

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

Програмно-технічний засіб реалізації медіаплеєра на FPGA
Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»
Назва

Шифр КвРКІ 22045.22.03.15 ПЗ

Виконав здобувач IV курсу, група KI2-22-3


Підпис

Тимур КРИСЮК
Ініціали, прізвище

Керівник

Науковий ступінь, учене звання


Підпис

Сергій ЛИСЕНКО
Ініціали, прізвище

Нормоконтролер

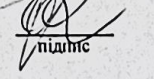
Науковий ступінь, учене звання


Підпис

Сергій ЛИСЕНКО
Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІС
«01» червня 2026 р.

дата


Підпис

Ольга ПАВЛОВА
Ініціали, прізвище

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

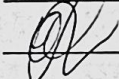
Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КПС

 Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Крисюку Тимуру Сергійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб реалізації медіаплеєра на FPGA

Керівник проекту (роботи) Лисенко Сергій Миколайович, д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Програмно технічні основи реалізації медіаплеєра на FPGA та аналіз принципів його функціонування

Проектування програмно технічного засобу медіаплеєра на базі FPGA платформи Terasic DE1-SoC

Програмно технічна реалізація медіаплеєра на FPGA на базі плати Terasic DE1-SoC

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Архітектура ПЗ проекту

Архітектура ПЗ для медіаплеєра FPGA

Апаратне забезпечення проекту


6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання « 10 » 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконано
3	Робота над розділом 1 – програмно технічні основи реалізації медіаплесра на FPGA та аналіз принципів його функціонування	01.03.2026	виконано
4	Робота над розділом 2 – проектування програмно технічного засобу медіаплесра на базі FPGA платформи Terasic DE1-SoC	01.04.2026	виконано
5	Робота над розділом 3 – програмно технічна реалізація медіаплесра на FPGA на базі плати DE1-SoC	29.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2026	виконано
7	Попередній захист ВКР	26.05.2026	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач  Тимур КРИСЮК
Підпис Імя, ПРИЗВИЩЕ

Керівник кваліфікаційної роботи  Сергій ЛИСЕНКО
Підпис Імя, ПРИЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб реалізації медіаплеєра на FPGA».

Автор роботи: Тимур КРИСІЮК.

Керівник роботи: Сергій ЛИСЕНКО.


Пояснювальна записка: 64 с., 28 рис., 0 табл., 4 дод., 50 джерел.

Графічна частина: 3 креслення.

ПРОГРАМНО ТЕХНІЧНИЙ ЗАСІБ МЕДІАПЛЕЄРА НА FPGA
АРХІТЕКТУРА ЦИФРОВОЇ СИСТЕМИ АУДІОІНТЕРФЕЙС ОБРОБКА
СИГНАЛІВ ВБУДОВАНІ СИСТЕМИ.

Кваліфікаційна робота бакалавра присвячена розробці та дослідженню програмно технічного засобу реалізації медіаплеєра на базі програмованої логічної інтегральної схеми FPGA що забезпечує відтворення цифрових аудіоданих із використанням апаратних ресурсів реконфігурованої логіки та спеціалізованих периферійних інтерфейсів. Актуальність теми зумовлена стрімким розвитком цифрових мультимедійних технологій та підвищенням вимог до швидкодії надійності енергоефективності та гнучкості електронних пристроїв обробки та відтворення інформації що використовуються у складі сучасних інформаційно керуючих систем навчальних лабораторних комплексів вбудованих пристроїв та спеціалізованих електронних платформ.

Метою роботи є проектування реалізація та експериментальне дослідження програмно технічного засобу медіаплеєра на базі FPGA що забезпечує приймання цифрових аудіоданих їх оброблення синхронізацію передавання до аудіокодека та відтворення звукової інформації у реальному часі з використанням апаратних модулів цифрової логіки. Для досягнення поставленої мети було виконано аналіз сучасних підходів до побудови цифрових медіаплеєрів на основі реконфігурованих логічних структур.


Підпис здобувача

30.05.2026

Дата

ЗМІСТ

ВСТУП.....	4
1 ПРОГРАМНО ТЕХНІЧНІ ОСНОВИ РЕАЛІЗАЦІЇ МЕДІАПЛЕЄРА НА FPGA ТА АНАЛІЗ ПРИНЦИПІВ ЙОГО ФУНКЦІОНУВАННЯ.....	5
1.1 Що це таке медіаплеєр і FPGA.....	5
1.2 Медіаплеєри на FPGA.....	9
1.3 Принцип роботи медіаплеєра на FPGA.....	11
1.4 Для чого використовують медіаплеєри на FPGA.....	13
1.5 Як користуватись медіаплеєром на FPGA.....	14
1.6 Конфігурація та програмування медіаплеєра на FPGA.....	17
1.7 Переваги та обмеження використання FPGA для медіаплеєрів.....	18
1.8 Висновки до першого розділу.....	21
2 ПРОЄКТУВАННЯ ПРОГРАМНО ТЕХНІЧНОГО ЗАСОБУ МЕДІАПЛЕЄРА НА БАЗІ FPGA ПЛАТФОРМИ TERAASIC DE1 SOC.....	22
2.1 Апаратна платформа медіаплеєра на базі FPGA та її основні характеристики	22
2.2 Архітектурна організація плати Terasic DE1 SoC та структура підключення основних компонентів.....	25
2.3 Апаратні інтерфейси та роз'єми плати Terasic DE1 SoC.....	27
2.4 Аудіопідсистема плати Terasic DE1 SoC та її використання на FPGA.....	30
2.5 Організація зберігання та зчитування аудіоданих на FPGA.....	33
2.6 Функціональні можливості середовища Intel Quartus Prime при розробці медіаплеєра на FPGA.....	36
2.7 Процес конфігурації та програмування FPGA на базі Terasic DE1-SoC із використанням середовища Intel Quartus Prime.....	39
2.8 Висновки до другого розділу.....	42

КвРКІ.22045.22.03.15 ПЗ								
Зм.	Арк.	№докум.	Підпис	Дата	Програмно-технічний засіб реалізації медіаплеєра на FPGA Пояснювальна записка	Літера	Аркуш	Аркушів
Виконав		Гимур КРИСЮК		01.06		у		
Перевір.		Сергій ЛИСЕНКО		01.06			2	64
Н.контр.		Сергій ЛИСЕНКО		01.06		ХНУ КІ2-22-3		
Затвер.		Ольга ПАВЛОВА		01.06				

3 ПРОГРАМНО ТЕХНІЧНА РЕАЛІЗАЦІЯ МЕДІАПЛЕЄРА НА FPGA НА БАЗІ ПЛАТИ TERCASIC DE1 SOC	43
3.1 Створення проєкту у середовищі Quartus та початкова підготовка системи	43
3.2 Розроблення модуля обробки кнопок button_controller.v	46
3.3 Розроблення модуля керування player_fsm.v	48
3.4 Розроблення модуля генерації аудіосигналу audio_sample_generator.v	50
3.5 Розроблення модуля передачі аудіоданих i2s_driver.v.....	53
3.6 Розроблення модуля ініціалізації аудіокодека wm8731_init.v	55
3.7 Розроблення верхнього модуля системи media_player_top.v та інтеграція всіх компонентів.....	57
3.8 Налаштування виводів FPGA та запуск проєкту на апаратній платформі	60
3.9 Висновки до третього розділу.....	66
ВИСНОВКИ.....	67
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	68
ДОДАТОК А АРХІТЕКТУРА ПЗ МЕДІАПЛЕЄРА FPGA	73
ДОДАТОК Б АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ПРОЕКТУ	74
ДОДАТОК В СТРУКТУРНА СХЕМА ОБРОБКИ ДАНИХ.....	75
ДОДАТОК Г ЛІСТИНГ КОДУ ПРОГРАМИ	76

ВСТУП

Сучасний розвиток цифрових електронних технологій характеризується широким впровадженням програмно технічних засобів оброблення та відтворення мультимедійної інформації у навчальні дослідницькі та прикладні інформаційні системи. Значна частина сучасних електронних пристроїв працює з цифровими аудіоданими що потребує використання високопродуктивних апаратних платформ здатних забезпечувати стабільну роботу у режимі реального часу та підтримувати точну синхронізацію сигналів керування і передавання інформації. Реалізація медіаплеєра на базі програмованої логіки дозволяє забезпечити ефективне керування потоками цифрових аудіоданих синхронізацію роботи периферійних пристроїв аудіотракту організацію взаємодії з аудіокодеком та формування стабільного відтворення звукової інформації у режимі реального часу що є важливою складовою функціонування сучасних цифрових систем мультимедійного призначення. Метою дипломної роботи є розроблення програмно технічного засобу реалізації медіаплеєра на FPGA що забезпечує приймання синхронізацію оброблення та передавання цифрових аудіоданих до аудіокодека для подальшого відтворення звукової інформації у режимі реального часу із використанням апаратних ресурсів програмованої логічної інтегральної схеми. Об'єктом дослідження є процес функціонування програмно технічного засобу реалізації медіаплеєра на базі програмованої логічної інтегральної схеми FPGA у складі цифрової системи відтворення аудіоінформації. Предметом дослідження є методи апаратної реалізації цифрових модулів керування аудіоінтерфейсом синхронізації потоків даних та організації взаємодії програмованої логічної інтегральної схеми FPGA з аудіокодеком.

					КвРКІ.22045.22.03.15 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ПРОГРАМНО ТЕХНІЧНІ ОСНОВИ РЕАЛІЗАЦІЇ МЕДІАПЛЕЄРА НА FPGA ТА АНАЛІЗ ПРИНЦИПІВ ЙОГО ФУНКЦІОНУВАННЯ

1.1 Що це таке медіаплеєр і FPGA

Медіаплеєр - це програмний або апаратний пристрій, призначений для відтворення мультимедійного контенту, такого як відео аудіо та графічні файли. Основною функцією медіаплеєра є декодування стиснених даних відповідно до певного стандарту кодека та формування сигналу, придатного для відображення на екрані або відтворення через аудіосистему.

Медіаплеєри можуть бути реалізовані у різних формах: як програмне забезпечення для персональних комп'ютерів, як вбудовані системи у телевізорах та приставках, як автономні пристрої наприклад мультимедійних приставок, а також як спеціалізовані промислові модулі. З технічної точки зору медіаплеєр виконує такі основні функції: процес роботи медіаплеєра включає кілька послідовних етапів: спочатку здійснюється зчитування даних із носія або мережі після чого відбувається буферизація потоку для забезпечення безперервності відтворення. Далі дані проходять декодування, що дозволяє перетворити відео- та аудіоінформацію у придатний для відтворення формат. Наступним кроком є обробка сигналу, яка може включати масштабування зображення та перетворення кольорового простору.

Завершальним етапом є виведення сигналу через відповідний інтерфейс, наприклад HDMI, VGA чи аудіовихід, що забезпечує відтворення мультимедійного контенту на кінцевому пристрої [6]. FPGA у цьому контексті виступає як апаратна платформа що дозволяє реалізувати всі ці процеси на рівні логічних блоків, забезпечуючи високу швидкість обробки та мінімальні затримки. Завдяки цьому медіаплеєри на FPGA можуть працювати у реальному часі та підтримувати сучасні формати мультимедіа з високою роздільною здатністю [42].

					КВРКІ.22045.22.03.15 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

На рисунку 1.1 показаний стандартний медіаплеєр для телевізора.



Рисунок 1.1 - Медіаплеєр Vinga 042 [32]

https://rozetka.com.ua/ua/vinga_vmp_043_324/p216450115/

На рисунку 1.2 показаний компактний медіаплеєр.



Рисунок 1.2 - Медіаплеєр Dune HD Magic 4K Plus [34]

<https://rozetka.com.ua/ua/376971381/p376971381/>

Field Programmable Gate Array - це програмована логічна інтегральна схема архітектура якої може бути налаштована користувачем після виробництва. На відміну від мікропроцесорів із фіксованою внутрішньою структурою, FPGA дозволяє створювати власну цифрову схему відповідно до вимог конкретного проєкту [44]. FPGA являє собою інтегральну схему яка складається з конфігурованих логічних блоків, програмованих з'єднань між ними, вбудованих

					КВРКІ.22045.22.03.15 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

блоків пам'яті апаратних модулів цифрової обробки сигналів та блоків керування тактовими сигналами. Її конфігурація здійснюється за допомогою мов опису апаратури таких як VHDL або Verilog після чого синтезований проєкт перетворюється у конфігураційний файл, що завантажується у мікросхему та визначає її функціональність. Найбільш відомими виробниками FPGA є компанії Xilinx та Intel.

На рисунку 1.3 показаний FPGA на базі Xilinx.

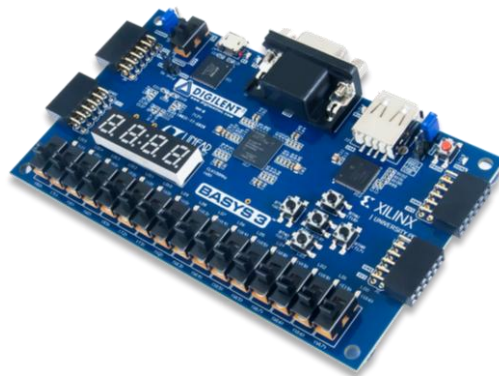


Рисунок 1.3 - FPGA Xilinx Artix-7 [43]

<https://digilent.com/shop/basys-3-amd-artix-7-fpga-trainer-board-recommended-for-introductory-users/>

На рисунку 1.4 показаний FPGA на базі Intel.

