того, щоб світлодіод проводив електрику. Наприклад, всі 5мм світлодіоди мають номінальний струм 20мА, але номінальна пряма напруга змінюється на різних світлодіодах.

Червоні світлодіоди мають максимальне значення напруги 2,2 В, сині світлодіоди мають максимальне значення напруги 3,4 В, а білі світлодіоди мають максимальне значення напруги 3,6 В.

Наступне зображення (рисунок 1.2) показує схему простого світлодіодного ланцюга, що складається з 5мм білого світлодіода з джерелом живлення 5В.

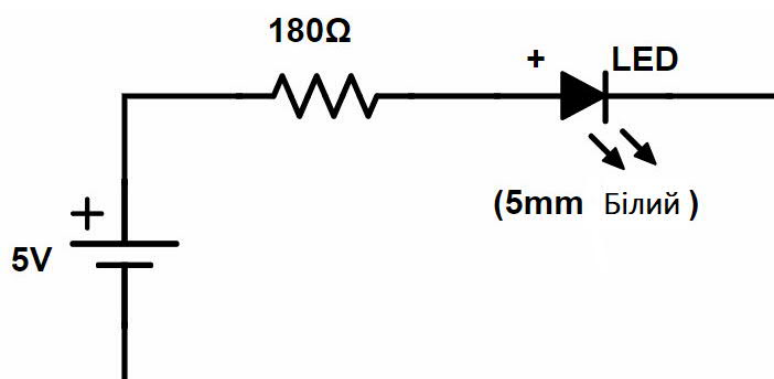


Рисунок 1.2 – Проста схема для роботи світлодіода

Світлодіоди, що проходять через отвір (рисунок 1.3) доступні в різних формах і розмірах, а найбільш поширеними є світлодіоди 3 мм, 5 мм і 8 мм. Ці світлодіоди доступні в різних кольорах, таких як червоний, синій, жовтий, зелений, білий тощо.



Рисунок 1.3 – Світлодіоди для отворів

Поверхневі або SMD світлодіоди (рисунок 1.4) являють собою спеціальні пакети, які можна легко встановлювати на поверхню друкованої плати. SMD світлодіоди зазвичай диференціюються на основі їх фізичних розмірів. Наприклад, найпоширеніші світлодіоди SMD - 3528 і 5050 [3].

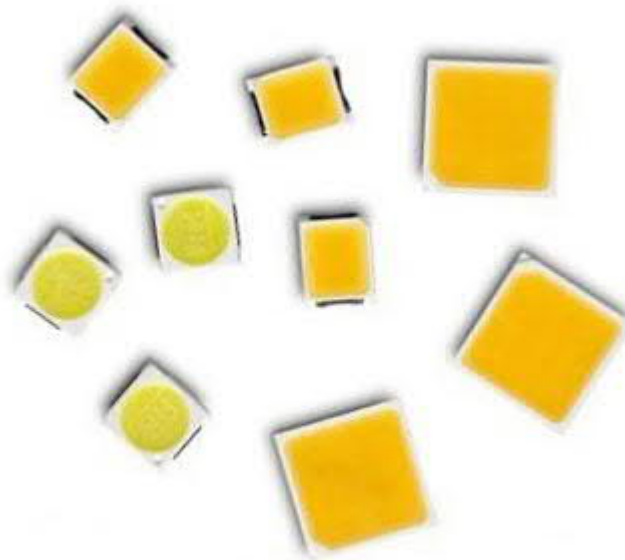


Рисунок 1.4 - SMD світлодіоди

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

Наступний тип світлодіодів (рисунок 1.5) - двоколірний світлодіод, як випливає з назви, може випромінювати два кольори. Двоколірні світлодіоди мають три провідника, як правило, два аноди і загальний катод. Залежно від вибору провідників, певний колір буде активовано.

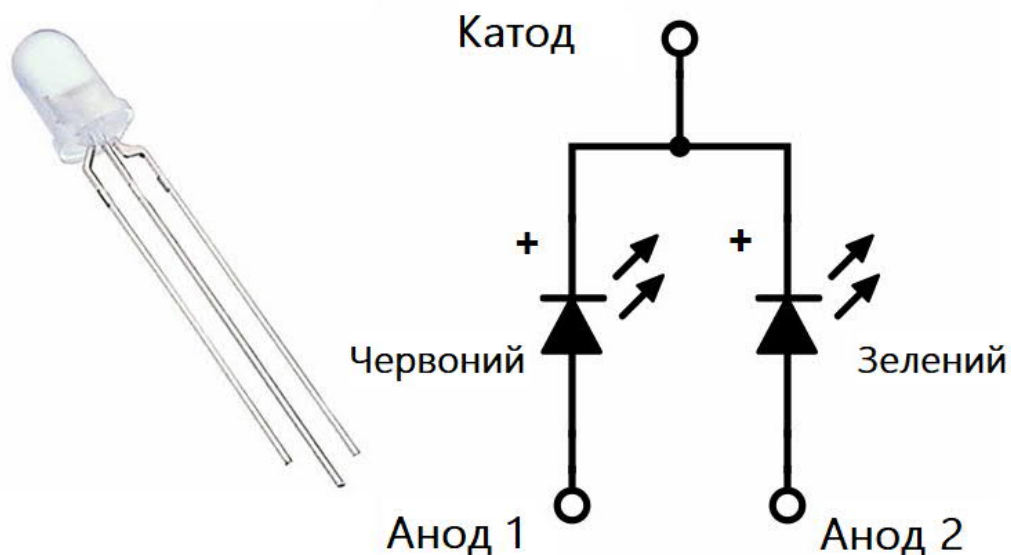


Рисунок 1.5 – Двоколірні світлодіоди

Світлодіоди RGB - це найулюбленіші і найпопулярніші світлодіоди серед любителів та дизайнерів (рисунок 1.6). Навіть комп'ютерні збірки дуже популярні для реалізації світлодіодів RGB в комп'ютерних корпусах, материнських платах, оперативної пам'яті тощо. Світлодіоди RGB виробляють широкий спектр кольорів і велику різноманітність застосувань.



Рисунок 1.6 – RGB світлодіоди

Світлодіод з потужністю, більшою або рівною 1 Вт, називається світлодіодом високої потужності (рисунок 1.7). Це пояснюється тим, що звичайні світлодіоди мають розсіювання потужності в декілька міліват.

Високоякісні світлодіоди дуже яскраві і часто використовуються у ліхтарях, фарах автомобілів, прожекторах і т.п.

Оскільки потужність цих світлодіодів висока, потрібно правильне охолодження і використання радіаторів. Крім того, потреба вхідної потужності для цих світлодіодів буде зазвичай дуже високою.

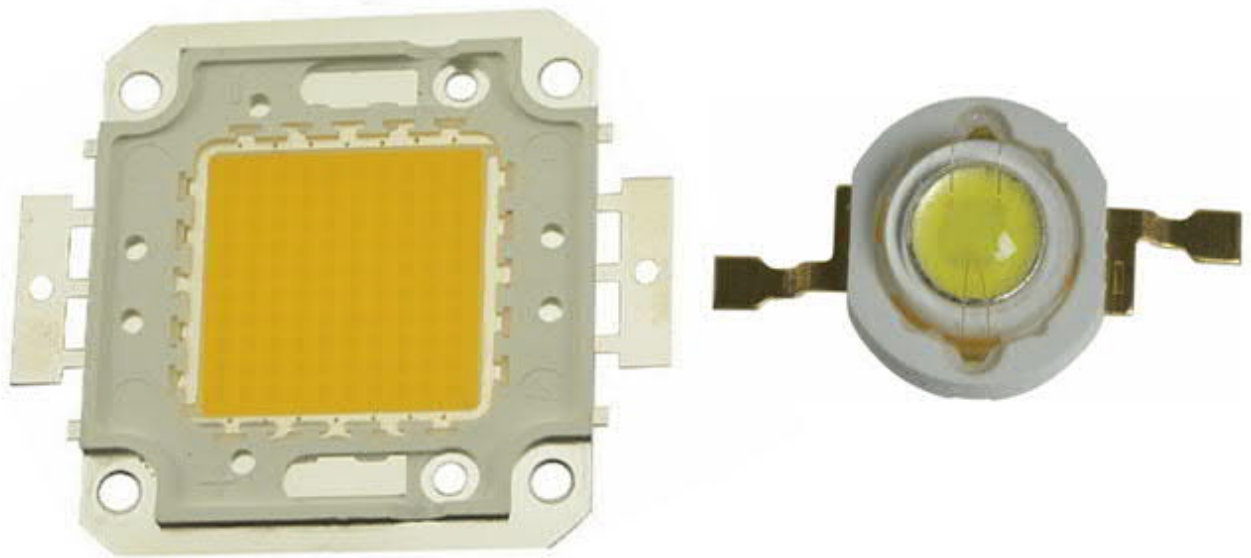


Рисунок 1.7 – Високопотужні світлодіоди

Буквенно-цифрові світлодіоди (рисунок 1.8). У цьому підтипі специфічних для прикладних світлодіодів фактично є 4 буквенно-цифрові підтипи: 7-сегментний обробляє певний набір букв і всі цифри; 14- і 16-сегментний Це буквенно-цифрові світлодіоди, широко відомі як відображення зоряного вибуху, оскільки вони можуть охоплювати весь римський алфавіт у верхньому регістрі разом з цифрами від 0 до 9. 16-сегментний світлодіод має перерву на нижній та верхній панелях, щоб покращити вигляд деяких символів.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