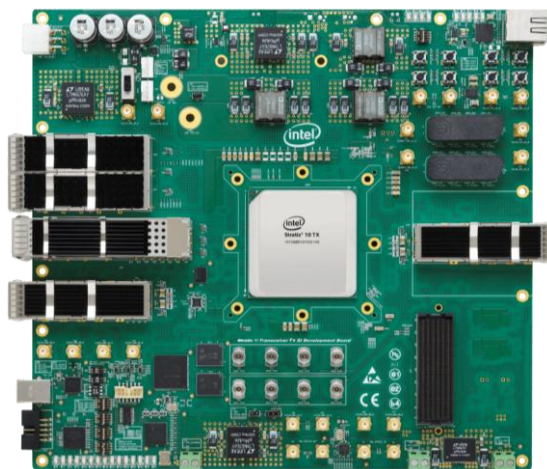


Рисунок 1.4 - FPGA на Intel [46]

<https://blog.samtec.com/post/new-intel-fpga-platform-features-samtec-interconnect/>

					КВРКІ.22045.22.03.15 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

У контексті даного проєкту медіаплеєр і FPGA поєднуються в єдину інтегровану систему в якій функції обробки мультимедійних даних реалізуються на апаратному рівні цифрової логіки. На відміну від класичних програмних медіаплеєрів, що працюють на універсальних процесорах із послідовним виконанням інструкцій у даному випадку значна частина обчислювально складних операцій переноситься безпосередньо у структуру програмованої логіки FPGA.

Процеси як демультимплексування потоку, декодування відео та аудіоданих, буферизація, перетворення кольорового простору наприклад, YCbCr RGB, масштабування зображення, формування синхросигналів та генерація відеовихідного інтерфейсу HDMI, VGA тощо, можуть бути реалізовані у вигляді паралельно працюючих апаратних модулів. Кожен модуль функціонує незалежно, що суттєво підвищує загальну пропускну здатність системи.

Використання FPGA дозволяє створити спеціалізовану архітектуру обробки сигналів із високим рівнем паралелізму що забезпечує: високу швидкодію за рахунок одночасного виконання кількох операцій; мінімальні затримки обробки low latency, що є критично важливим для відеосистем реального часу; стабільність роботи незалежно від навантаження операційної системи; детермінованість часових характеристик; можливість глибокої оптимізації під конкретний формат, роздільну здатність або алгоритм обробки. FPGA забезпечує гнучкість модернізації системи шляхом перепрограмування логіки без необхідності зміни апаратної платформи.

Архітектура FPGA дозволяє використовувати вбудовані блоки пам'яті апаратні множники та DSP-модулі для реалізації цифрової обробки сигналів без залучення зовнішніх процесорних ресурсів. Це зменшує енергоспоживання системи та підвищує її надійність.

У проєкті медіаплеєр розглядається не лише як програмний застосунок, а як комплексний програмно-технічний засіб.

					КВРКІ.22045.22.03.15 ПЗ	Арк. 8
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2 Медіаплеєри на FPGA

Медіаплеєри на базі FPGA можуть відрізнятися за архітектурою, функціональним призначенням, рівнем складності реалізації та сферою застосування. Класифікація таких систем дозволяє чітко визначити їх технічні особливості та обрати оптимальний варіант для конкретного проєкту. За архітектурою реалізації повністю апаратні медіаплеєри виконують усі основні функції, такі як демультимплексування, декодування обробка відео та аудіо, а також формування інтерфейсів виводу у вигляді логічних модулів FPGA.

У таких системах відсутній або майже не використовується вбудований процесор, адже обробка здійснюється паралельно на рівні цифрових блоків. Це забезпечує високу продуктивність чітку детермінованість часових характеристик та мінімальну затримку сигналу. Водночас складність може виникати при реалізації алгоритмів стиснення які потребують значних апаратних ресурсів і ускладнюють проєктування системи. Повністю апаратні медіаплеєри мають перевагу у стабільності роботи, адже вони не залежать від операційних систем чи програмних драйверів, що зменшує ризик збоїв.

Такі рішення часто застосовуються у критично важливих сферах, де необхідна безперервна обробка даних у реальному часі, наприклад у професійних відеопроцесорах, системах телемовлення чи спеціалізованій промисловій техніці. Окрім повністю апаратних рішень, існують комбіновані архітектури, де частина функцій реалізується апаратно, а частина програмно на вбудованому процесорі. Це щоб зменшити складність проєктування та забезпечити більшу гнучкість у налаштуванні системи. Такі медіаплеєри можуть підтримувати широкий спектр форматів і стандартів, а також швидко адаптуватися до нових вимог завдяки оновленню програмної частини.

Використання SoC FPGA, де інтегроване процесорне ядро працює разом із апаратною логікою. Це поєднання дозволяє створювати багатфункціональні системи, які одночасно забезпечують високу продуктивність і простоту

					КВРКІ.22045.22.03.15 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

керування. У таких рішеннях програмна частина може відповідати за інтерфейси користувача, мережеву взаємодію та керування потоками даних, тоді як апаратна логіка виконує найскладніші обчислювальні завдання.

На рисунку 1.5 показаний апаратний медіаплеєр.



Рисунок 1.5 - Апаратний медіаплеєр [13]

<https://www.rakoit.com/do-i-need-a-phono-preamp-for-my-turntable/>

Гібридні медіаплеєри на базі SoC FPGA поєднують у собі апаратну логіку та інтегроване процесорне ядро, наприклад ARM. У такій архітектурі програмна частина відповідає за керування файловою системою, роботу з мережевими протоколами та організацію інтерфейсу користувача, тоді як апаратна частина бере на себе ресурсоємні обчислення, зокрема декодування відео, масштабування та фільтрацію. Завдяки цьому підходу досягається універсальність системи, адже вона поєднує гнучкість програмного забезпечення з високою продуктивністю апаратної логіки [29]. Перевагою гібридних медіаплеєрів є можливість швидкої адаптації до нових стандартів і форматів мультимедіа. Якщо апаратна частина відповідає за базові операції то програмна може бути оновлена для підтримки нових кодеків, протоколів чи функцій без необхідності змінювати апаратну архітектуру. Це робить такі системи більш

довговічними та економічно вигідними, адже модернізація здійснюється переважно на рівні програмного забезпечення [28].

Можливість інтеграції додаткових сервісів, таких як потокове відео, хмарні технології чи системи захисту даних. Процесорне ядро може виконувати завдання з обробки мережевих запитів, шифрування та автентифікації, тоді як FPGA-логіка забезпечує безперервне відтворення мультимедіа без затримок. Це відкриває перспективи для використання гібридних медіаплеєрів у комерційних платформах, інтерактивних інформаційних системах та навіть у промислових рішеннях, де важлива безпека й стабільність роботи. У сфері освіти та досліджень такі плеєри дозволяють студентам і науковцям експериментувати з різними архітектурними підходами, поєднуючи апаратну оптимізацію з програмною гнучкістю. Це сприяє розвитку навичок у галузі цифрової логіки, систем-на-кристалі та високорівневого програмування [14].

На рисунку 1.6 показаний гібридний медіаплеєр.



Рисунок 1.6 - Гібридний медіаплеєр [29]

<https://allo.ua/ua/products/mediapleery/gibridnyj-mediapleer-mecool-kiii-pro-dvb-s2-t2-c.html?srsId=AfmBOoqdYFFjk8rKDwXL6hXtuJMSWQcQO0oqQ40iHN2H-XuFAhEVuG6Y>

1.3 Принцип роботи медіаплеєра на FPGA

					КВРКІ.22045.22.03.15 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

Медіаплеер на FPGA базується на апаратній реалізації алгоритмів обробки мультимедійних даних із високим рівнем паралелізму та детермінованою затримкою. Основна ідея полягає у перенесенні ключових обчислювально складних операцій із програмного забезпечення на апаратні модулі FPGA, що дозволяє досягти мінімальної затримки та стабільної обробки потоків відео й аудіо. Це особливо важливо для систем, які працюють у режимі реального часу, де навіть невелика затримка може призвести до втрати якості відтворення або розсинхронізації сигналів.

Дані можуть надходити з локальних носіїв, таких як SD-карти чи USB-накопичувачі, або з мережевих потоків через Ethernet чи IPTV. FPGA реалізує апаратні контролери інтерфейсів, що забезпечують швидку передачу даних у буферну пам'ять. Буферизація є критично важливим етапом, адже вона дозволяє компенсувати нерівномірність надходження даних і гарантує безперервність відтворення. Виконання цього процесу апаратними засобами мінімізує затримки та забезпечує стабільність роботи навіть при високих навантаженнях.

Декодування мультимедійних потоків здійснюється апаратними модулями всередині FPGA. Для цього можуть використовуватися блоки ентропійного декодування, зворотного перетворення DCT, компенсації руху та кольорового перетворення. Завдяки паралельній архітектурі FPGA можлива одночасна обробка кількох кадрів, що забезпечує реальний час роботи навіть із відео високої роздільної здатності. Це дозволяє підтримувати сучасні формати, такі як Full HD, 4K чи навіть 8K, без перевантаження системи.

Обробка відеосигналу включає масштабування зображення, корекцію кольору, накладання графічних елементів та формування сигналів синхронізації. FPGA дозволяє реалізувати апаратні фільтри та ефекти, які працюють у реальному часі без втрати продуктивності. Це відкриває можливості для створення інтерактивних систем, де користувач може змінювати параметри відтворення без зниження якості. Аудіопотоки проходять через DSP-блоки, які виконують декодування, фільтрацію, еквалізацію та мікшування. Завдяки цьому

забезпечується високоякісний звук із мінімальною затримкою, що є важливим для професійних аудіосистем та мультимедійних комплексів. FPGA може реалізувати складні алгоритми цифрової обробки сигналів, які у програмному виконанні потребували б значних ресурсів і часу. Завершальним етапом є виведення сигналу на зовнішні пристрої через HDMI, VGA, DisplayPort або цифрові аудіоінтерфейси. Апаратна синхронізація аудіо та відео виключає розсинхронізацію сигналів, що гарантує плавне та якісне відтворення. Це особливо важливо для систем, які використовуються у телебаченні, відеоспостереженні чи автомобільній електроніці де точність і стабільність мають вирішальне значення.

1.4 Для чого використовують медіаплеєри на FPGA

Медіаплеєри на FPGA застосовуються у випадках, коли необхідна висока продуктивність, мінімальна затримка та стабільність роботи системи. Вони використовуються у цифровому телебаченні та IPTV для швидкої обробки потоків і підтримки сучасних стандартів стиснення, у промислових відеосистемах та системах відеоспостереження для аналізу й передачі даних у реальному часі, в автомобільній електроніці для роботи з камерами та мультимедійними системами, а також у вбудованих і спеціалізованих рішеннях, де важлива енергоефективність і безпека. FPGA активно застосовується в освітніх та наукових проєктах для дослідження цифрової обробки сигналів і створення прототипів, а також у комерційних і розважальних пристроях для відтворення контенту високої роздільної здатності без затримок. Завдяки гнучкості архітектури такі плеєри можуть адаптуватися до різних форматів і вимог, що робить їх універсальним інструментом для сучасних мультимедійних систем.

Використання FPGA у медіаплеєрах дозволяє інтегрувати апаратні алгоритми обробки даних, які забезпечують стабільність роботи навіть при

					КВРКІ.22045.22.03.15 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

високих навантаженнях. Це актуально для систем потокового відео, де важлива безперервність трансляції та відсутність затримок. У промислових умовах такі рішення допомагають підвищити точність аналізу відео та забезпечити швидке реагування на зміни у виробничих процесах.

В автомобільній сфері FPGA-плеєри можуть бути частиною систем допомоги водієві, де обробка сигналів у реальному часі є критичною для безпеки. У сфері освіти та науки FPGA відкриває широкі можливості для досліджень і практичного навчання. Студенти та інженери можуть створювати власні прототипи систем, експериментувати з алгоритмами обробки сигналів та вивчати принципи паралельної архітектури. Це сприяє розвитку навичок апаратного програмування та формує базу для майбутніх інновацій.

У комерційних і розважальних системах FPGA забезпечує плавне відтворення контенту високої роздільної здатності, включно з HDR та 3D, що значно покращує користувацький досвід. Медіаплеєри на FPGA можуть бути інтегровані у системи доповненої та віртуальної реальності, де потрібна обробка великих обсягів даних у реальному часі. Це відкриває перспективи для створення інтерактивних мультимедійних платформ, які поєднують відео, аудіо та графіку з високою точністю синхронізації. У сфері реклами та маркетингу такі рішення дозволяють реалізувати цифрові вивіски та інформаційні панелі з динамічним контентом, що привертає увагу користувачів. FPGA-плеєри можуть застосовуватися для моделювання та тестування нових алгоритмів стиснення або обробки сигналів, що дає змогу перевіряти їх ефективність у реальних умовах. Це особливо важливо для розробки нових стандартів мультимедіа, які потребують високої продуктивності та оптимізації енергоспоживання.

1.5 Як користуватись медіаплеєром на FPGA

Використання медіаплеєра на FPGA має свої особливості, адже він поєднує апаратні та програмні компоненти. Для роботи необхідно підключити плату до

					КВРКІ.22045.22.03.15 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

джерела живлення та інтерфейсів, завантажити конфігураційний файл, підготувати мультимедійний контент у підтримуваних форматах і налаштувати параметри відтворення.

Керування може здійснюватися апаратними кнопками, графічним або мережевим інтерфейсом. У процесі роботи дані проходять через модулі декодування, буферизації та обробки після чого готовий сигнал виводиться на зовнішні пристрої з апаратною синхронізацією. Для стабільної роботи рекомендується використовувати сумісні носії та формати, забезпечувати якісне живлення й охолодження плати, а також контролювати пропускну здатність мережі при потоковому відтворенні.

Правильне користування медіаплеєром на FPGA забезпечує високу якість відтворення без затримок, стабільність у реальному часі та можливість швидкої адаптації системи під різні джерела сигналу, що робить його ефективним рішенням як для навчальних, так і для промислових застосувань. Налаштування FPGA-плеєра може відрізнятись залежно від конкретної платформи та архітектури.