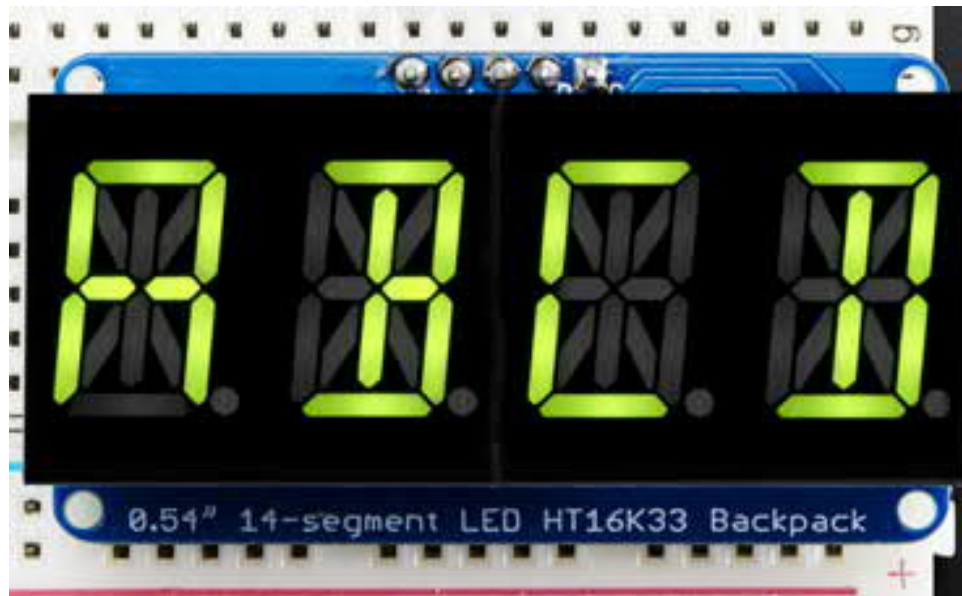


Рисунок 1.8 – Буквенно-цифрові світлодіоди

Matrix – це найрізноманітніший алфавітно-цифровий світлодіод і, він обробляє як малі, так і великі букви алфавіту, всі цифри і навіть широкий спектр символів [3], він зображений на рисунку 1.9.



Рисунок 1.9 – Матричні світлодіоди

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		



За допомогою цього типу світлодіодів, можна реалізувати широкий спектр завдань.

### 1.3 Постановка задачі та аналіз дерева рішень

В даному дипломному проєкті поставлена задача розробки HDL– моделі роботи панелі LED освітлення.

Дана компонента розробляється в середовищі Active-HDL 8.1. Програма дозволяє проєктувати пристрій за допомогою мов опису апаратури, а також за допомогою структурних схем. Спочатку програма підтримувала тільки мову VHDL, але з часом додалася підтримка мов Verilog і SystemC. За допомогою програми можна графічно проєктувати кінцеві автомати, а також конвертувати HDL опис в графічні структурні схеми і навпаки [4].

Led панелі освітлення можна на практиці використати, наприклад, в автомобілях або рекламі (рисунок 1.10). Тому застосувати цю розробку можна в задачах даного типу.

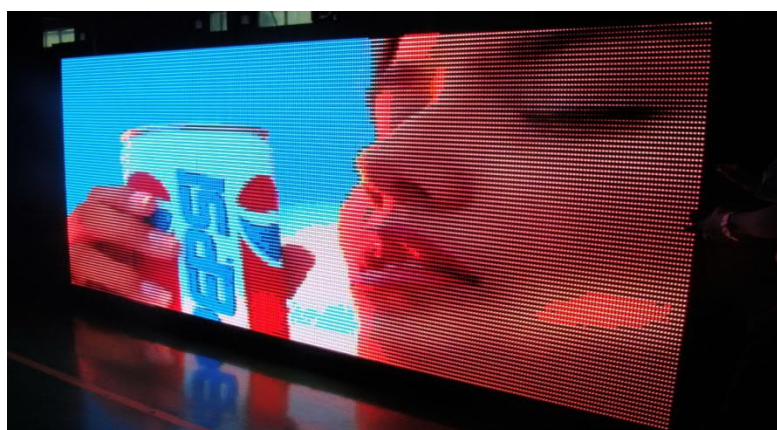


Рисунок 1.10 – Led реклама

Спочатку для більшого розуміння опишемо подане дерево рішень дипломного проєктування зображеного на рисунку 1.11. Першим кроком є –

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

загальний аналіз проблеми освітлення. Наступне - це вибір LED компонентів. Третім кроком дипломного проектування є завдання побудувати структуровану схему LED панелі освітлення. Після цього вибираємо відповідну систему автоматизованого проектування. П'ятим кроком є створення VHDL- моделі LED панелі освітлення. Заключним етапом, зображеним на дереві рішень дипломного проектування є перевірка правильності роботи поданої VHDL-моделі.



Рисунок 1.11 – Дерево рішень дипломного проектування

Процес проектування детально описаний в даній бакалаврській роботі.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		



## 2 ПРОЕКТУВАННЯ ПРИСТРОЮ

### 2.1 Опис середовища Active-HDL

Для розробки і реалізації дипломного проекту, потрібне сучасне, функціональне середовище розробки, доцільно буде використати систему автоматизованого проектування.

Яскравий представник таких САПР нового покоління це інтегроване середовище проектування Active-HDL. Її вирізняє особлива легкість і, якщо можна так висловитися, елегантність в роботі. Фірма ALDEC, творець цього програмного продукту, рік від року удосконалює своє дітище і наділяє його новими можливостями. Пропонована версія даного пакету по численних опитувань користувачів і фахівців займає лідируючі позиції в світі [5].

Active-HDL — середовище розробки, моделювання та верифікації проектів для програмованих логічних інтегральних схем, розроблене фірмою Aldec. Перша версія програмного забезпечення вийшла в 1997 році.

Програма дозволяє створювати описи пристроїв за допомогою мов опису апаратури, а також за допомогою структурних схем. Спочатку програма підтримувала тільки мову VHDL, але з часом додалася підтримка мов Verilog і SystemC [6].

Програма також дозволяє створювати графічні моделі кінцевих автоматів та конвертувати HDL опис в графічні структурні схеми і назад.

Програма забезпечена потужним ядром моделювання. Підтримується спільна робота з програмами MATLAB і Simulink.

Active-HDL базується на платформі Windows, інтегрована FPGA Створення дизайну та рішення моделювання для командних середовищ [7]. Інтегроване середовище проектування Active-HDL (в IDE) включає в себе повний HDL і графічний набір інструментів дизайну і RTL рівень змішаної мови, тренажер для швидкого розгортання і верифікації FPGA проектів. Схематичне зображення процесу створення або редагування проектів, зображено на рисунку 2.1.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

## FPGA Design Flow

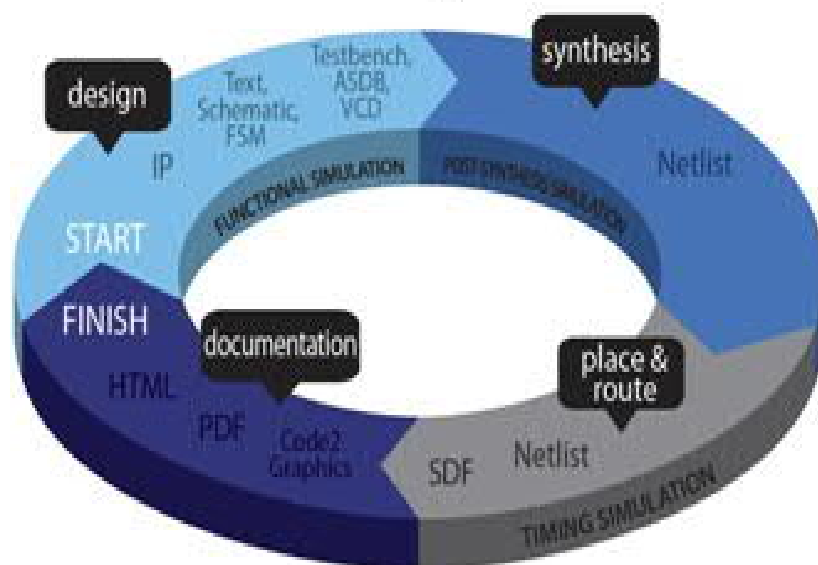


Рисунок 2.1 – Послідовність проектування Active-HDL

Менеджер по дизайну потім викликає понад 120 EDA і FPGA інструментів, під час введення дизайну, моделювання, синтезу і реалізації потоків і дозволяє командам залишатися в межах однієї загальної платформи протягом усього процесу розробки FPGA. Active-HDL підтримує промисловість провідних FPGA пристроїв від Altera®, Atmel®, Lattice®, Microsemi™ (Actel), Quicklogic®, Xilinx® і багато чого іншого [8].

Основні можливості і переваги управління проектом:

- об'єднана команда на основі управління Дизайном підтримує однорідність по локальним або віддаленим командам;
- настроюється FPGA / Менеджер інтерфейсів EDA Flow з понад 120 постачальниками інструментів, дозволяє командам залишатися на одній платформі, на протязі розвитку FPGA [9];
- швидке розгортання проектів з використанням тексту, схеми і State Machine;
- розподілити IP-адресу, використовуючи більш безпечний і надійний стандарт шифрування функціонально сумісного моделювання та налагодження;

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

- потужне загальне ядро змішаного симулятора мови, який підтримує VHDL, Verilog, SystemVerilog (Design) і SystemC;
- забезпечення якості і надійності коду, використання графічного інтерактивного налагодження коду;
- виконання метрик: приводяться перевірки, щоб визначити нетреновані частини вашого дизайну з використанням інструментів аналізу покриття коду;
- поліпшення якості перевірки і знаходження більшої кількості помилок за допомогою ABV - твердження на основі верифікації (SVA, PSL, OVA);
- підключення розриву між моделюванням HDL і навколишнім середовищем математичного моделювання високого рівня для блоків DSP, використовуючи інтерфейс MATLAB / Simulink;
- абстрактний зрозумілий дизайн і представлення в легко зрозумілу графічну форму, використовуючи HDL до схематичного перетворювача;
- частка швидко конструкції з автоматичної генерації проектної документації в HTML і PDF.

Середовище Active-HDL має дружній інтерфейс та багато засобів моделювання. Крім того, відображення дерева проекту дозволяє легко переключатися між файлами та здійснювати моделювання і симуляцію проекту (рисунок 2.2).

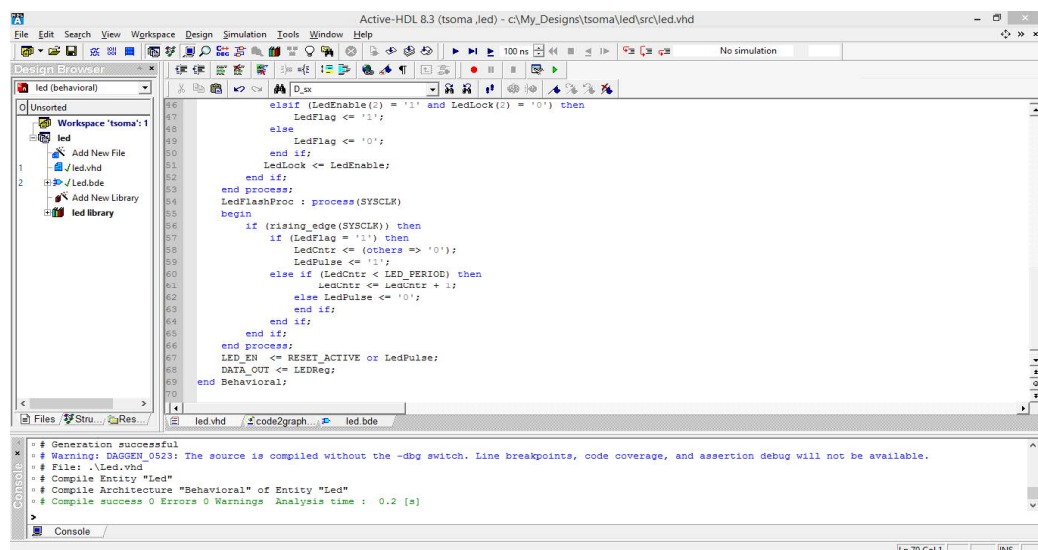


Рисунок 2.2 – Загальний вигляд робочої області середовища Active-HDL

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2.2 Особливості розробленої моделі

Сучасні системи освітлення та індикації використовують різноманітні технології їх реалізації. Одним із широко розповсюджених типів освітлення є використання LED компонентів. Проте використання стандартних засобів не завжди забезпечує необхідні параметри обладнання.

Тому розробка сучасних засобів освітлення з різноманітними функціями є важливою та актуальною задачею.

HDL-MDLED0605.432 – 6-канальний 5А інтелектуальний світлодіодний модуль, він може бути використаний для керування світлодіодами, які мають постійну напругу живлення. Він має 6 вихідних каналів, кожен канал має кнопку обходу для ручного керування [10].

До 6 окремих областей, максимум 12 сценаріїв у кожній області. Кожна область має 1 послідовність кожна послідовність має 12 кроків, час роботи для кожного кроку становить 60 хвилин послідовність має 4 режими роботи: 'вперед', 'назад', 'вперед і назад', 'випадковий', низький, високий, максимальний поріг, для кожного каналу різне навантаження [11].

Кнопка обходу для ручного керування, доступна для кожного каналу, це додаткове налаштування вибору призначеного сценарію або повернення до попереднього сценарію після включення живлення.

Функції присторою:

- захист від короткого замикання і високого струму;
- максимальний струм в кожному каналі: 5А;
- вихід постійної напруги PWM;
- підтримка легкого програмування;
- підтримка онлайн-оновлення HDL-Buspro.

Електричні параметри:

- робоча напруга шини DC15 ~ 30V;
- споживання електроенергії шиною 40mA / DC24V;

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

- світлодіодний вихідний канал 6СН / 5А;
- світлодіодна вхідна напруга DC12 ~ 30V.

Умови навколишнього середовища:

- робоча температура  $-5\text{ C}^0 \sim 45\text{ C}^0$ ;
- робоча відносна вологість до 90%;
- температура зберігання  $-20\text{ C}^0 \sim + 60\text{ C}^0$ ;
- відносна вологість зберігання до 93%.
- затверджено CE, RoHS.

Інформація про продукт:

- розміри  $144 \times 90 \times 66$  (мм);
- вага нетто 351г;
- матеріал корпусу Нейлон, ПК;
- монтаж 35 мм монтаж DIN-рейки;
- ступінь захисту IP20.

На рисунку 2.3 зображено пристрій і його органи керування.

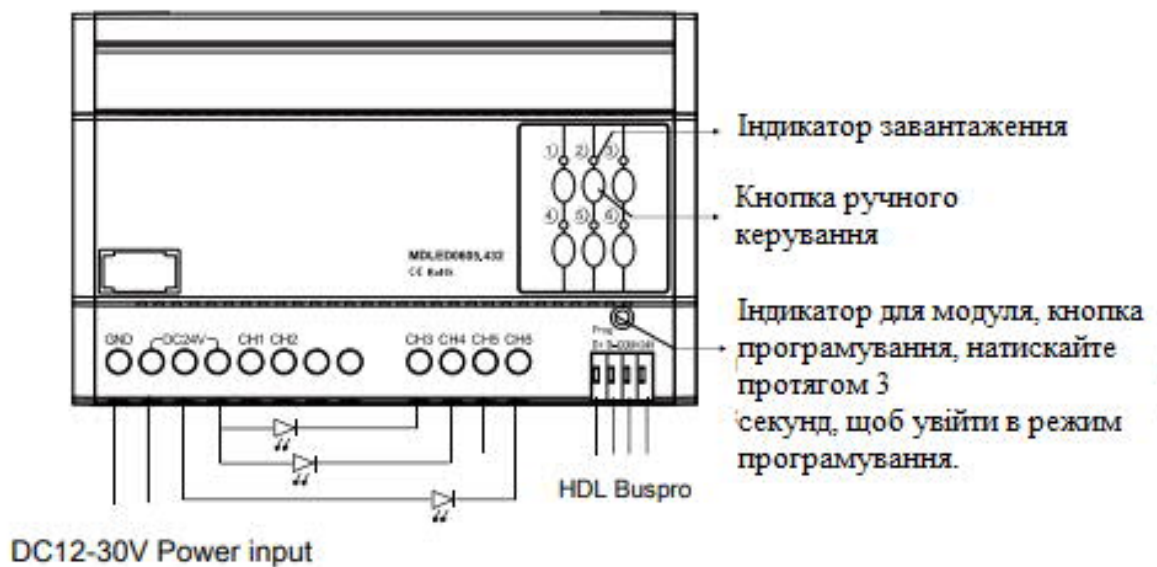


Рисунок 2.3 – Загальний вигляд і опис органів керування

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

Таким чином, даний пристрій являється універсальним засобом керування світлодіодними модулями, має широку палітру налаштувань і, відповідно, застосувань.

### 2.3 Розробка структурної схеми

Для розробки пристрою, спочатку потрібно описати його роботу, для цього використано програмний опис, за допомогою якого здійснюється управління окремих елементів і пристрою в цілому. Отже, програмний опис, складений в додатку Active-HDL, відображений в додатку А.

Історично розвиток засобів проектування ПЛІС відбувався на основі трьох парадигм:

- логічного проектування;
- схемного проектування;
- проектування за допомогою мов опису апаратного забезпечення.