У випадку використання SoC FPGA, де інтегроване процесорне ядро працює разом із апаратною логікою, користувач може налаштовувати параметри відтворення через програмні інтерфейси, що робить систему більш гнучкою та зручною у використанні. Це допомагає швидко змінювати конфігурацію для підтримки нових форматів або інтегрувати додаткові функції, наприклад мережеве керування чи віддалене оновлення прошивки.

FPGA допомагає студентам зрозуміти принципи роботи цифрових систем, адже вони можуть на практиці побачити, як дані проходять через апаратні модулі, як здійснюється синхронізація потоків та як оптимізується робота системи. У промислових умовах правильне користування таким плеєром дозволяє інтегрувати його у складні виробничі процеси, забезпечуючи стабільність і високу якість відтворення мультимедійних даних.

Користування FPGA-плеєром вимагає дотримання певних правил безпеки та технічної дисципліни. важливо контролювати температуру роботи плати, оскільки при обробці відео високої роздільної здатності навантаження на апаратні модулі значно зростає. Для цього застосовуються системи охолодження, які гарантують стабільність роботи навіть у тривалих режимах.

Також що при потоковому відтворенні через мережу необхідно забезпечити достатню пропускну здатність каналу, щоб уникнути втрати кадрів чи розсинхронізації аудіо та відео.

На рисунку 1.7 показано як взаємодіє користувач з медіаплеєром.

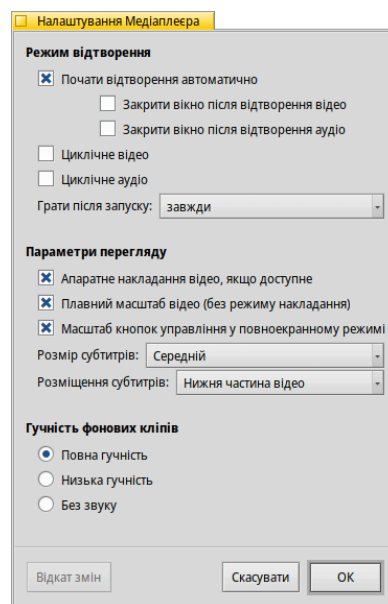


Рисунок 1.7 - Взаємодія користувача з медіаплеєром [29]

<https://www.haiku-os.org/docs/userguide/uk/applications/mediaplayer.html>

У застосуваннях користування медіаплеєром на FPGA дозволяє створювати інтерактивні системи, які можуть працювати без операційної системи, що підвищує їхню надійність та безпеку. Це актуально для цифрових вивісок, інформаційних панелей чи мультимедійних кіосків, де важлива безперервність роботи та можливість швидкої адаптації. Використання FPGA дає змогу оптимізувати оброблення аудіо та відеоданих під конкретні вимоги системи, забезпечуючи стабільну роботу навіть при високому навантаженні.

Завдяки можливості перепрограмування такі медіаплеєри можуть легко оновлюватися та розширювати свій функціонал без заміни апаратної частини.

1.6 Конфігурація та програмування медіаплеєра на FPGA

Медіаплеєр на FPGA поєднує апаратну та програмну складові, тому його налаштування та програмування потребує знань цифрової логіки, мов опису апаратури та принципів обробки мультимедійних потоків. Конфігурація визначає поведінку FPGA та реалізацію всіх функцій плеєра, включаючи декодування відео й аудіо, обробку сигналів і вивід на зовнішні пристрої.

Основні етапи конфігурації охоплюють вибір апаратної платформи залежно від продуктивності, обсягу логічних блоків, пам'яті та підтримуваних інтерфейсів, підготовку середовища розробки на базі спеціалізованих IDE, створення проєкту мовами VHDL чи Verilog із описом функціональних блоків та використанням готових IP-модулів, а також синтез і генерацію конфігураційного файлу, який завантажується на плату. У випадку SoC FPGA програмна частина на базі процесорного ядра виконує керування потоками даних, обробку команд користувача та мережеву взаємодію, а програмування здійснюється мовами високого рівня. Тестування включає симуляцію, налагодження на платі та оптимізацію конфігурації для досягнення реального часу обробки навіть при високій роздільній здатності.

Практичні рекомендації передбачають використання готових IP-блоків для популярних кодеків, забезпечення достатнього обсягу пам'яті, ретельне планування архітектури та поступове ускладнення проєктів від простих форматів до сучасних потоків. Правильна конфігурація забезпечує створення високопродуктивного медіаплеєра для будь-якого формату, гнучке налаштування апаратних ресурсів, мінімальні затримки при обробці мультимедіа та можливість інтеграції у промислові й вбудовані системи з високими вимогами до надійності.

Процес конфігурації FPGA-плеєра може включати оптимізацію використання апаратних ресурсів, таких як логічні блоки, блоки пам'яті та апаратні множники, що дозволяє досягти максимальної ефективності системи. У складних проєктах часто застосовується модульна архітектура, де кожен функціональний блок реалізується окремо й може бути повторно використаний у майбутніх розробках. Це значно скорочує час створення нових систем і підвищує їхню надійність.

Аспектом є також інтеграція FPGA-плеєра з іншими цифровими пристроями, наприклад сенсорами, мережевими модулями чи системами зберігання даних, що розширює його функціональні можливості. У промислових умовах правильна конфігурація медіаплеєра на FPGA дозволяє забезпечити стабільну роботу обладнання навіть при високих навантаженнях, а також гарантує точність синхронізації сигналів.

Це актуально для систем відеоспостереження, телемовлення чи автомобільної електроніки, де затримки або помилки можуть мати критичні наслідки. У навчальних і дослідницьких проєктах конфігурація FPGA-плеєра допомагає студентам та інженерам зрозуміти принципи роботи цифрових систем, адже вони можуть на практиці побачити, як апаратна логіка взаємодіє з програмними компонентами.

1.7 Переваги та обмеження використання FPGA для медіаплеєрів

Використання програмованих логічних інтегральних схем FPGA для реалізації медіаплеєрів є сучасним підходом до побудови високопродуктивних мультимедійних систем. На відміну від традиційних програмних рішень, що виконуються на центральних процесорах CPU або графічних процесорах GPU, FPGA дозволяє реалізувати обробку відео та аудіо потоків на апаратному рівні, що забезпечує високу швидкодію, мінімальні затримки та можливість гнучкої

адаптації архітектури до конкретних вимог. Проте разом із перевагами існують і певні обмеження, які необхідно враховувати при проєктуванні медіаплеєра.

Переваги FPGA є можливість паралельної обробки даних. На відміну від класичної послідовної архітектури процесора, програмована логіка дозволяє одночасно виконувати велику кількість операцій. У медіаплеєрі це особливо важливо для декодування відео високої роздільної здатності Full HD, 4K, обробки аудіосигналів, масштабування зображення, перетворення кольорових просторів та буферизації кадрів. Завдяки цьому система може працювати в режимі реального часу без втрати кадрів і затримок відтворення. Іншою суттєвою перевагою є низька латентність затримка сигналу. Оскільки обробка виконується на рівні апаратної логіки, відсутні накладні витрати, характерні для операційних систем або програмного стеку.

Це для вбудованих систем, мультимедійних пристроїв з жорсткими вимогами до синхронізації аудіо та відео, а також для промислових застосувань, де критичною є точність часових характеристик. FPGA також забезпечує високу гнучкість архітектури. Розробник може змінювати структуру обчислювальних блоків, додавати або видаляти модулі декодування, реалізовувати спеціалізовані алгоритми обробки сигналів відповідно до потреб проєкту. Це дозволяє створювати медіаплеєри, оптимізовані під конкретні формати відео, типи інтерфейсів HDMI, VGA, DisplayPort або вимоги до енергоспоживання. Можливість апаратної оптимізації алгоритмів.

Реалізація декодерів, фільтрів, перетворень Фур'є або операцій масштабування безпосередньо в логічних блоках FPGA дозволяє досягти високої продуктивності при відносно невеликому енергоспоживанні порівняно з універсальними процесорами. У випадку SoC-FPGA систем на кристалі з інтегрованим процесорним ядром поєднується гнучкість програмного керування та швидкодія апаратного прискорення. Крім того, FPGA дозволяє забезпечити довготривалу підтримку та модернізацію пристрою. За необхідності можна оновити конфігураційний файл без заміни апаратного забезпечення, що є

економічно вигідним для підприємств і виробників мультимедійного обладнання.

Попри значні переваги, використання FPGA має низку обмежень. Насамперед це складність розробки. Проектування медіаплеєра на FPGA вимагає знань мов опису апаратури VHDL або Verilog, принципів цифрової схемотехніки, синхронізації сигналів та оптимізації ресурсів. На відміну від програмної розробки, помилки в апаратній логіці можуть призвести до некоректної роботи всієї системи або перевитрати ресурсів кристала. Фактором є обмеженість апаратних ресурсів. Кількість логічних елементів, блоків пам'яті та апаратних множників у FPGA є фіксованою. При реалізації складних відеокодеків або підтримці високих роздільних здатностей може виникнути потреба у використанні дорогих моделей FPGA з великим обсягом ресурсів, що збільшує загальну вартість системи.

Враховуючи складність реалізації сучасних відеокодеків H.264, H.265, AV1 повністю з нуля. Вони містять складні алгоритми прогнозування, ентропійного кодування та компенсації руху. Без використання готових IP-ядер розробка таких модулів є трудомісткою і потребує значного часу. Використання ж ліцензованих IP-блоків може підвищити вартість проєкту. Порівняно довший цикл розробки. Процеси синтезу, трасування та генерації можуть займати значний час, особливо для великих проєктів. Це уповільнює тестування та внесення змін у систему порівняно з програмними рішеннями. FPGA зазвичай поступаються спеціалізованим ASIC у показниках енергоспоживання та максимальної частоти роботи.

Якщо необхідне масове виробництво однакових пристроїв із фіксованим функціоналом, економічно доцільнішим може бути використання ASIC-рішень. Проте для гнучких або експериментальних систем FPGA залишається більш універсальним варіантом. Використання FPGA у медіаплеєрах відкриває широкі можливості для інтеграції новітніх технологій, таких як штучний інтелект та машинне навчання. Завдяки апаратній логіці можна реалізувати алгоритми

					КВРКІ.22045.22.03.15 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

розпізнавання об'єктів, аналізу відеопотоків чи адаптивної обробки аудіо без значних затримок. Це робить FPGA-плеєри перспективними для застосування у системах відеоспостереження, автомобільних комплексах допомоги водієві ADAS та інтелектуальних мультимедійних платформах.

1.8 Висновки до першого розділу

У першому розділі було проведено аналіз основних принципів побудови програмно технічних засобів реалізації медіаплеєра на базі програмованих логічних інтегральних схем FPGA та розглянуто особливості їх застосування у складі сучасних цифрових мультимедійних систем. У процесі виконаного дослідження визначено сутність медіаплеєра як спеціалізованого цифрового пристрою призначеного для приймання оброблення та відтворення аудіоданих у реальному режимі часу із забезпеченням стабільної синхронізації сигналів керування та передавання інформації між функціональними модулями системи. Особливу увагу приділено аналізу можливостей використання програмованих логічних інтегральних схем FPGA як універсальної апаратної платформи для реалізації цифрових пристроїв обробки аудіосигналів що дозволяє забезпечити високу швидкодію гнучкість конфігурації та ефективність функціонування медіаплеєра у складі сучасних вбудованих систем. У ході аналізу розглянуто основні типи медіаплеєрів що можуть бути реалізовані на базі FPGA з урахуванням особливостей організації цифрового тракту передавання аудіоданих структури взаємодії функціональних модулів та способів синхронізації сигналів керування периферійними пристроями аудіотракту.

					КвРКІ.22045.22.03.15 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ПРОГРАМНО ТЕХНІЧНОГО ЗАСОБУ МЕДІАПЛЕЄРА НА БАЗІ FPGA ПЛАТФОРМИ TERASIC DE1 SOC

2.1 Апаратна платформа медіаплеєра на базі FPGA та її основні характеристики

Проєктування програмно технічного засобу реалізації медіаплеєра на FPGA починається з вибору апаратної платформи яка визначає можливості майбутньої системи рівень її продуктивності доступні периферійні інтерфейси та особливості організації цифрової логіки. У якості базової платформи у даній роботі використовується відладочна плата Terasic DE1 SoC що побудована на основі програмованої логічної інтегральної схеми Intel Cyclone V моделі 5CSEMA5F31C6.

На рисунку 2.1 показано плату Terasic DE1 SOC.

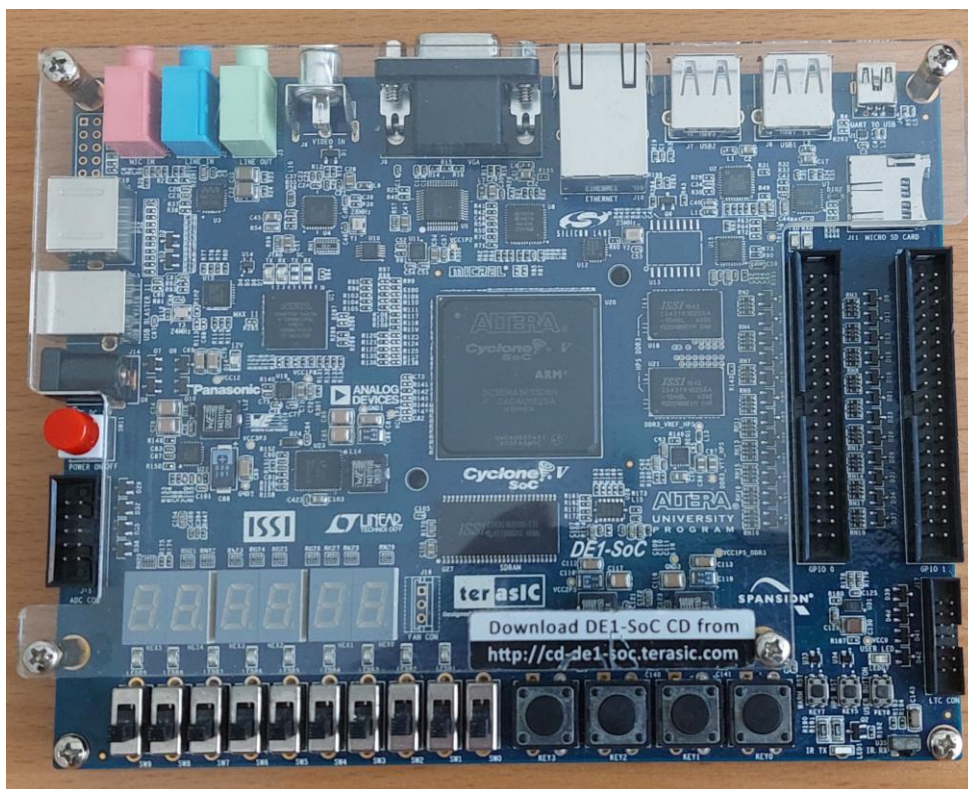


Рисунок 2.1 - Terasic DE1 SOC модель плати на якій зроблений проєкт

Дана плата поєднує у собі можливості класичної FPGA логіки та вбудованої процесорної системи що створює гнучке середовище для розроблення складних цифрових пристроїв зокрема мультимедійних систем реального часу.