Остання концепція є найновішою і переживає зараз бурхливий розвиток. Ідея даного підходу полягає в тому, що розробник описує необхідну поведінку цифрового пристрою за допомогою поведінкової моделі на функціональному рівні, а система автоматизованого проектування синтезує логічну структуру, що відтворює описану поведінку. Поведінкові моделі можуть бути створені різними способами, наприклад, за допомогою спеціалізованих мов опису апаратного забезпечення, графів скінченних автоматів та ін. Основна увага приділяється методам синтезу на основі мови опису апаратного забезпечення VHDL. У процесі поведінкового проектування цифрових пристроїв будується цілий ряд моделей різного типу та різного рівня абстракції. На рисунку 2.4 зображено один з можливих варіантів побудови процесу проектування цифрового пристрою на базі ПЛІС від формулювання проектного завдання до кінцевого результату – апаратно реалізованого зразка.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

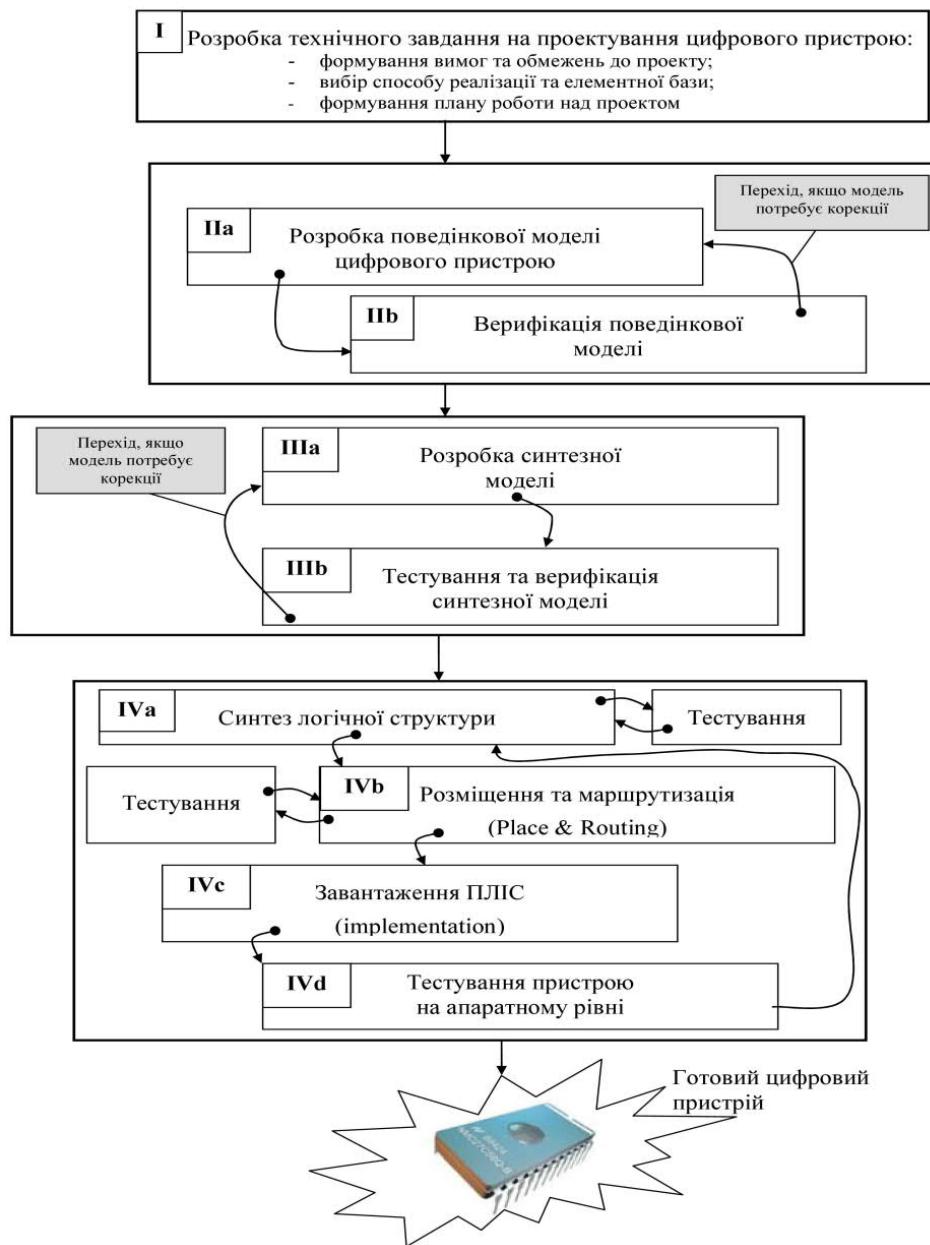


Рисунок 2.4 - Процес проектування індикатора

У процесі поведінкового проектування на всіх його етапах створюється ряд моделей різних аспектів роботи цифрових пристроїв та різних рівнів абстракції. Причому процес проектування полягає в переході від моделей з вищим рівнем абстракції до моделей з нижчим рівнем абстракції, тобто поступовій деталізації описання цифрового пристрою. Наведемо класифікацію моделей, що застосовуються в процесі проектування.

Поведінкова модель (Behavioral model, Interpreted model) показує реакцію моделі на зміну вхідних сигналів з урахуванням часу реакції, але не містить детального опису апаратної реалізації пристрою, що моделюється.

Рівень абстракції залежить від детальності опису моделі. Наприклад, поведінкова модель може описувати процесор, що виконує абстрактний алгоритм, або це може бути модель процесора на нижчому рівні абстракції – рівні множини інструкцій. Точність деталізації вхідних і вихідних даних залежить від рівня абстракції моделі.

Функціональна модель (Functional model) описує функції системи без вказання способу реалізації цих функцій. Дана модель показує лише реакцію системи чи її компонента без урахування часового фактора (визначає значення виходу, але не час його встановлення). Рівень абстракції залежить від ступеня деталізації моделі. Рівень деталізації вхідних та вихідних даних залежить від рівня абстракції.

Структурна модель (Structural model) представляє компоненти чи системи з точки зору взаємозв'язків між підкомпонентами, що їх утворюють. Структурна модель має відповідати фізичній ієрархії в описуваному об'єкті. Ієрархія, в свою чергу, визначається фізичною організацією конкретної реалізації. Структурна модель описує фізичну структуру конкретної реалізації шляхом визначення компонентів та топології їх взаємозв'язків.

Компоненти можуть бути описані на структурному, функціональному чи поведінковому рівнях. Моделювання структурної моделі вимагає наявності поведінкових моделей всіх нижчих гілок ієрархії, отже, ступінь деталізації модельного часу, значень об'єктів даних та функціональності моделі залежить від ступеня деталізації моделей компонентів.

Модель продуктивності (Performance Model, Uninterpreted Model) дозволяє визначити лише часові аспекти роботи цифрового пристрою, що моделюється (тобто швидкість реакції на зміну вхідного сигналу, але не значення вхідного сигналу). Модель інтерфейсу (Interface Model, bus functional) називають ще чорним ящиком. Може містити деталізацію по всіх аспектах процесів обміну

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		



інформацією між об'єктом та зовнішнім середовищем, включаючи функціональність, часові характеристики, значення даних тощо. Така модель не містить інформації про внутрішню структуру об'єкта.

Модель змішаного рівня, гібридна модель (Mixed-Level Model, Hybrid Model) – модель, що містить компоненти, описані на різних рівнях абстракції або різними класами моделей. Віртуальний прототип (Virtual Prototype) – це комп'ютерна імітаційна модель кінцевого продукту, компонента чи системи. При цьому від такої моделі не вимагається дотримання жодних спеціальних умов щодо її характеристик. Термін “віртуальний прототип” позначає клас моделей, що відіграють певну роль у процесі проектування, зокрема:

- показує можливі варіанти реалізації проекту;
- демонструє концепцію проекту;
- дає можливість перевірки проекту на відповідність вимогам та адекватності поставленій задачі.

Повернемося до процесу проектування, наведеному на рисунку 2.4. Весь процес розбито на 4 головні етапи.

На першому етапі формується технічне завдання для проектування цифрового пристрою. На цьому етапі формуються інтерфейсна, продуктивна та функціональна моделі, із застосуванням яких можна постійно перевіряти моделі, сформовані на наступних етапах на відповідність технічному завданню. Так, продуктивна модель визначає швидкодію проектованого пристрою, інтерфейсна – спосіб його інтеграції до вищого ієрархічного рівня, а функціональна модель визначає алгоритм перетворення інформації в проектованому пристрої.

На другому етапі формується поведінкова модель. У VHDL під поведінковою розуміється модель, написана із застосуванням всіх існуючих в цій мові конструкцій та типів даних, наприклад, дійсних чисел, файлів, вказівників динамічної пам'яті тощо. Поведінкова модель, розроблена на другому етапі, повинна повністю відповідати вимогам, сформованих на першому етапі. Саме це й перевіряється на підетапі Пв.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

Після формування остаточного вигляду поведінкової моделі розробник переходить до етапу III – створення синтезної моделі. Синтезна модель також належить до класу поведінкових, однак може бути написана лише за допомогою певної підмножини конструкцій мови VHDL, які підтримуються засобами синтезу логічної структури. На даному етапі розвитку засоби синтезу (перетворення VHDL- програми на схему логічних елементів) підтримують не всі можливі у VHDL мовні конструкції, наприклад, не підтримуються операції з дійсними числами чи вказівники. Такі конструкції, як, наприклад, файли, із зрозумілих причин відносяться до природно-несинтезованих. Перехід від поведінкової моделі до синтезованої характеризується зниженням рівня абстракції описання пристрою.

На четвертому етапі здійснюється перехід від синтезної моделі до логічної структури та бітового потоку, який завантажується безпосередньо в ПЛІС. Підетап IVa представляє собою перехід від синтезної моделі до моделі рівня регістрових передач (Register Transfer Level Model – RTL-model). Така модель може включати лише стандартні компоненти цифрових пристроїв, наприклад, регістри, лічильники, дешифратори, тригери тощо. На підетапі IVb RTL-модель розміщується на ПЛІС і забезпечується зв'язок між окремими компонентами. На підетапі IVc формується бітовий потік та завантажується до ПЛІС, після чого проводиться остаточне тестування вже реалізованого апаратно-цифрового пристрою. На всіх етапах тестування як генератор еталонних вихідних значень пристрою використовується поведінкова модель, розроблена на другому етапі.