Використаний кристал FPGA містить значну кількість логічних елементів таблиць істинності регістрів та апаратних блоків обробки сигналів що дозволяє реалізувати цифрові модулі різного рівня складності включаючи генератори тактових сигналів інтерфейси передавання аудіоданих модулі синхронізації та керування периферійними пристроями. Наявність вбудованих блоків пам'яті дає змогу організовувати буферизацію аудіоданих що є необхідною умовою для забезпечення безперервності відтворення звукового потоку.

Апаратна структура FPGA дозволяє реалізувати паралельну обробку сигналів що підвищує ефективність функціонування медіаплеєра. Плата Terasic DE1 SoC оснащена широким набором периферійних інтерфейсів що значно спрощує процес розроблення та тестування цифрових систем. До складу плати входять універсальні порти введення та виведення які реалізовані у вигляді роз'ємів для підключення зовнішніх пристроїв що дозволяє розширювати функціональні можливості системи залежно від поставлених задач. Наявність світлодіодних індикаторів забезпечує можливість візуального контролю стану логічних сигналів у процесі налагодження системи.

Тактові генератори що інтегровані у плату формують базові сигнали синхронізації які використовуються для побудови внутрішніх цифрових трактів обробки даних. Особливу роль у реалізації медіаплеєра відіграє аудіопідсистема плати яка представлена вбудованим аудіокодеком що забезпечує перетворення цифрових аудіоданих у аналоговий сигнал для подальшого відтворення через підключені акустичні пристрої. Для взаємодії з аудіокодеком використовується спеціалізований цифровий інтерфейс що забезпечує передавання аудіоданих та сигналів синхронізації між FPGA та периферійним пристроєм. Це дозволяє реалізувати повноцінний аудіотракт у межах однієї відладочної плати без

необхідності використання додаткових зовнішніх компонентів. Наявність відеовиходів інтерфейсів зв'язку портів пам'яті та інших допоміжних модулів розширює можливості використання плати та створює умови для реалізації складних мультимедійних систем із підтримкою різних форматів представлення інформації.

Елементом системи є організація обміну даними між пам'яттю та обчислювальними блоками. Використання зовнішньої динамічної пам'яті дозволяє зберігати великі аудіофайли та реалізовувати буферизацію, що згладжує можливі затримки під час зчитування даних. Завдяки цьому забезпечується безперервність відтворення навіть при нерівномірному доступі до джерела даних. Обмін інформацією між процесорною частиною та логічними модулями FPGA реалізується через високошвидкісні внутрішні шини, що дозволяє досягти низької латентності передачі та високої пропускну здатності. Особливу увагу при проектуванні приділено питанням синхронізації цифрових потоків. У системі використовуються генератори тактових сигналів та схеми фазового автопідлаштування частоти, що забезпечують формування стабільних тактових доменів для різних частин медіаплеєра. Це дозволяє узгодити роботу аудіокодека, логіки обробки сигналу та системи пам'яті, що є критичним для коректного відтворення звуку без спотворень.

У межах даної роботи основна увага приділяється використанню аудіопідсистеми та цифрових ресурсів FPGA для побудови медіаплеєра що здатний працювати у режимі реального часу із забезпеченням стабільного відтворення звукової інформації. Вибір плати Terasic DE1 SoC обумовлений її широкими функціональними можливостями наявністю необхідних апаратних ресурсів для реалізації цифрового аудіотракту а також доступністю інструментальних засобів розроблення що підтримують дану платформу. Використання цієї плати можна створити ефективний програмно технічний засіб медіаплеєра який поєднує у собі апаратну реалізацію ключових функцій та

можливість подальшої модернізації системи шляхом зміни конфігурації програмованої логіки.

2.2 Архітектурна організація плати Terasic DE1 SoC та структура підключення основних компонентів

Архітектура плати Terasic DE1 SoC побудована за принципом інтеграції програмованої логічної частини та процесорної підсистеми в межах одного кристалу що забезпечує високий рівень взаємодії між апаратними модулями та дозволяє ефективно організувати оброблення цифрових даних. Центральним елементом плати є мікросхема Intel Cyclone V яка містить як логічну матрицю FPGA так і вбудовану процесорну систему на базі двоядерного обчислювального ядра архітектури ARM. Така побудова відкриває можливість реалізації складних цифрових систем у яких частина функцій виконується апаратно у логічній матриці а частина може бути покладена на програмну обробку у процесорному середовищі.

Логічна частина FPGA використовується для реалізації швидкодіючих модулів обробки сигналів які потребують високої точності синхронізації та мінімальних затримок при передаванні даних. Саме у цій частині доцільно реалізовувати аудіоінтерфейси генератори тактових сигналів модулі керування потоками даних а також блоки буферизації що забезпечують безперервність відтворення аудіосигналу. Вбудована пам'ять FPGA дозволяє створювати внутрішні буфери для тимчасового зберігання аудіоданих що надходять із зовнішніх джерел або генеруються всередині системи. Процесорна підсистема що інтегрована у кристал використовується для виконання допоміжних функцій пов'язаних із керуванням системою обробленням команд користувача та налаштуванням параметрів роботи медіаплеєра.

Взаємодія між процесорною та логічною частинами здійснюється через високошвидкісні внутрішні шини що забезпечують обмін даними між різними

					КВРКІ.22045.22.03.15 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

компонентами системи без значних затримок. Така організація дозволяє розподілити навантаження між апаратною та програмною частинами що підвищує загальну ефективність функціонування пристрою.

Плата містить розвинену систему підключення зовнішніх компонентів що реалізована через різноманітні роз'єми та інтерфейси. До них належать універсальні порти введення та виведення що дозволяють підключати додаткові модулі сенсори або виконавчі пристрої. Також на платі передбачені інтерфейси для підключення пам'яті що дає змогу зберігати аудіодані великого обсягу та забезпечувати їх подальше відтворення. Наявність комунікаційних портів дозволяє організувати обмін даними з іншими пристроями або комп'ютером що використовується для завантаження конфігурації FPGA та керування системою. Особливе значення для реалізації медіаплеєра має структура підключення аудіокодека до FPGA.

Взаємодія з аудіокодеком здійснюється через спеціалізований цифровий інтерфейс який передбачає використання сигналів тактування синхронізації та передавання даних. Для забезпечення коректної роботи аудіосистеми необхідно організувати точну синхронізацію між FPGA та кодеком що досягається шляхом використання відповідних тактових сигналів та узгодження режимів роботи обох компонентів. Окрім аудіопідсистеми плата містить засоби введення користувача що представлені кнопками та перемикачами які можуть використовуватися для керування режимами роботи медіаплеєра. Виведення інформації може здійснюватися за допомогою світлодіодних індикаторів або дисплейних модулів що дозволяє реалізувати базовий інтерфейс взаємодії з користувачем. Це створює можливість побудови повноцінної системи керування відтворенням аудіоінформації без використання додаткових зовнішніх пристроїв.

Саме використання зовнішньої пам'яті є критично важливим для задачі відтворення аудіо оскільки дозволяє реалізувати ефективну буферизацію поточкових даних та уникнути втрат інформації при нерівномірному доступі до джерела даних. Для прискорення обміну між пам'яттю та обчислювальними

блоками можуть застосовуватися механізми прямого доступу до пам'яті що зменшує навантаження на процесорну підсистему та підвищує загальну продуктивність системи. Окрему роль у забезпеченні стабільної роботи відіграє система генерації тактових сигналів яка реалізується за допомогою вбудованих модулів фазового автопідлаштування частоти та дозволяє формувати декілька синхронних тактових доменів для різних частин системи. Це є необхідним для узгодження роботи аудіоінтерфейсу підсистеми пам'яті та логічних модулів обробки сигналів. Така архітектура дозволяє реалізувати гібридний підхід до побудови медіаплеєра коли високорівневе керування виконується програмно а критичні за часом операції обробляються апаратно в межах FPGA що забезпечує оптимальне поєднання гнучкості та продуктивності системи.

2.3 Апаратні інтерфейси та роз'єми плати Terasic DE1 SoC

Плата Terasic DE1 SoC містить розвинений набір апаратних інтерфейсів та роз'ємів що забезпечують підключення зовнішніх пристроїв та організацію повноцінного функціонування програмно технічного засобу медіаплеєра на FPGA. Наявність великої кількості портів введення та виведення дозволяє реалізувати гнучку систему взаємодії між логічною частиною FPGA та периферійними компонентами що є необхідним для оброблення та відтворення аудіоданих у реальному режимі часу. Ключовим елементами плати є універсальні цифрові порти введення та виведення що виведені на спеціальні роз'єми для підключення зовнішніх модулів. Порти які здійснюють передавання цифрових сигналів між FPGA та зовнішніми пристроями включаючи додаткові аудіомодулі датчики або інші елементи цифрової системи. Використання таких портів відкриває можливість розширення функціональності медіаплеєра та адаптації його до різних умов експлуатації. Також це забезпечує швидкий обмін даними між окремими компонентами системи без затримок, що є критично важливим для якісного відтворення аудіосигналів.

					КВРКІ.22045.22.03.15 ПЗ	Арк. 27
Зм.	Арк.	№ докум.	Підпис	Дата		

На рисунку 2.3 показані всі роз'єми і їх назви для плати Terasic DE1 SOC.

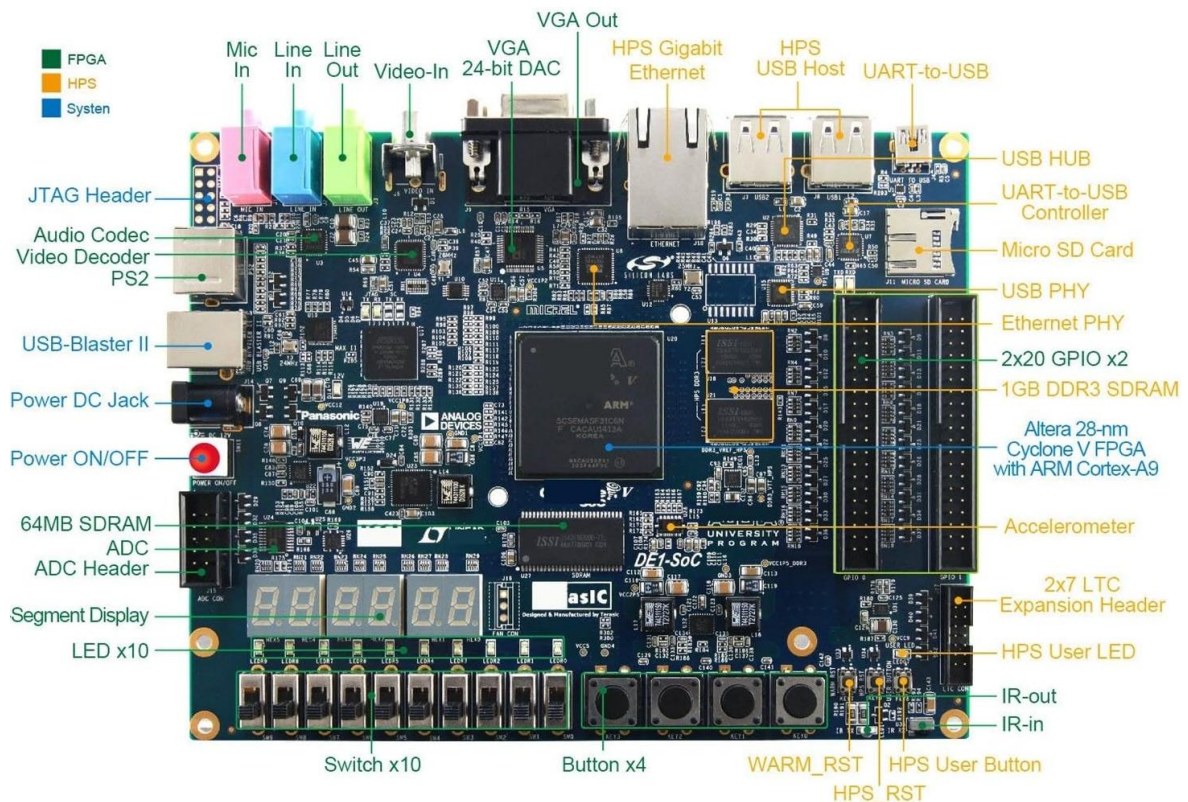


Рисунок 2.3 - Назва всіх роз'ємів плати Terasic DE1 SOC

<https://www.mouser.co.il/new/terasic-technologies/terasic-de1-soc/>

Для реалізації аудіопідсистеми важливе значення має наявність спеціалізованого аудіоінтерфейсу що забезпечує взаємодію FPGA з вбудованим аудіокодеком. Цей інтерфейс включає сигнали передавання цифрових аудіоданих сигнали синхронізації та тактові сигнали що забезпечують узгоджену роботу всіх компонентів аудіотракту.