Запропонований індикатор отримує сигнал з одного з чотирьох входів для включення чи вимкнення одного з двох виходів. Даний індикатор може застосовуватися в передній панелі автомобіля.

Відносно готового опису, розробляється структурна схема, на якій описані всі входи-виходи і функціональні блоки.

На рисунку 2.5 подано структурну схему розробленого проекту, що моделюється та симулюється в середовищі Active-HDL для подальшого проектування на ПЛІС чи ПЛМ.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

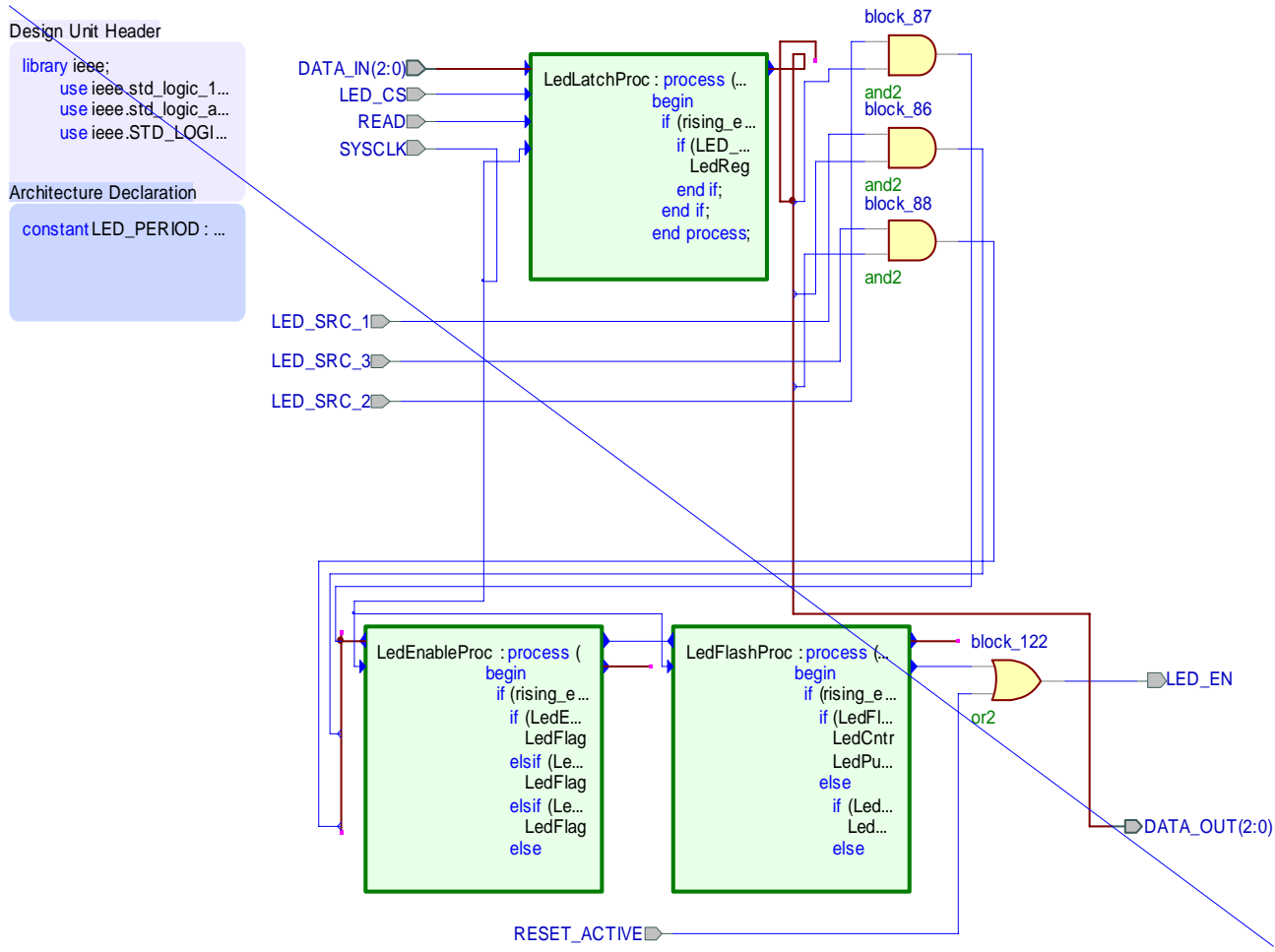


Рисунок 2.5 - Структурна схема пристрою

Отже, за описом та структурною схемою пристрою, можна побачити основні функціональні блоки і елементи керування.

## 3 ДОСЛІДЖЕННЯ РОБОТИ РОЗРОБЛЕНОЇ HDL-МОДЕЛІ

### 3.1 Розробка моделі індикатора LED-панелі

VHDL — мова опису апаратних інтегральних схем. Мова проектування VHDL є базовою мовою при розробці апаратури сучасних обчислювальних систем.

Мова VHDL створена як засіб опису цифрових систем, однак існує підмножина мови — VHDL AMS (аналогових та змішаних сигналів), що дозволяє описувати як чисто аналогові, так і змішані, цифро-аналогові схеми.

Active-HDL заснована на платформі Windows, інтегрована FPGA для створення дизайну і моделювання та вирішення в командних середовищах. Інтегроване середовище проектування Active-HDL (в IDE) включає в себе повний HDL і графічний дизайн, набір інструментів і RTL рівня змішаної мови для швидкого розгортання і верифікації FPGA конструкцій.

Менеджер по дизайну потоку включає 120 EDA і FPGA інструментів, під час введення дизайну, моделювання, синтезу і реалізації потоків і дозволяє командам залишатися в межах однієї спільної платформи протягом усього процесу розробки FPGA. Active-HDL підтримує провідні в галузі FPGA пристроїв від Altera, Atmel, Lattice, Microsemi (Actel), Quicklogic, Xilinx і багато іншого.

Основні можливості і переваги в управлінні проектами:

- об'єднана команда на основі управління дизайну підтримує однорідність для локальних або віддалених команд;
- настроюється FPGA / Менеджер інтерфейси EDA потоку з 120 постачальниками інструментів дозволяє командам залишатися на одній платформі протягом всього розвитку FPGA.

В створенні графічного дизайну:

- швидке розгортання конструкцій за допомогою тексту, схематичного зображення і State Machine;

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		

- розподілити або доставити IP-адреси, використовуючи більш безпечний і надійний стандарт шифрування інтероперабельності.

Моделювання та налагодження:

- потужне загальне ядро мішаного симулятора мову, який підтримує VHDL, Verilog, SystemVerilog (дизайн) і SystemC;

- забезпечення якості програмного коду і надійність за допомогою графічно інтерактивного налагодження коду та інструментів якості;

- поліпшення якості перевірки і виявлення помилок за допомогою ABV-твердження на основі верифікації (SVA, PSL, OVA);

- з'єднує розрив між моделювання HDL і навколишнім середовищем математичного моделювання високого рівня для блоків DSP з використанням інтерфейсу MATLAB / Simulink.

Документація HTML / PDF:

- анотація інтелекту дизайну і представлення його в легко зрозумілій графічній формі, використовуючи HDL до схематичного перетворювача;

- швидке конструювання з автоматичної генерації проектної документації в форматі HTML і PDF.

Розроблений індикатор складається з таких основних компонентів:

– світлодіодний перемикач LedLatchProc;

– компонента вмикання LedEnableProc;

– компонент спалаху діода LedFlashProc.

Далі розглянемо кожен компонент детальніше.

Компоненти мають наступний ряд входів, які виконують відповідні функції:

– SYSCLK – системний годинник, що використовується в процесі симуляції та подільшій роботі моделі;

– LED\_CS – змінна вводу-виводу інформації;

– READ читає чи записує змінну N коли CS=1;

– RESET\_ACTIVE – здійснює включення індикатору;

– LED\_SRC\_1 - один короткий спалах першого діода, коли цей сигнал зростає;

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						33
Зм.	Арк.	№ докум.	Підпис	Дата		

–LED\_SRC\_2 - один короткий спалах другого діода, коли цей сигнал зростає;

–LED\_SRC\_3 - один короткий спалах третього діода, коли цей сигнал зростає;

–DATA\_IN – позначення активного індикатора.

Та відповідний перелік виходів які виконують наступні функцій:

–DATA\_OUT – перезагрузка компоненти;

–LED\_EN – вивід на світлодіодну панель кінцевого сигналу.

Для проектування приймача використані наступні бібліотеки стандарту IEEE: std\_logic\_1164.all, std\_logic\_arith.all, std\_logic\_unsigned.all. Повний HDL – код розробленої моделі знаходиться у додатку А .

В результаті було розроблено всі необхідні компоненти для розробки запропонованого світлодіодного індикатора.