Передавання даних здійснюється у послідовному форматі що дозволяє ефективно організувати обмін інформацією між FPGA та аудіокодеком із високою точністю синхронізації. На платі також реалізовані інтерфейси підключення зовнішньої пам'яті що дозволяють зберігати аудіофайли та забезпечувати їх подальше відтворення. Використання таких інтерфейсів створює можливість організації локального сховища даних що є важливим для автономної роботи медіаплеєра. Наявність роз'ємів для підключення карт

пам'яті що дозволяє використовувати змінні носії інформації для завантаження аудіоданих. Для взаємодії з користувачем плата оснащена кнопками та перемикачами що дозволяють задавати режими роботи медіаплеєра змінювати параметри відтворення та здійснювати керування системою без використання зовнішніх пристроїв. Світлодіодні індикатори що розташовані на платі використовуються для відображення поточного стану системи та можуть застосовуватися для налагодження роботи цифрових модулів. Додатково передбачено семисегментні індикатори що дозволяють відображати числову інформацію наприклад значення параметрів або режими роботи пристрою.

Плата також містить інтерфейси для підключення відеовиходів та комунікаційних портів що дозволяють організувати взаємодію з іншими пристроями або комп'ютерними системами. Наявність мережевих інтерфейсів забезпечує можливість передавання даних через мережу що може бути використано для віддаленого керування медіаплеєром або оновлення його конфігурації. Крім цього на платі реалізовані інтерфейси підключення до персонального комп'ютера що використовуються для завантаження конфігурації FPGA та налагодження роботи системи. Система тактування плати представлена генераторами сигналів що формують базові частоти для роботи логічних модулів FPGA та периферійних пристроїв.

Плата Terasic DE1 SoC містить інтегровану підсистему взаємодії з комп'ютером через інтерфейс JTAG який використовується для програмування конфігурації FPGA та налагодження апаратної логіки. Це забезпечує можливість оперативного завантаження проєктів у кристал та проведення діагностики роботи окремих цифрових модулів під час розроблення медіаплеєра. Також підтримується інтерфейс USB який може використовуватися для передачі даних та взаємодії з зовнішніми пристроями що розширює можливості системи у частині завантаження аудіофайлів та керування конфігурацією пристрою.

Елементом апаратної структури є наявність відеоінтерфейсів таких як VGA та HDMI що дозволяють організувати виведення графічної інформації або

створення користувацького інтерфейсу медіаплеєра. Хоча основний акцент у даній роботі зроблено на аудіопідсистемі ці інтерфейси можуть бути використані для візуалізації стану відтворення або відображення службової інформації що підвищує зручність експлуатації системи. Також на платі присутній мережевий Ethernet інтерфейс який відкриває можливість підключення до локальної мережі та реалізації функцій віддаленого доступу до медіаплеєра або передачі аудіоданих через мережеві протоколи.

Використання стабільних джерел тактових сигналів забезпечує коректну роботу аудіоінтерфейсів та узгодження швидкості передавання даних між різними компонентами системи. Це є важливою умовою для забезпечення якісного відтворення аудіосигналу без спотворень та затримок. Розглянутий набір апаратних інтерфейсів та роз'ємів плати Terasic DE1 SoC створює всі необхідні умови для реалізації програмно технічного засобу медіаплеєра на FPGA оскільки дозволяє організувати повноцінний цифровий аудіотракт забезпечити взаємодію з користувачем та реалізувати ефективне керування процесом відтворення аудіоданих.

2.4 Аудіопідсистема плати Terasic DE1 SoC та її використання на FPGA

Складовим елементом плати Terasic DE1 SoC є інтегрована аудіопідсистема яка забезпечує можливість відтворення звукової інформації та відіграє ключову роль у реалізації програмно технічного засобу медіаплеєра на FPGA. Основою аудіопідсистеми є вбудований аудіокодек що виконує перетворення цифрових аудіоданих у аналоговий сигнал придатний для подальшого відтворення за допомогою зовнішніх акустичних пристроїв. Використання апаратного аудіокодека дозволяє значно спростити реалізацію медіаплеєра оскільки більшість операцій пов'язаних із перетворенням сигналів виконується на рівні спеціалізованого апаратного компонента. Взаємодія між програмованою логічною частиною FPGA та аудіокодеком здійснюється через

цифровий аудіоінтерфейс що забезпечує передавання аудіоданих а також сигналів синхронізації та керування.

Для коректної роботи аудіотракту необхідно забезпечити точне узгодження частот тактування та послідовності передавання бітів що досягається шляхом формування відповідних сигналів у логічній частині FPGA. У межах реалізації медіаплеєра це означає створення спеціалізованих модулів що відповідають за генерацію тактових сигналів синхронізацію потоків даних та керування передаванням аудіоінформації до аудіокодека. Аудіокодек плати підтримує роботу з цифровими аудіоданими стандартних форматів що дозволяє використовувати його для відтворення різних типів аудіофайлів після їх попереднього оброблення у логічній частині FPGA. Передавання аудіоданих здійснюється у вигляді послідовного потоку що синхронізується з тактовими сигналами що генеруються системою. Це забезпечує стабільність відтворення та відсутність спотворень звукового сигналу при роботі у режимі реального часу. Для керування режимами роботи аудіокодека використовується окремий інтерфейс що дозволяє задавати параметри відтворення такі як рівень гучності режим роботи каналів частота дискретизації та інші характеристики аудіосигналу.

Реалізація цього інтерфейсу у логічній частині FPGA передбачає створення модуля керування що формує відповідні сигнали та передає їх до аудіокодека у визначеній послідовності. Це забезпечує можливість гнучкого налаштування параметрів відтворення аудіоінформації відповідно до вимог користувача або умов функціонування системи. Окрім безпосереднього відтворення аудіосигналу аудіопідсистема дозволяє організувати зворотний канал для приймання звукової інформації що може використовуватися для реалізації додаткових функцій обробки сигналів або створення інтерактивних мультимедійних систем. Наявність такого функціоналу розширює можливості використання плати та створює передумови для реалізації складніших цифрових пристроїв. Особливістю використання аудіопідсистеми у складі медіаплеєра є необхідність

					КВРКІ.22045.22.03.15 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

забезпечення безперервності потоку аудіоданих що досягається шляхом застосування буферизації у внутрішній пам'яті FPGA. Буфери дозволяють компенсувати можливі затримки у передаванні даних та забезпечити рівномірне відтворення звукового сигналу без переривань.

Аудіопідсистеми значною мірою залежить від правильної організації тактових доменів у межах FPGA оскільки будь яка нестабільність або розузгодження частот може призводити до появи спотворень у звуковому сигналі або втрати частини аудіоданих. Саме тому у процесі проектування медіаплеєра особлива увага приділяється формуванню єдиної системи синхронізації яка забезпечує узгоджену роботу всіх блоків що беруть участь у формуванні аудіопотоку. Важливим аспектом також є реалізація правильного формату представлення аудіоданих всередині логічної частини FPGA оскільки більшість аудіокодеків працює з фіксованою розрядністю та певними стандартними форматами передавання даних. Це вимагає попереднього перетворення або узгодження формату даних що надходять із пам'яті або процесорної підсистеми до формату який підтримується аудіокодеком. У результаті формується уніфікований цифровий аудіотракт який забезпечує коректну передачу сигналу від джерела даних до вихідного перетворювача. використання буферизації у поєднанні з потоковою обробкою даних дозволяє реалізувати стійку до зовнішніх затримок систему відтворення яка здатна підтримувати безперервний аудіопотік

Реалізація механізмів буферизації є важливою складовою проектування медіаплеєра та потребує ретельного налаштування параметрів роботи системи. Розглянута аудіопідсистема плати Terasic DE1 SoC створює необхідні умови для реалізації повноцінного програмно технічного засобу медіаплеєра на FPGA оскільки забезпечує всі необхідні функції для перетворення цифрових аудіоданих у звуковий сигнал та їх відтворення у реальному режимі часу. Використання можливостей аудіокодека у поєднанні з апаратними ресурсами FPGA створює ефективну цифрову систему відтворення аудіоінформації.

2.5 Організація зберігання та зчитування аудіоданих на FPGA

Реалізація програмно технічного засобу медіаплеєра на FPGA передбачає ефективну організацію процесів зберігання та зчитування аудіоданих що є необхідною умовою для забезпечення стабільного відтворення звукової інформації у реальному режимі часу. Для досягнення цієї мети використовується поєднання зовнішніх носіїв інформації та внутрішніх ресурсів пам'яті програмованої логічної інтегральної схеми що дозволяє організувати безперервний потік цифрових аудіоданих до аудіопідсистеми. Зберігання аудіофайлів у межах даної системи може здійснюватися із використанням зовнішніх носіїв таких як карти пам'яті або зовнішні модулі пам'яті що підключаються до відповідних інтерфейсів плати Terasic DE1 SoC. Це дозволяє використовувати аудіофайли значного обсягу та забезпечує можливість гнучкого оновлення вмісту без необхідності зміни апаратної частини пристрою.

Процес зчитування аудіоданих із зовнішнього носія передбачає послідовне отримання цифрових значень що відповідають відлікам звукового сигналу та їх подальшу передачу до внутрішніх модулів обробки. Для забезпечення безперервності відтворення використовується механізм буферизації що реалізується за допомогою внутрішньої пам'яті FPGA. Буфер виконує роль проміжного сховища даних у якому тимчасово зберігаються аудіосемпли перед їх передаванням до аудіокодека. Така організація дозволяє компенсувати можливі затримки при зчитуванні даних із зовнішніх джерел та забезпечити рівномірний потік інформації що надходить до аудіопідсистеми. Розмір буфера та швидкість його заповнення визначаються параметрами аудіосигналу та можливостями апаратної платформи. При цьому важливим є правильний вибір глибини буфера що дозволяє уникнути як переповнення так і спустошення буферної пам'яті що може призвести до появи спотворень або переривання звуку.

Зчитування аудіоданих у логічній частині FPGA організовується у вигляді послідовного процесу що синхронізується із тактовими сигналами системи. Кожен відлік аудіосигналу передається у визначений момент часу що забезпечує правильну часову структуру звукового потоку. Для цього використовуються спеціалізовані модулі керування що координують процес читання даних із буфера та їх передачу до аудіоінтерфейсу. У більш складних реалізаціях застосовуються кінцеві автомати станів що визначають порядок переходів між етапами зчитування обробки та передавання даних. Такий підхід дозволяє чітко контролювати часові параметри та забезпечує стабільну роботу системи навіть при змінному навантаженні. Важливою складовою є узгодження швидкості зчитування даних із частотою дискретизації аудіосигналу що дозволяє уникнути спотворень при відтворенні. Це досягається шляхом використання генераторів тактових сигналів та схем фазового автопідлаштування частоти які забезпечують точне формування часових інтервалів між окремими аудіосемплами. Крім того застосовується механізм керування потоком даних що дозволяє динамічно регулювати швидкість передачі інформації залежно від стану буфера. У межах реалізації медіаплеєра також враховується формат аудіоданих що визначає спосіб їх представлення та оброблення.

Використання незжатих форматів що містять послідовність цифрових відліків звукового сигналу. Такі дані можуть бути безпосередньо передані до аудіокодека після відповідної організації потоку. У випадку використання інших форматів необхідно реалізувати додаткові модулі декодування що виконують перетворення даних у форму придатну для відтворення. Реалізація таких модулів на FPGA дозволяє виконувати обробку у паралельному режимі що значно підвищує продуктивність системи та зменшує затримки. Організація процесу зберігання та зчитування аудіоданих також включає механізми керування доступом до пам'яті та синхронізацію роботи різних модулів системи. Це забезпечує узгоджену роботу всіх компонентів медіаплеєра та дозволяє досягти стабільного функціонування у режимі реального часу. Використання зовнішньої

синхронної динамічної пам'яті або інших типів зовнішніх модулів пам'яті доступ до даних координується через спеціалізований контролер пам'яті що реалізує черги запитів та арбітраж доступу між різними блоками системи. Такий підхід уникає конфліктів доступу та забезпечує ефективне використання пропускної здатності пам'яті.

У сучасних реалізаціях застосовується принцип подвійної буферизації при якому один буфер використовується для зчитування даних тоді як інший паралельно заповнюється новими даними із зовнішнього джерела. Після завершення обробки відбувається перемикання буферів що дозволяє забезпечити безперервність потоку без пауз. Така організація значно підвищує надійність роботи системи та зменшує ризик виникнення затримок. Важливим аспектом є також використання принципів потокової обробки даних коли аудіосемпли передаються через послідовність апаратних модулів без необхідності їх повного накопичення в одному блоці пам'яті. Кожен модуль виконує окрему функцію наприклад фільтрацію нормалізацію або перетворення формату і передає результат далі по конвеєру.

Реалізуючи високопродуктивну обробку даних за рахунок паралельності є однією з ключових переваг FPGA. Взаємодія між модулями зчитування обробки та передавання даних реалізується через внутрішні сигнальні лінії FPGA що забезпечують високошвидкісний обмін інформацією з мінімальними затримками. Використання апаратних шин та інтерфейсів дозволяє організувати ефективну передачу даних між компонентами системи без значного навантаження на керуючу логіку. При цьому важливо забезпечити правильну синхронізацію сигналів та уникнення метастабільних станів що можуть виникати при перетині різних тактових доменів.

Організація зберігання та зчитування аудіоданих створює основу для ефективної роботи програмно технічного засобу медіаплеєра на FPGA оскільки дозволяє забезпечити безперервність аудіопотоку точну синхронізацію та високу якість відтворення звукової інформації у межах розроблюваної цифрової

системи а також відкриває можливості для подальшого розширення функціональності системи шляхом додавання нових модулів обробки та підтримки різних форматів аудіоданих.