Інтерфейсний опис проекту має наступний вигляд:

```
entity LedLatchV40 is
  Port ( SYSCLK      : in  std_logic;
         LED_CS      : in  std_logic;
         READ        : in  std_logic;
         RESET_ACTIVE : in  std_logic;
         LED_SRC_1   : in  std_logic;
         LED_SRC_2   : in  std_logic;
         LED_SRC_3   : in  std_logic;
         DATA_IN    : in  std_logic_vector(2 downto 0);
         DATA_OUT   : out std_logic_vector(2 downto 0);
         LED_EN      : out std_logic
  );
end LedLatchV40;
```

Компонента перемикання індикатора LedLatchProc спроектована як процес і має наступний опис:

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

```

LedLatchProc : process(SYSCLK)
begin
    if (rising_edge(SYSCLK)) then
        if (LED_CS = '1' and READ = '0') then
            LedReg <= DATA_IN;
        end if;
    end if;
end process;

```

Компонента вмикання LedEnableProc спроектована наступним чином:

```

LedEnableProc : process(SYSCLK)
begin
    if (rising_edge(SYSCLK)) then
        if (LedEnable(0) = '1' and LedLock(0) = '0') then
            LedFlag <= '1';
        elsif (LedEnable(1) = '1' and LedLock(1) = '0') then
            LedFlag <= '1';
        elsif (LedEnable(2) = '1' and LedLock(2) = '0') then
            LedFlag <= '1';
        else
            LedFlag <= '0';
        end if;
        LedLock <= LedEnable;
    end if;
end process;

```

Нарешті, компонент спалаху діода LedFlashProc має наступний опис:

```

LedFlashProc : process(SYSCLK)
begin
    if (rising_edge(SYSCLK)) then
        if (LedFlag = '1') then
            LedCntr <= (others => '0');
        end if;
    end if;
end process;

```

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						35
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        LedPulse <= '1';
    else
        if (LedCntr < LED_PERIOD) then
            LedCntr <= LedCntr + 1;
        else
            LedPulse <= '0';
        end if;
    end if;
end if;
end process;

```

Крім того, для правильної роботи проекту застосовуються наступні сигнали:

```

constant LED_PERIOD:std_logic_vector(19 downto 0):= x"F4240";
signal LedReg: std_logic_vector(2 downto 0);
signal LedEnable: std_logic_vector(2 downto 0);
signal LedLock: std_logic_vector(2 downto 0);
signal LedCntr: std_logic_vector(19 downto 0);
signal LedFlag, LedPulse: std_logic;

```

Вказані сигнали працюють наступним чином:

```

LedEnable(0) <= LedReg(0) and LED_SRC_1;
LedEnable(1) <= LedReg(1) and LED_SRC_2;
LedEnable(2) <= LedReg(2) and LED_SRC_3;

```

### 3.2 Функціональна симуляція проекту

Використовуючи вбудований у пакет Active HDL засіб Test Bench проведено функціональну верифікацію проекту – робота симетричного блокового алгоритму шифрування IDEA.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		



Test Bench (випробувальний стенд) призначений для перевірки правильності роботи проєктованого пристрою (компоненти). Тестована компонента перевіряється за допомогою стимулів (вхідним сигналам та шинам даних присвоюється необхідне для роботи пристрою значення) і при цьому контролюється реакція даної компоненти шляхом нагляду та дослідження (зондування) вихідних сигналів.

Саме пакет Active HDL дозволяє в деякій мірі автоматизувати розробку випробувального стенду.

Для того, щоб використати згадану автоматизацію необхідно виконати наступну послідовність дій:

- в меню Tools вибрати команду „Generate Test Bench...”;
- необхідно вказати до якого інтерфейсного та архітектурного описів буде генеруватися даний випробувальний стенд та вибрати тип випробувального стенду (Single Process або Waves Based);
- можна додати тестові вектори із зовнішнього файлу (вказати шлях та ім'я даного файлу);
- вибрати ім'я інтерфейсного опису даного випробувального стенду та ім'я папки, де буде зберігатися даний випробувальний стенд.

Саме автоматично згенерований випробувальний стенд складається з наступних частин:

- Add your library and packages declaration – декларація бібліотек та пакетів, необхідних для даного проєкту;
- Component declaration of the tested unit – декларація компонент саме тої компоненти, що тестується;
- Stimulus signals - signals mapped to the input and inout ports of tested entity – визначення сигналів, до яких можна подавати стимули (сигнали тестованої компоненти);
- Observed signals - signals mapped to the output ports of tested entity – сигнали, за якими необхідно наглядати (вихідні сигнали компоненти);

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

- Add your code – частина випробувального стенду, куди можна ввести додатковий код (звичайно, якщо в цьому є необхідність);

- Unit Under Test port map – визначення портів тестованої компоненти;

- Add your stimulus – місце для введення стимулів на порти компоненти, що досліджується.

Testbench виконує три основні операції:

- операція вхідного інтерфейсу;

- операція вихідного інтерфейсу;

- операція інтерфейсу занесення ключа в пам'ять.

Підключення саме цих бібліотек дозволить користуватися функціями читання та запису у файл, а саме використано для запису отриманих поточних результатів роботи (отриманих на вихідних сигналах та шинах) у текстовий файл під час симуляції проекту.

Також додатково введено змінну „clock”, що має тип даних „time” і рівна 10 ns. Змінна введена для того, щоб користувачу спростити процедуру вчасного визначення подачі стимулів на порти тестованої компоненти.

Як було згадано раніше, мова VHDL дозволяє описувати процеси, що виконуються паралельно (у часі). Така особливість була використана при написанні випробувального стенду і якості подачі стимулів на входи досліджуваної компоненти.

Із часових діаграм, отриманих у пакеті Active HDL при симуляції роботи компоненти, було обрано час завантаження і частина часу роботи (генерування ПВЧ) компоненти.

На рисунку 3.1 подано часову діаграму роботи пристрою.

Для здійснення процесу симуляції початково задані відповідні початкові стимули вхідних сигналів і здійснено аналіз результатів моделювання через перевірку правильності вихідних даних.

Дяна часова діаграма підтверджує правильність роботи розробленого проекту.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

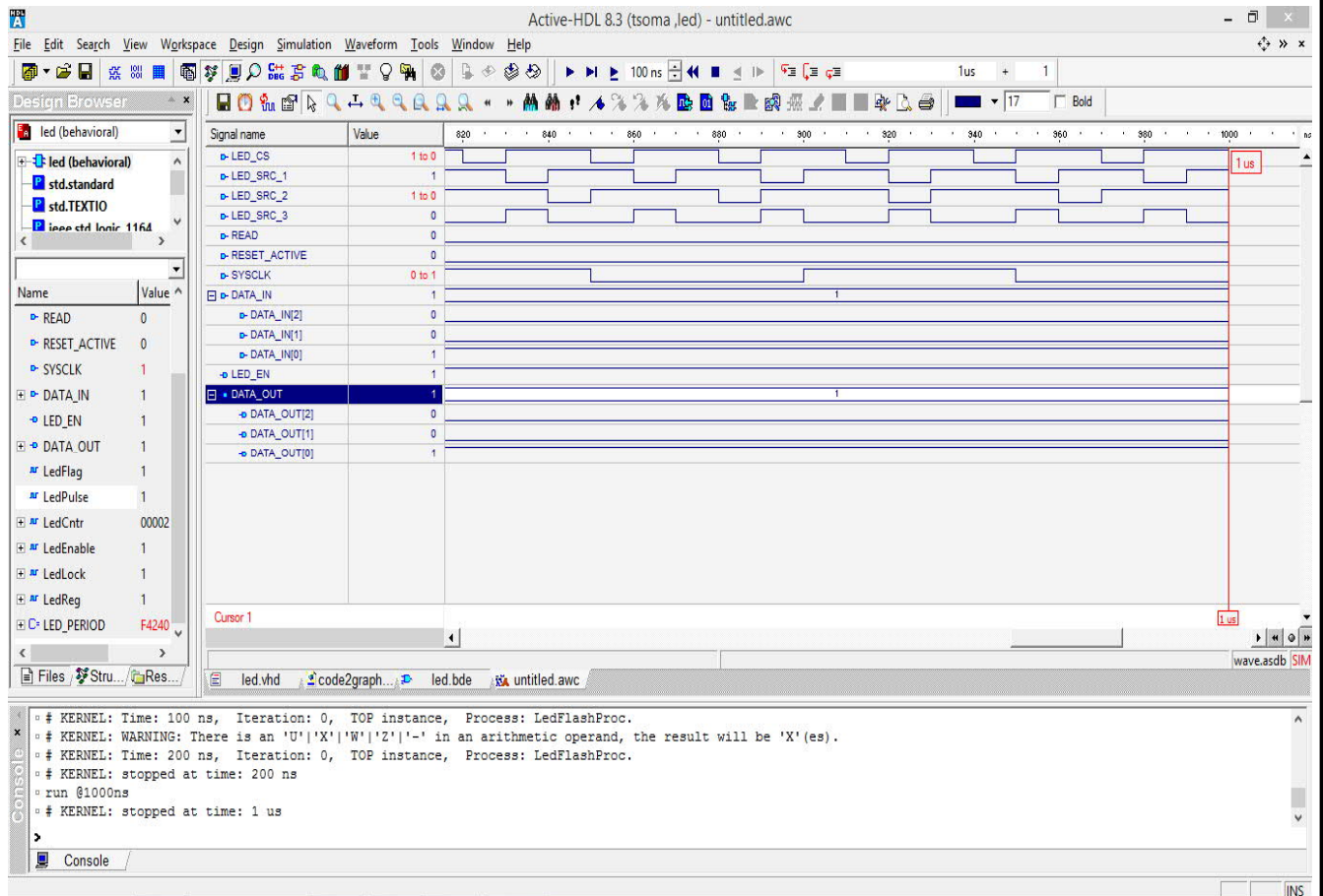


Рисунок 3.1 – Часова діаграма роботи проекту

### 3.3 Верифікація проекту

Послідовність проектування пристрою на функціональному рівні можна поділити на такі етапи: Створення файлів проекту: файли опису пристрою, тестових файлів, макросів. Компіляція файлів проекту. Симуляція на функціональному рівні.

Active-HDL дозволяє створювати та редагувати різні компоненти такі як файли опису пристрою, використані в проекті бібліотеки, допоміжні файли, сукупність яких називається проектом (Design). Файли проекту за замовчуванням зберігаються в єдиній директорії, що носить назву проекту.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						39
Зм.	Арк.	№ докум.	Підпис	Дата		



Дані, подані на рисунку 3.2 підтверджує працездатність розробленої моделі. Тому розроблений проект можна використати для програмування ПЛІС чи ПЛМ, що може використатись в панелі світло діодів автомобіля.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

## 4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ

В цьому розділі бакалаврської роботи (БР) проводиться економічне обґрунтування доцільності розробки програмного забезпечення. Зокрема, здійснюється розрахунок витрат на розробку програмного забезпечення, експлуатаційних витрат, ціни споживання програмного забезпечення. В заключній частині визначаються показники економічної ефективності нового програмного продукту, обґрунтовуються відповідні висновки.

### 4.1 Розрахунок витрат на розробку програмного забезпечення

Витрати на розробку і впровадження програмних засобів ( $K$ ) включають:

$$K = K_1 + K_2 \quad (4.1)$$

де  $K_1$  - витрати на розробку програмних засобів, грн;

$K_2$  - витрати на відлагодження і дослідну експлуатацію програми рішення задачі на комп'ютері, грн.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробників ( $B_{оп}$ );
- витрати на відрахування у спеціальні державні фонди ( $B_{ф}$ );
- витрати на покупні вироби ( $Пв$ );
- витрати на придбання спецобладнання для проведення експериментальних робіт ( $Об$ );
- накладні витрати ( $H$ );
- інші витрати ( $Iв$ ).

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

#### 4.1.1 Розрахунок витрат на оплату праці

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудомісткості відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці програмного забезпечення задіяні наступні спеціалісти - розробники, а саме – керівник проекту, студент-дипломник, консультант техніко-економічного розділу.

Таблиця 4.1 - Вихідні дані для розрахунку витрат на оплату праці

Посада виконавців	Місячний оклад, грн
Керівник ВКР, доцент	5286
Консультант техніко-економічного розділу, доцент	6086
Студент	1330

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{оп} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij} \quad (4.2)$$

де  $n_{ij}$  – чисельність розробників  $i$ -ої спеціальності  $j$ -го тарифного розряду, осіб;

$t_{ij}$  – затрачений час на розробку проекту співробітником  $i$ -ої спеціальності  $j$ -го тарифного розряду, год;

$C_{ij}$  – годинна ставка працівника  $i$ -ої спеціальності  $j$ -го тарифного розряду, грн.

Середньо годинна ставка працівника може бути розрахована за формулою:

$$C_{ij} = \frac{C_{ij}^0(1+h)}{PЧ_i}, \quad (4.3)$$

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

де  $C_{ij}$  – основна місячна заробітна плата розробника  $i$ -ої спеціальності  $j$ -го тарифного розряду, грн;

$h$  – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$РЧ_i$  - місячний фонд робочого часу працівника  $i$ -ої спеціальності  $j$ -го тарифного розряду, год (приймаємо 168 год).

Результати розрахунку записують до таблиці 4.2.

Таблиця 4.2 - Розрахунок витрат на оплату праці

Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год	Витрати на розробку, грн
Керівник ВКР, доцент	16	56,6	905,6
Консультант техніко-економічного розділу, доцент	2	76	152
Студент	100	7,9	790
Разом	1847,6		

#### 4.1.2 Відрахування на соціальні заходи

Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5 % від суми заробітної плати:

$$B_{\phi} = \frac{20,5}{100} \cdot 1847,6 = 378,76 \text{ грн.} \quad (4.4)$$

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		



#### 4.1.3 Розрахунок витрат на матеріали та комплектуючі

У таблиці 4.3 наведений перелік купованих виробів і розраховані витрати на них.

Таблиця 4.3 – Розрахунок витрат на матеріали та комплектуючі

Найменування купованих виробів	Одиниця виміру	Ціна, грн	Кількість купованих виробів	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
Папір (формат А4)	уп	90,0	1	90,00	9,0	99,0
Ручка кулькова	шт	20,0	1	20,00	2,00	22,0
Олівець простий	шт	4,0	1	4,00	0,4	4,40
Зошит, 18 арк	шт	6,0	1	6,00	0,6	6,60
Тонер для принтера	уп	50	1	50	5,0	55,0
Разом						187

#### 4.1.4 Витрати на використання комп'ютерної техніки

Витрати на використання комп'ютерної техніки включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним забезпеченням, витрати на електроенергію, що споживається комп'ютером. За даними обчислювального центру ТНЕУ для комп'ютера типу IBM PC/ATX вартість години роботи становить 5,2 грн. Середній щоденний час роботи на

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

комп'ютері – 2 години. Розрахунок витрат на використання комп'ютерної техніки приведений в таблиці 4.4.

Таблиця 4.4- Розрахунок витрат на використання комп'ютерної техніки

Назва етапів робіт, при виконанні яких використовується комп'ютер	Час використання комп'ютера, год.	Витрати на використання комп'ютера, грн.
Проведення досліджень та оформлення їх результатів	60	312
Оформлення техніко-економічного розділу	8	41,6
Оформлення ВКР	12	62,4
Разом	80	416

#### 4.1.5 Накладні витрати

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H = 1,5 \cdot 1847,6 = 2771,4 \text{ (грн.)} \quad (4.5)$$

#### 4.1.6 Інші витрати

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

$$I = 1847,6 \cdot 0,1 = 184,76 \text{ (грн)} \quad (4.6)$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{OP} + B_{\Phi} + B_{PB} + H + I \quad (4.7)$$

$$K_1 = 1847,6 + 378,76 + 187 + 2771,4 + 184,76 = 5369,86 \text{ (грн)} \quad (4.8)$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою:

$$K_2 = S_{m.g.} \cdot t_{vid} \quad (4.9)$$

де  $S_{m.g.}$  - вартість однієї машино-години роботи ПК, грн/год.;