2.6 Функціональні можливості середовища Intel Quartus Prime при розробці медіаплеєра на FPGA

Розробка програмно-апаратного забезпечення для FPGA-пристроїв, зокрема медіаплеєра, неможлива без використання спеціалізованого середовища проектування, яке забезпечує повний цикл створення, аналізу та налагодження цифрових систем. Одним із найбільш поширених інструментів такого типу є програмне середовище Intel Quartus Prime, яке призначене для розробки, синтезу, моделювання та конфігурації проєктів для FPGA, включаючи сімейство Cyclone V FPGA.

На рисунку 2.4 показано середовище Quartus.



Рисунок 2.4 - Quartus 2 середовище для розробки проєктів під цифрові мікросхеми

Дане середовище виступає як інтегрована система проєктування, що поєднує в собі редактори апаратного опису, компілятор, засоби аналізу часових характеристик, а також інструменти програмування кристалу. Однією з ключових функцій середовища є підтримка мов апаратного опису, зокрема Verilog та VHDL, які дозволяють описувати структуру і поведінку цифрових систем на рівні регістрів та логічних елементів. Завдяки цьому розробник має можливість створювати складні функціональні блоки медіаплеєра, такі як декодери аудіо та відео потоків, модулі буферизації, інтерфейси введення та виведення, а також блоки керування відтворенням. Мова Verilog зазвичай використовується для більш компактного опису логіки та швидкого прототипування, тоді як VHDL забезпечує строгішу типізацію і краще підходить для великих структурованих проєктів. Важливою складовою середовища є процес компіляції, який включає кілька етапів, серед яких синтез логічної схеми, оптимізація, розміщення елементів у структурі FPGA та трасування з'єднань між ними.

Опис, створений мовами HDL, перетворюється у логічну схему, що складається з базових елементів, таких як логічні вентиля, тригери та блоки пам'яті. Далі виконується етап розміщення та трасування, під час якого визначається фізичне розташування елементів у кристалі FPGA та оптимальні маршрути сигналів між ними. Це має критичне значення для забезпечення необхідної швидкодії медіаплеєра, особливо при обробці потокового відео та аудіо. Середовище надає потужні засоби аналізу часових характеристик, які дозволяють визначити затримки сигналів, частоту роботи системи та перевірити коректність синхронізації між різними модулями. Оскільки синхронізація аудіо та відео потоків безпосередньо впливає на якість відтворення. Інструменти таймінг-аналізу дозволяють виявити критичні шляхи та оптимізувати їх для досягнення стабільної роботи системи.

Середовище також підтримує функціональне та часово-залежне моделювання, що дозволяє перевірити правильність роботи розроблених

					КВРКІ.22045.22.03.15 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

модулів ще до їх фізичної реалізації на FPGA. Це значно зменшує кількість помилок та спрощує процес налагодження. Для цього використовуються вбудовані засоби симуляції або інтеграція з зовнішніми симуляторами, що працюють із мовами Verilog та VHDL. Окрему роль відіграють засоби інтеграції апаратних компонентів, зокрема використання вбудованих IP-ядер, які дозволяють швидко реалізувати стандартні функції, такі як контролери пам'яті, інтерфейси передачі даних або цифрові фільтри. Це значно прискорює розробку медіаплеєра та дозволяє зосередитися на реалізації специфічної логіки обробки мультимедійних даних.

Програмування FPGA, під час якого згенерований конфігураційний файл завантажується у кристал, після чого пристрій починає виконувати задану логіку. Середовище Quartus забезпечує повний цикл розробки від початкового опису системи до її фізичної реалізації, що робить його ключовим інструментом при створенні медіаплеєра на базі FPGA.

Окрім базових етапів компіляції, важливою функціональною складовою середовища є підтримка ієрархічного проектування, що дозволяє розбивати складну систему медіаплеєра на окремі функціональні модулі. Кожен із таких модулів може розроблятися, тестуватися та оптимізуватися незалежно, після чого інтегрується у загальну структуру проєкту. Це особливо актуально для систем обробки мультимедійних даних, де присутні різноманітні компоненти, такі як модулі декодування, керування буферами, синхронізації потоків та взаємодії з периферійними пристроями. Ієрархічний підхід може значно спростити процес розробки та підвищити його надійність.

Середовище також надає розширені можливості для аналізу використання апаратних ресурсів FPGA. Після виконання синтезу розробник отримує детальну інформацію про кількість задіяних логічних елементів, блоків пам'яті, мультиплікаторів та інших апаратних ресурсів. Це оцінює ефективність реалізації та, за необхідності, виконати оптимізацію структури проєкту. У контексті медіаплеєра це має особливе значення, оскільки обробка аудіо та відео

потоків може вимагати значних апаратних ресурсів, і їх раціональне використання безпосередньо впливає на продуктивність системи.

2.7 Процес конфігурації та програмування FPGA на базі Terasic DE1-SoC із використанням середовища Intel Quartus Prime

Виконання процесу конфігурації та програмування програмованої логічної інтегральної схеми, що забезпечує фізичне впровадження розробленої цифрової логіки у апаратну структуру пристрою. У межах даного проєкту використовується відлагоджувальна плата Terasic DE1-SoC, яка побудована на основі FPGA сімейства Cyclone V та поєднує у собі як програмовану логіку, так і процесорну підсистему, що значно розширює функціональні можливості розроблюваної системи. Процес конфігурації FPGA здійснюється із використанням спеціалізованого програмного середовища Intel Quartus Prime, яке забезпечує повний цикл підготовки проєкту до завантаження у кристал.

На рисунку 2.5 показано процес конфігурації FPGA в Quartus.

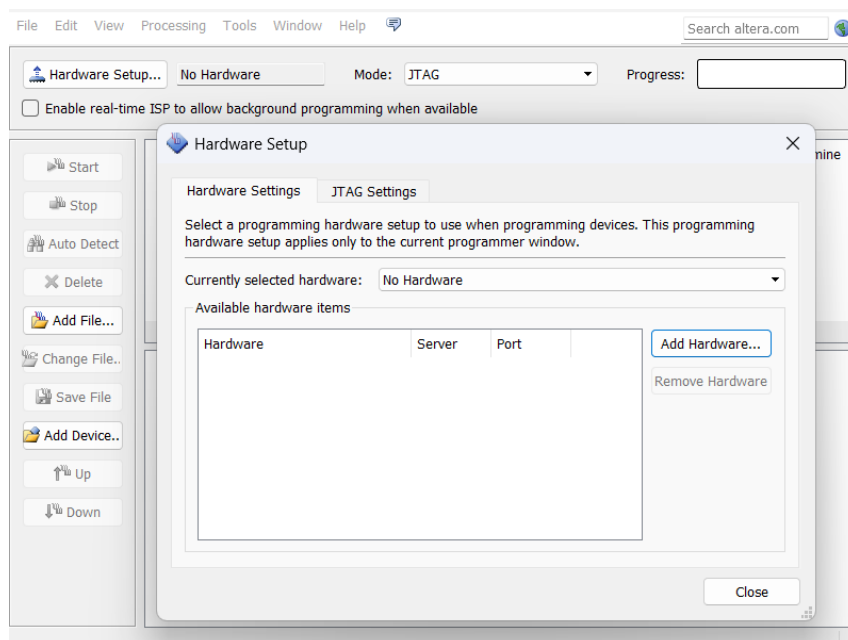


Рисунок 2.5 - Процес конфігурації та програмування FPGA в Quartus

Завершення етапів синтезу, оптимізації, розміщення та трасування формується конфігураційний файл, що містить опис реалізованої цифрової структури. Даний файл є безпосереднім представленням логічної схеми у вигляді бітового потоку, який може бути завантажений у FPGA для виконання. Завантаження конфігурації у FPGA виконується через інтерфейс програмування, який реалізується за допомогою стандартного протоколу JTAG.

Для цього використовується вбудований програматор плати або зовнішній програмувальний пристрій, що забезпечує передачу конфігураційного файлу з персонального комп'ютера до кристалу FPGA. Після завершення процесу програмування логічна структура пристрою перебудовується відповідно до заданого опису, і система починає функціонувати відповідно до розробленого проєкту медіаплеєра. Важливою особливістю FPGA є її енергозалежність, що означає необхідність повторного завантаження конфігурації після кожного вимкнення живлення. Для вирішення цього завдання у складі плати Terasic DE1-SoC передбачено використання зовнішньої конфігураційної пам'яті, у якій може зберігатися бітовий файл. Після подачі живлення FPGA автоматично зчитує конфігурацію з пам'яті та переходить у робочий стан без необхідності ручного програмування.

У процесі розробки медіаплеєра важливу роль відіграє можливість багаторазового перепрограмування FPGA, що дозволяє вносити зміни у логіку системи, тестувати різні варіанти реалізації та виконувати поетапне налагодження. Середовище Intel Quartus Prime забезпечує зручні інструменти для керування процесом програмування, включаючи вибір цільового пристрою, налаштування параметрів завантаження та контроль успішності виконання операції. Окрему увагу слід приділити взаємодії між програмованою логікою та процесорною підсистемою, яка реалізована у структурі SoC. У випадку використання процесора ARM, інтегрованого у Cyclone V, частина функціональності медіаплеєра може бути реалізована на програмному рівні, тоді як апаратно критичні модулі виконуються у FPGA.

Для забезпечення обміну даними між цими підсистемами використовуються спеціалізовані інтерфейси, що дозволяють організувати ефективну передачу інформації між апаратною та програмною частинами системи. Крім безпосереднього програмування, середовище розробки надає можливості для перевірки коректності конфігурації, включаючи верифікацію бітового потоку та аналіз відповідності апаратній платформі. Це уникає помилок під час завантаження та забезпечити стабільну роботу системи після її запуску. У контексті медіаплеєра це має особливе значення, оскільки будь-які помилки конфігурації можуть призвести до порушення відтворення аудіо або некоректної роботи інтерфейсів.

Середовище розробки дозволяє здійснювати перевірку використання логічних елементів, блоків пам'яті та апаратних множників, що дає змогу оптимізувати структуру системи ще на етапі підготовки до завантаження у кристал. Це актуально для медіаплеєра, де одночасно функціонує велика кількість модулів обробки та передачі аудіоданих, і перевищення доступних ресурсів може призвести до неможливості реалізації проєкту або зниження його продуктивності. Перевірка коректності сигналів керування та обміну даними допомагає впевнитися у правильності функціонування всієї системи в цілому. У разі необхідності використовуються вбудовані засоби апаратного аналізу, які дають змогу відстежувати сигнали безпосередньо у працюючому пристрої, що значно спрощує процес пошуку та усунення помилок.

Процес конфігурації та програмування FPGA є етапом розробки, що забезпечує перехід від логічного опису системи до її фізичної реалізації пристрою. Використання середовища Intel Quartus Prime у поєднанні з апаратною платформою Terasic DE1-SoC може реалізувати ефективний та гнучкий підхід до створення медіаплеєра, забезпечуючи можливість швидкого налагодження, модифікації та оптимізації розроблюваної цифрової системи що забезпечує надійну роботу стабільність обробки аудіоданих та відповідність функціонування пристрою поставленим вимогам. Застосування FPGA-

технологій дозволяє створити функціонально завершений, масштабований і надійний медіаплеєр із високою ефективністю обробки аудіосигналів.

2.8 Висновки до другого розділу

У другому розділу було детально розглянуто процес проєктування програмно технічного засобу реалізації медіаплеєра на FPGA із використанням апаратної платформи Terasic DE1 SoC що базується на програмованій логічній інтегральній схемі Intel Cyclone V. Проведений аналіз дозволив визначити основні характеристики обраної платформи її архітектурні особливості та можливості застосування для побудови цифрової системи відтворення аудіоінформації у режимі реального часу. Опис апаратної структури плати дав змогу сформулювати уявлення про наявні ресурси логічної матриці систему підключення периферійних пристроїв та особливості організації взаємодії між окремими функціональними модулями. У процесі дослідження було розглянуто архітектурну організацію плати що включає логічну частину FPGA та інтегровану процесорну підсистему що забезпечує гнучкість побудови цифрових систем різного рівня складності. Значну увагу приділено аналізу апаратних інтерфейсів та роз'ємів що дозволяють здійснювати підключення зовнішніх пристроїв організувати обмін даними та реалізувати повноцінний аудіотракт. Встановлено що наявність спеціалізованих інтерфейсів забезпечує ефективну взаємодію між програмованою логікою та периферійними компонентами що є необхідною умовою для реалізації медіаплеєра.

					КвРКІ.22045.22.03.15 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

3 ПРОГРАМНО ТЕХНІЧНА РЕАЛІЗАЦІЯ МЕДІАПЛЕЄРА НА FPGA НА БАЗІ ПЛАТИ TERASIC DE1 SoC

3.1 Створення проєкту у середовищі Quartus та початкова підготовка системи

Реалізація програмно технічного засобу медіаплеєра на FPGA починається зі створення нового проєкту у середовищі розроблення Quartus що є основним інструментом для проєктування цифрових систем на базі програмованих логічних інтегральних схем. На початковому етапі виконується запуск середовища після чого відкривається головне вікно у якому обирається створення нового проєкту. Для цього використовується відповідний майстер створення що запускається через меню створення нового проєкту. Після запуску відкривається послідовність налаштувань у якій необхідно поетапно задати всі параметри майбутньої системи.

На першому кроці необхідно вказати робочу директорію у якій будуть зберігатися всі файли проєкту. Це важливий момент оскільки правильна організація структури файлів дозволяє уникнути помилок у подальшій роботі. Також задається назва проєкту що використовується для ідентифікації розробки та автоматично створюється базовий файл верхнього рівня який у подальшому буде містити основну логіку системи. Після цього виконується перехід до наступного кроку де пропонується додати вже існуючі файли у проєкт проте на початковому етапі цей крок пропускається оскільки всі модулі будуть створюватися з нуля. Наступним етапом є вибір цільового пристрою що є одним із найважливіших кроків у всьому процесі.