$t_{vid}$  - комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 40 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 5,2 грн. Тому:

$$K_2 = 5,2 \cdot 80 = 416 \text{ грн.} \quad (4.10)$$

На основі отриманих даних складаємо кошторис витрат на розробку програмного забезпечення (таблиця 2.5).

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.5 – Кошторис витрат на розробку програмного забезпечення

Найменування витрат	Сума витрат, грн
Витрати на оплату праці	1847,6
Відрахування у спеціальні державні фонди	378,76
Витрати на куповані вироби	187
Накладні витрати	2771,4
Інші витрати	184,76
Витрати на відлагодження і дослідну експлуатацію програмного продукту	416
Разом	5785,52

#### 4.2 Визначення експлуатаційних витрат

Для оцінки економічної ефективності розроблюваного програмного продукту слід порівняти його з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

Експлуатаційні одноразові витрати по програмному забезпеченню і аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_{\text{п}} = E_{1\text{п}} + E_{2\text{п}} \quad (4.11)$$

де  $E_{\text{п}}$  - одноразові експлуатаційні витрати на ПЗ (аналог), грн.;

$E_{1\text{п}}$  - вартість підготовки даних для експлуатації ПЗ (аналог), грн;

$E_{2\text{п}}$  - вартість роботи комп'ютера для розробки програмного забезпечення (аналог), грн.

Річні експлуатаційні витрати  $B_{\text{еп}}$  визначаються за формулою:

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

$$B_{E\Pi} = E_{\Pi} * N_{\Pi} \quad (4.12)$$

де  $N_{\Pi}$  - періодичність експлуатації ПЗ (аналогу), раз/рік.

Вартість підготовки даних для роботи на комп'ютері визначається за формулою:

$$E_{\Pi\Pi} = \sum_{i=1}^n n_i t_i c_i \quad (4.13)$$

де  $i$  - категорії працівників, які приймають участь у підготовці даних ( $i=1,2,\dots,n$ );

$n_i$  - кількість працівників  $i$ -ої категорії, осіб;

$t_i$  - трудомісткість роботи співробітників  $i$ -ої категорії по підготовці даних, год.;

$c_i$  - середнього годинна ставка працівника  $i$ -ої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення:

$$c_i = \frac{c_i^0 (1+b)}{m} \quad (4.14)$$

де  $c_i^0$  - основна місячна заробітна плата працівника  $i$ -ої категорії, грн;

$b$  - коефіцієнт, який враховує додаткову заробітну плату;

$m$  - кількість робочих годин у місяці, год.

Для роботи з даними як для поточного програмного забезпечення так і аналогу потрібен один працівник, основна місячна заробітна плата якого складає:

$c^0 = 1330$  грн. Тоді:

$$c_1 = \frac{1330(1+0,57)}{22 * 8} = 11,86 \text{ грн/год} \quad (4.15)$$

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

Трудомісткість підготовки даних для програмного забезпечення складає 1 год., для аналога 1,5 год.

Таблиця 4.6 – Розрахунок витрат на підготовку даних та реалізацію програмного забезпечення на комп'ютері

Час роботи співробітників, год	Середньогодинна заробітна плата, грн/год	Витрати, грн
Проектне рішення		
1	11,86	11,86
Аналог		
1,5	11,86	17,79

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2П} = t * S_{МГ} \quad (4.16)$$

де  $t$  - витрати машинного часу для реалізації програмного продукту, год;

$S_{МГ}$  - вартість однієї години роботи комп'ютера, грн/год.

$$E_{2n} = 1 \cdot 5,2 = 5,2 \text{ (грн)}; E_{2a} = 1,5 \cdot 5,2 = 7,8 \text{ (грн)} \quad (4.17)$$

$$E_n = 11,86 + 5,2 = 17,06 \text{ (грн)}; E_a = 17,79 + 7,8 = 25,59 \text{ (грн)} \quad (4.18)$$

$$B_{en} = 17,06 \cdot 252 = 4299,12 \text{ (грн)}; B_{ea} = 25,59 \cdot 252 = 6448,68 \text{ (грн)} \quad (4.19)$$

### 4.3 Розрахунок ціни споживання програмного продукту

Ціна споживання - це витрати на придбання і експлуатацію програмного продукту за весь строк його служби:

$$Ц_{C(\Pi)} = Ц_{\Pi} + B_{(E)NPV} \quad (4.20)$$

де  $Ц_{\Pi}$  - ціна придбання програмного продукту, грн.

$$Ц_{\Pi} = K \left(1 + \frac{\Pi_p}{100}\right) + K_o + K_{\kappa} \quad (4.21)$$

де  $K$  - кошторисна вартість;

$\Pi_p$  - рентабельність;

$K_o$  - витрати на прив'язку та освоєння програмного забезпечення на конкретному об'єкті, грн;

$K_{\kappa}$  - витрати на доукомплектування технічних засобів на об'єкті, грн.

$$Ц_{\Pi} = 578552 \cdot (1 + 0,3) = 75212 \text{ (грн.)} \quad (4.22)$$

Вартість витрат на експлуатацію програмного забезпечення (за весь час його експлуатації), грн:

$$B_{enpv} = \sum_{t=0}^T \frac{B_{e\Pi}}{(1+R)^t} \quad (4.23)$$

де  $B_{en}$  - річні експлуатаційні витрати, грн;

$T$  - термін служби програмного забезпечення, років;

$R$  - річна ставка проценту банку.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

$$B_{епрв} = \sum_{t=1}^5 \frac{4299,12}{(1+0,08)^t} = 17200,15 \text{ (грн)} \quad (4.24)$$

$$B_{епрв} = \sum_{t=1}^5 \frac{6448,68}{(1+0,08)^t} = 25800,2 \text{ (грн)} \quad (4.25)$$

Тоді ціна споживання програмного забезпечення дорівнюватиме:

$$Ц_{сн} = 7521,2 + 17200,15 = 24721,35 \text{ (грн)} \quad (4.26)$$

Аналогічно визначається ціна споживання для аналогу:

$$Ц_{са} = 6500,0 + 25800,2 = 32300,2 \text{ (грн)} \quad (4.27)$$

#### 4.4 Визначення показників економічної ефективності

Економічний ефект в сфері розробки програмного продукту:

$$E_{ПР} = Ц_{П} - Ц_{А} \quad (4.28)$$

$$E_{ПР} = 7521,2 - 6500,0 = 1021,2 \text{ (грн)} \quad (4.29)$$

Річний економічний ефект в сфері експлуатації:

$$E_{КС} = B_{ЕА} - B_{ЕП} \quad (4.30)$$

$$E_{КС} = 6448,68 - 4299,12 = 2149,56 \text{ (грн)} \quad (4.31)$$

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		



Додатковий економічний ефект у сфері експлуатації:

$$\Delta E_{екс} = \sum_{t=1}^T E_{екс} (1+R)^{T-t} \quad (4.32)$$

$$\Delta E_{екс} = \sum_{t=1}^5 2149,56 * (1 + 0,08)^{5-t} = 12595,71 \text{ (грн)} \quad (4.33)$$

Сумарний ефект складає:

$$E = E_{тп} + \Delta E_{екс} = 1021,2 + 12595,71 = 13616,91 \text{ (грн)} \quad (4.34)$$

В даному розділі проведено розрахунок витрат на розробку програмного забезпечення. Здійснено порівняння з існуючим аналогом, і цим показано, що вказане програмне забезпечення має переваги в порівнянні з аналогами, зокрема: надійність, простота використання, зручність. Згідно проведеного економічного обґрунтування зазначене програмного забезпечення є конкурентоздатним. Крім того, отримано економічний ефект у розмірі 13616 грн. і тому розробка і впровадження цього програмного забезпечення є економічно доцільними.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						53
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

В результаті розробки даної бакалаврської роботи:

- 1) розглянуто сучасні системи LED-освітлення;
  - 2) проведено аналіз засобів освітлення, що дозволяє виділити основні характеристики та алгоритми роботи;
  - 3) розглянуто особливості застосування панелі LED-освітлення;
  - 4) для реалізації проекту вибрано сучасну систему автоматизованого проектування Active-HDL, що дозволяє здійснювати проектування будь-якої компоненти комп'ютерних систем;
  - 5) описано модель основних компонентів проекту та описано їх інтерфейс;
  - 6) здійснено симуляцію та верифікацію проекту, що дозволило підтвердити його працездатність;
  - 7) наведено результати виконання програмного комплексу;
  - 8) здійснено техніко-економічне обґрунтування проекту.
- Результати розробки мають апробацію на конференції (додаток Б) та практичне використання (додаток В).

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Даткич: веб-сайт. URL: <http://l2u.su/qcui> (дата звернення: 12.02.19).
2. Бурштинський М.В., Хай М.В., Харчишин М.Б. Давачі: навчальний посібник. Львів, 2014. 198 с.
3. Гаврилюк М.О. Давачі систем автоматики: навчальний посібник. Львів, 2010. 180с.
4. Аш Ж.Л. Датчики измерительных систем. Москва, 2008. 904 с.
5. Гуманюк Н.М. Датчики в автоматике. Москва, 2015. 264 с.
6. Іванов А.О. Теорія автоматичного керування: підручник. Дніпропетровськ, 2008. 250 с.
7. Тесленко В.А. Датчики в системах сбора данных и управления. Київ, 2006. 318 с.
8. Класифікаційний аналіз помилок вимірювання сенсора: веб-сайт URL: <http://l2u.su/qcvY> (дата звернення: 12.02.19).
9. Active-HDL: веб сайт. URL: <http://l2u.su/qcvZ> (дата звернення: 13.02.19).
10. Сенсор температури AD7814: веб-сайт. URL: <http://l2u.su/qcv0> (дата звернення: 13.02.19).
11. AD7814: вед сайт. URL: <http://l2u.su/qcv1> (дата звернення 13.02.19).
12. VHDL: веб сайт. URL: <http://l2u.su/qcv2> (дата звернення 17.03.19).
13. Що таке VHDL ?: веб-сайт. URL: <http://l2u.su/qcv3> (дата звернення: 17.02.19).
14. FPGA Design Creation and FPGA Simulation: веб-сайт. URL: <http://l2u.su/qcv4> (дата звернення: 18.03.19).
15. How to Simulate Designs in Active-HDL: веб-сайт. URL <http://l2u.su/qcv5> (дата звернення: 18.03.19).
16. Типы std\_ulogic и std\_logic: веб- сайт. URL: <http://l2u.su/qcv6>: (дата звернення: 2.05.19).

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						55
Зм.	Арк.	№ докум.	Підпис	Дата		

17. Папушин В.А., Білецький В.С. Основи автоматизації виробництва. Донецьк, 2007. 168 с.

18. Ванько В.М., Поліщук С.Є., Дорожовець М.М. Вимірювальні перетворювачі (сенсори): підручник. Львів, 2015. 580 с.

19. Архітектура ПЛІС FPGA: веб-сайт. URL: <http://12u.su/qcv7> (дата звернення: 5.05.19)

20. Как работать с редактором временных диаграмм: веб-сайт. URL: <http://12u.su/qcv8> (дата звернення: 10.02.19).

21. AD7814 Datasheet: веб-сайт. URL: <http://12u.su/qcwH> (дата звернення: 5.05.19).

22. VHDL: веб-сайт. URL: <http://12u.su/qcwI> (дата звернення: 10.05.19).

23. Что такое VHDL: веб-сайт. URL: <http://12u.su/qcwJ> (дата звернення: 10.05.19).

24. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / І.В. Гураль, Л.О. Дубчак / Під ред. О.М. Березького. - Тернопіль: ТНЕУ, 2019. – 33 с.

25. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня «Бакалавр» напряму підготовки 6.050102 «Комп'ютерна інженерія» фахового спрямування «Комп'ютерні системи та мережі» / О.М. Березький, Л.О. Дубчак, Г.М. Мельник, Ю.М. Батько, С.В. Івасьєв / Під ред. О.М. Березького. – Тернопіль ТНЕУ, 2016, - 60с.

					БР.КСМ.07127/15.00.00.000 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		