У відповідному вікні необхідно обрати сімейство мікросхем що відповідає використовуваній платі. У даному випадку обирається сімейство Cyclone V після чого зі списку доступних моделей вибирається конкретна мікросхема що встановлена на платі Terasic DE1 SoC.

					КВРКІ.22045.22.03.15 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

На рисунку 3.1 показано створення проекту.

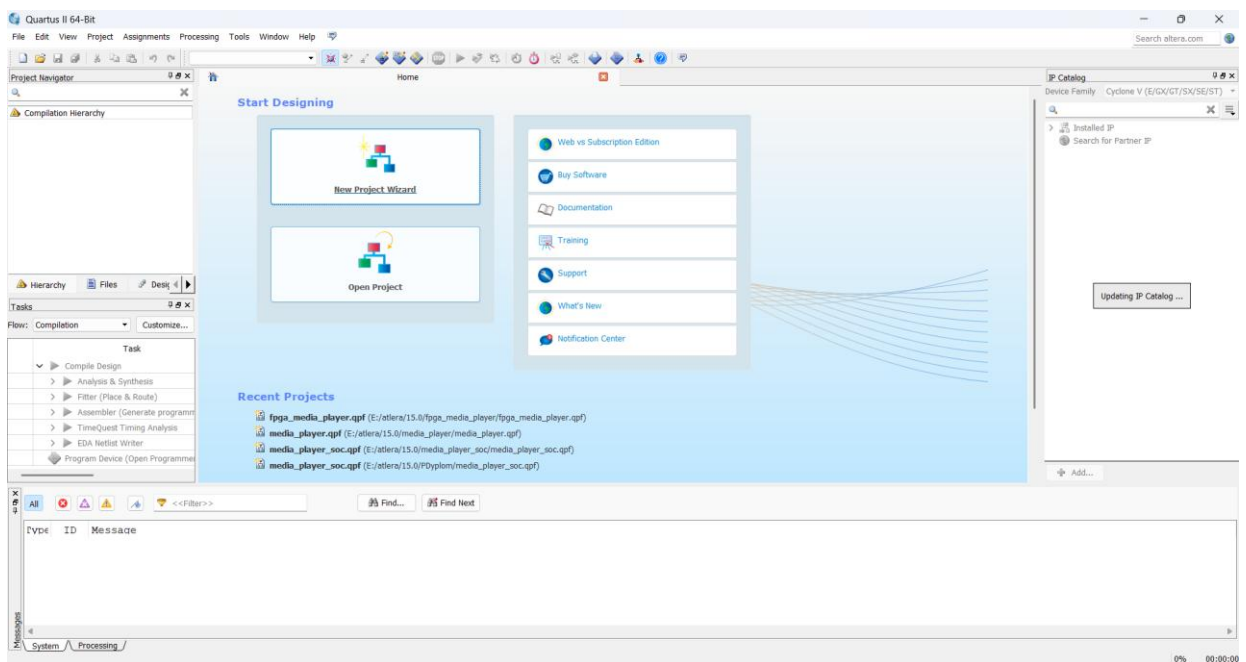


Рисунок 3.1 - Створення проекту у Quartus

Правильний вибір пристрою визначає відповідність розробленої логіки апаратним ресурсам та впливає на результати синтезу і трасування.

Після підтвердження вибору виконується перехід до наступного кроку. Далі пропонується вибір інструментів для синтезу та аналізу проте у більшості випадків залишаються стандартні налаштування що встановлені за замовчуванням.

Це дозволяє використовувати вбудовані засоби середовища без необхідності додаткової конфігурації.

Після завершення налаштування відкривається підсумкове вікно у якому відображаються всі вибрані параметри проекту. Після підтвердження створення проекту формується його структура та відкривається робоче середовище. Після створення проекту виконується додавання нових файлів опису апаратури. Для цього створюються файли мови Verilog у яких буде реалізовано окремі функціональні блоки системи.

На рисунку 3.2 показано меню вибору плати.

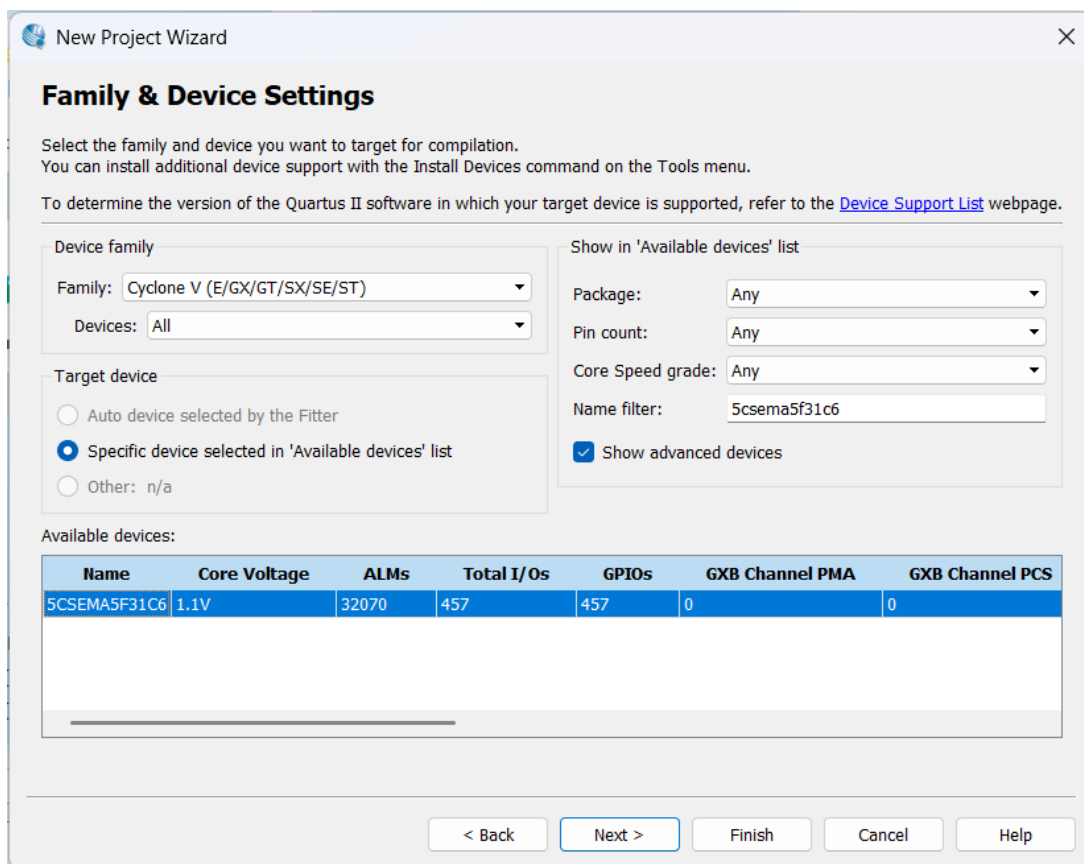


Рисунок 3.2 - Вибір плати для проекту

Кожен файл створюється через меню додавання нового файлу після чого відкривається редактор коду де виконується опис логіки. Спочатку створюються допоміжні модулі такі як обробка кнопок генерація сигналу та керування після чого формується верхній модуль що об'єднує всі компоненти. Паралельно із створенням файлів необхідно виконати налаштування відповідності логічних сигналів фізичним виводам плати. Для цього відкривається редактор призначення виводів у якому кожному сигналу верхнього рівня призначається конкретний контакт мікросхеми. Це забезпечує правильну роботу апаратних елементів таких як кнопки світлодіоди та аудіоінтерфейс. У разі відсутності правильної прив'язки сигнали не будуть передаватися до фізичних пристроїв що зробить систему нефункціональною. Після завершення написання початкового коду виконується перевірка синтаксису що дозволяє виявити помилки у

структурі опису. Далі запускається процес компіляції у якому система виконує синтез логічної структури. На цьому етапі опис мовою Verilog перетворюється у набір логічних елементів що можуть бути реалізовані у межах FPGA. Після синтезу виконується трасування під час якого визначаються фізичні з'єднання між елементами та їх розміщення у структурі мікросхеми. Цей процес враховує обмеження швидкодії та доступні ресурси пристрою. Завершальним етапом є формування конфігураційного файлу що містить повний опис цифрової системи.

3.2 Розроблення модуля обробки кнопок button_controller.v

Для початку нам потрібен файл який буде використовуватись для керування кнопками.

На рисунку 3.3 показано створення файлу для керування кнопками.

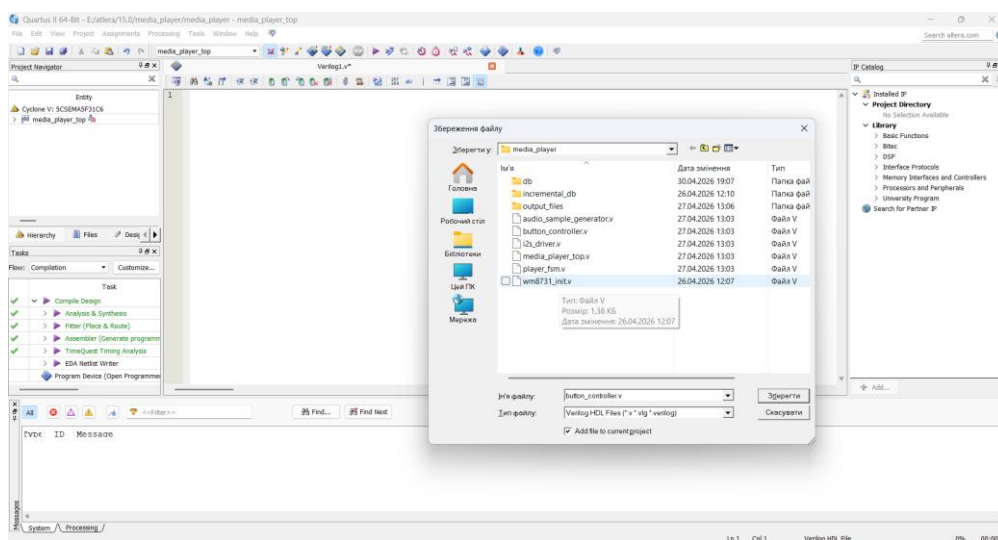


Рисунок 3.3 - Створення файлу button_controller.v

Структура модуля визначається наступним фрагментом коду

```
module button_controller( input clk, input [3:0] KEY, output reg
start = 0, output reg pause = 0, output reg next = 0, output reg back = 0
);
```

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

У цьому описі задаються вхідні та вихідні сигнали. Вхідний сигнал clk використовується для синхронізації роботи модуля із системним тактовим сигналом. Вектор KEY містить стан чотирьох кнопок плати що використовуються для керування. Вихідні сигнали start pause next та back формуються у вигляді коротких імпульсів що передаються до модуля керування. Для коректної роботи необхідно визначати момент натискання кнопки а не її поточний стан. Для цього у модулі використовується допоміжний регістр що зберігає попередній стан кнопок

```
reg [3:0] prev = 4'b1111;
```

Початкове значення встановлюється у стан логічної одиниці що відповідає відпущеним кнопкам. Це дозволяє на першому такті коректно визначити зміну стану при натисканні.

Основна логіка модуля реалізована у послідовному блоці що спрацьовує на кожному фронті тактового сигналу

```
always @(posedge clk) begin start <= 0; pause <= 0; next <= 0; back <= 0;
```

На початку кожного такту всі вихідні сигнали скидаються у нульовий стан. Це необхідно для формування короткого імпульсу тривалістю один такт що виникає лише у момент натискання кнопки. Без цього сигнал залишався б активним протягом усього часу утримання кнопки що призвело б до некоректної роботи системи. Далі виконується перевірка зміни стану кожної кнопки

```
if(KEY[0] == 0 && prev[0] == 1) start <= 1;
```

Цей фрагмент визначає перехід кнопки із відпущеного стану у натиснутий. Якщо у попередньому такті кнопка була не натиснута а у поточному стала натиснута формується сигнал start. Аналогічна логіка застосовується для інших кнопок

```
if(KEY[1] == 0 && prev[1] == 1) pause <= 1; if(KEY[2] == 0 && prev[2] == 1) next <= 1; if(KEY[3] == 0 && prev[3] == 1) back <= 1;
```

Кожна умова відповідає окремій функції керування медіаплеєром. Це дозволяє чітко розділити дії користувача та забезпечити їх незалежну обробку.

стан залежно від вхідних керуючих сигналів. Структура модуля визначається наступним фрагментом

```
module player_fsm( input clk, input start, input pause, input next,
input back, output reg play_enable = 0, output reg [2:0] tone_select =
3'b001 );
```

У цьому описі задаються вхідні сигнали що надходять від модуля обробки кнопок а також вихідні сигнали що використовуються іншими частинами системи. Змінна `play_enable` визначає чи відбувається відтворення звуку а змінна `tone_select` задає номер звукового сигналу.

Початкове значення `tone_select` встановлено рівним одиниці що відповідає першому режиму роботи. Основна логіка роботи модуля реалізована у синхронному блоці що виконується на кожному фронті тактового сигналу

```
always @(posedge clk) begin
```

Це означає що всі зміни стану системи відбуваються синхронно із системним тактом що забезпечує стабільність та передбачуваність роботи.

Обробка сигналу запуску реалізована наступним чином

```
if(start) play_enable <= 1;
```

При надходженні імпульсу `start` система переходить у режим відтворення. Значення `play_enable` встановлюється у одиницю що дозволяє передавати сформований аудіосигнал до аудіопідсистеми. Це є основним механізмом запуску медіаплеєра. Зупинка відтворення реалізована через обробку сигналу паузи

```
if(pause) play_enable <= 0;
```

У цьому випадку сигнал `play_enable` скидається у нуль що призводить до відключення аудіосигналу. Важливо що при цьому інші частини системи продовжують працювати що дозволяє швидко відновити відтворення без повторної ініціалізації.

Перемикання звукових сигналів у прямому напрямку реалізовано у наступному фрагменті

```
if(next) begin if(tone_select == 3'b111) tone_select <= 3'b001; else
tone_select <= tone_select + 1; end
```

У цьому блоці перевіряється чи досягнуто максимального значення. Якщо це так відбувається перехід до початкового значення що забезпечує циклічний режим роботи. У протилежному випадку значення збільшується на одиницю що відповідає переходу до наступного звуку. Така логіка дозволяє організувати безперервний перебір доступних режимів. Перемикання у зворотному напрямку реалізовано аналогічно

```
if(back) begin if(tone_select == 3'b001) tone_select <= 3'b111; else
tone_select <= tone_select - 1; end
```

У цьому випадку при досягненні мінімального значення виконується перехід до максимального що також забезпечує циклічність. У даному модулі відсутні складні стани чи багаторівнева логіка переходів що робить його простим для розуміння та надійним у роботі.

Усі керуючі сигнали обробляються незалежно один від одного що дозволяє уникнути конфліктів між командами користувача. Наприклад одночасне надходження сигналів не призводить до некоректної поведінки оскільки зміна станів виконується послідовно у межах одного тактового циклу. Використання регістрів для збереження стану системи. Це означає що після кожного такту значення змінних play_enable та tone_select зберігаються та використовуються у наступних обчисленнях. Такий підхід дозволяє реалізувати поведінку що залежить від попереднього стану що є характерною рисою послідовних цифрових систем.

3.4 Розроблення модуля генерації аудіосигналу audio_sample_generator.v

Наступним етапом реалізації медіаплеєра стало створення модуля генерації аудіосигналу який відповідає за формування цифрових звукових даних.

На рисунку 3.5 показано створення файлу для генерації аудіосигналу.

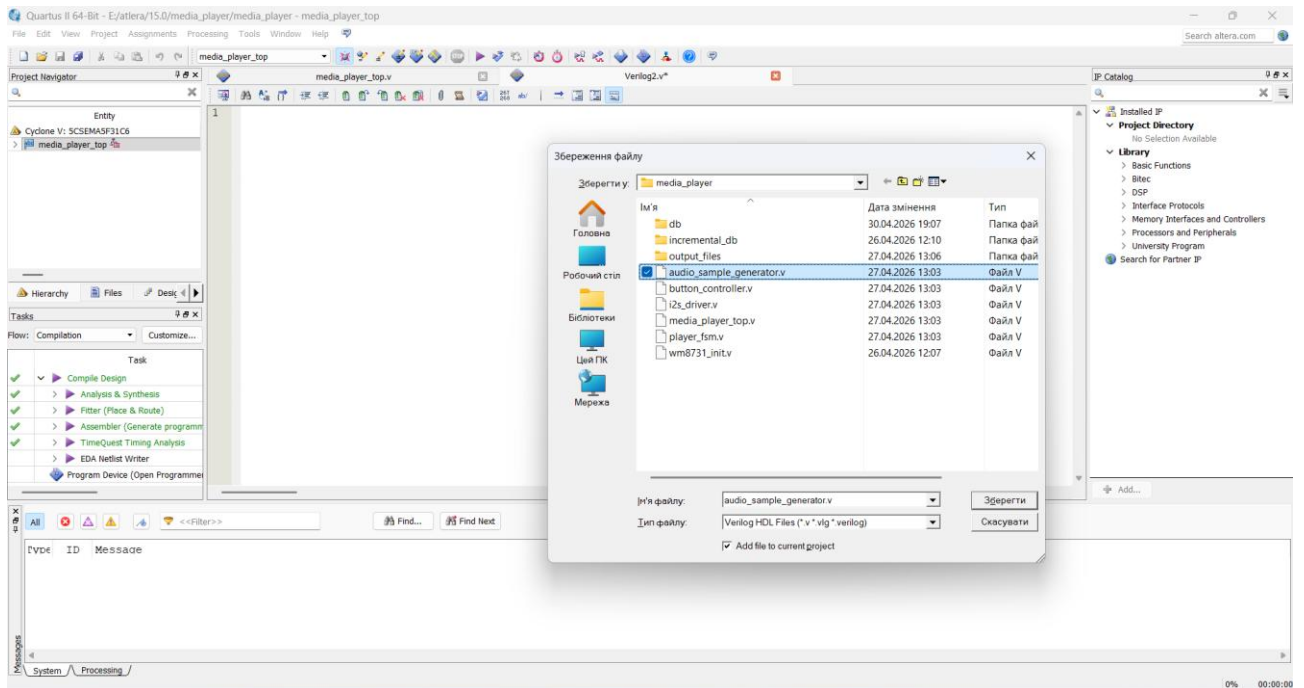


Рисунок 3.5 - Створення файлу audio_sample_generator.v

Саме цей модуль є основою всієї аудіосистеми оскільки він створює сигнал що у подальшому передається до аудіокодека та перетворюється у звук.

У межах даного проекту використовується підхід генерації прямокутного сигналу із різною частотою що дозволяє реалізувати набір звукових сигналів без використання складних алгоритмів оброблення аудіоданих. Структура модуля визначається наступним описом

```
module audio_sample_generator( input clk, input enable, input [2:0]
tone_select, output reg [15:0] sample = 0 );
```

У цьому фрагменті задаються основні сигнали модуля. Вхідний сигнал `clk` використовується для синхронізації роботи. Сигнал `enable` дозволяє вмикати або вимикати генерацію сигналу. Вхід `tone_select` визначає який саме звуковий сигнал буде сформований. Вихід `sample` містить цифрове значення аудіосемплу що передається до наступних блоків системи. Для реалізації генерації використовується внутрішній лічильник та змінна що визначає граничне значення

```
reg [15:0] counter = 0; reg [15:0] limit;
```

Лічильник відповідає за відлік тактів системного сигналу а змінна limit визначає період майбутнього сигналу. Зміна значення limit дозволяє змінювати частоту звуку. Вибір значення limit здійснюється залежно від сигналу tone_select

```
always @(*) begin case(tone_select) 3'b001: limit = 12500; 3'b010: limit = 6250; 3'b011: limit = 4166; 3'b100: limit = 3125; 3'b101: limit = 2500; 3'b110: limit = 2083; 3'b111: limit = 1785; default: limit = 12500; endcase end
```

У цьому фрагменті реалізовано вибір частоти сигналу. Кожне значення tone_select відповідає певній частоті. Наприклад значення 12500 використовується для формування сигналу із частотою близько двох тисяч герц а значення 1785 відповідає значно вищій частоті. Така реалізація дозволяє отримати набір дискретних звукових сигналів що можуть перемикатися під час роботи медіаплеєра. Безпосереднє формування сигналу виконується у синхронному блоці

```
always @(posedge clk) begin if(enable) begin if(counter >= limit) begin counter <= 0; sample <= ~sample; end else counter <= counter + 1; end end
```

Цей блок визначає поведінку лічильника та формування вихідного сигналу. На кожному такті значення counter збільшується. Коли воно досягає значення limit відбувається скидання лічильника та інверсія змінної sample. Саме інверсія створює зміну рівня сигналу що формує прямокутну форму хвилі. Період такого сигналу визначається часом між двома інверсіями а отже залежить від значення limit.

Використання прямокутного сигналу дозволяє реалізувати простий але ефективний спосіб генерації звуку у межах FPGA. Хоча така форма сигналу не є ідеальною з точки зору якості звучання вона достатня для демонстрації принципів роботи аудіосистеми. Сигнал enable дозволяє керувати роботою генератора. У випадку коли він активний відбувається формування сигналу у протилежному випадку стан генератора не змінюється. У даному проєкті цей

сигнал використовується разом із логікою верхнього рівня для реалізації режиму паузи.

3.5 Розроблення модуля передачі аудіоданих i2s_driver.v

Після формування аудіосигналу наступним етапом є його передавання до аудіокодека що виконується через спеціалізований інтерфейс.

На рисунку 3.6 показано створення файлу для передавання аудіосигналу до аудіокодека.

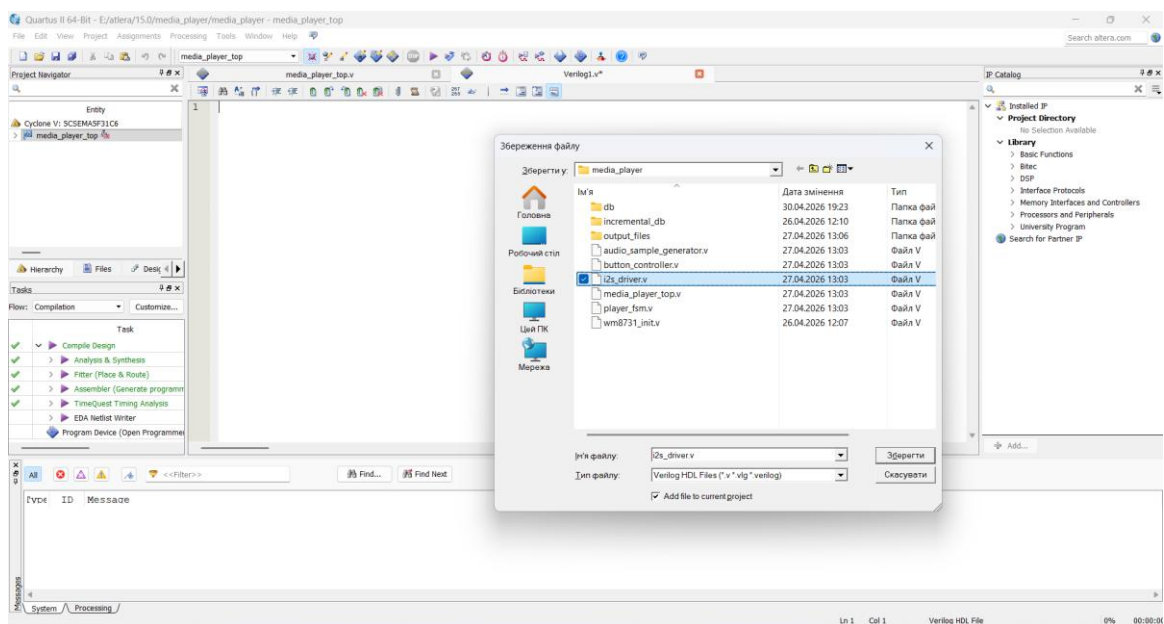


Рисунок 3.6 - Створення файлу i2s_driver.v

У даному проєкті для цього використовується модуль i2s_driver який реалізує послідовну передачу аудіоданих відповідно до принципів роботи цифрових аудіоінтерфейсів. Основною задачею цього модуля є перетворення паралельного цифрового сигналу у послідовний потік бітів із одночасним формуванням сигналів синхронізації.

Структура модуля визначається наступним описом

```
module i2s_driver( input clk, input [15:0] sample, output reg bclk = 0, output reg lrclk = 0, output reg data = 0 );
```

Зм.	Арк.	№ докум.	Підпис	Дата

У цьому фрагменті задаються вхідні та вихідні сигнали. Вхід clk використовується як основний тактовий сигнал системи. Вхід sample містить цифрове значення аудіосемплу що формується у попередньому модулі. Вихідні сигнали bclk lrcclk та data відповідають за передачу даних до аудіокодека. Сигнал bclk використовується як бітовий такт що визначає момент передавання кожного біта. Сигнал lrcclk визначає межі аудіоканалів а сигнал data містить безпосередньо передавані дані. Для реалізації передачі використовується внутрішній лічильник

```
reg [4:0] bit_counter = 0;
```

Ця змінна визначає номер біта що передається у поточний момент часу. Оскільки використовується шістнадцятибітний формат значення лічильника змінюється у відповідному діапазоні. Основна логіка модуля реалізована у синхронному блоці

```
always @(posedge clk) begin bclk <= ~bclk;
```

На кожному фронті тактового сигналу відбувається інверсія сигналу bclk що дозволяє сформувати бітовий такт із частотою що у два рази менша за системну. Це необхідно для організації передачі даних у правильному ритмі. Подальша логіка виконується у момент активного стану

```
bclk if(bclk) begin data <= sample[15 - bit_counter];
```

У цьому рядку виконується вибір конкретного біта із аудіосемплу. Передавання починається зі старшого біта що відповідає стандартній організації передачі цифрових аудіоданих. Змінна bit_counter визначає який саме біт буде переданий у поточний момент часу. Далі відбувається збільшення значення лічильника bit_counter <= bit_counter + 1; Це забезпечує послідовний перехід до наступного біта. Після передачі всіх бітів необхідно повернутися до початку та змінити канал

```
if(bit_counter == 15) begin bit_counter <= 0; lrcclk <= ~lrcclk; end
```

У цьому фрагменті перевіряється чи було передано останній біт. Якщо це так лічильник скидається а сигнал lrcclk змінює свій стан. Це означає перехід до іншого аудіоканалу. Така організація дозволяє синхронізувати передачу даних із роботою аудіокодека та забезпечити коректне відтворення звуку.

Розроблений модуль виконує критично важливу функцію у системі оскільки саме він забезпечує передачу сформованого сигналу до зовнішнього пристрою. Перетворення паралельного сигналу у послідовний потік є необхідною умовою для взаємодії із аудіокодеком що працює за відповідним протоколом. У результаті реалізації цього модуля забезпечується правильна синхронізація та передавання аудіоданих що дозволяє відтворювати сформований сигнал через аудіовихід плати.

3.6 Розроблення модуля ініціалізації аудіокодека `wm8731_init.v`

Після реалізації передачі аудіоданих наступним етапом є забезпечення правильної роботи аудіокодека що встановлений на платі Terasic DE1 SoC.

На рисунку 3.7 показано створення файлу для модифікації аудіокодека.

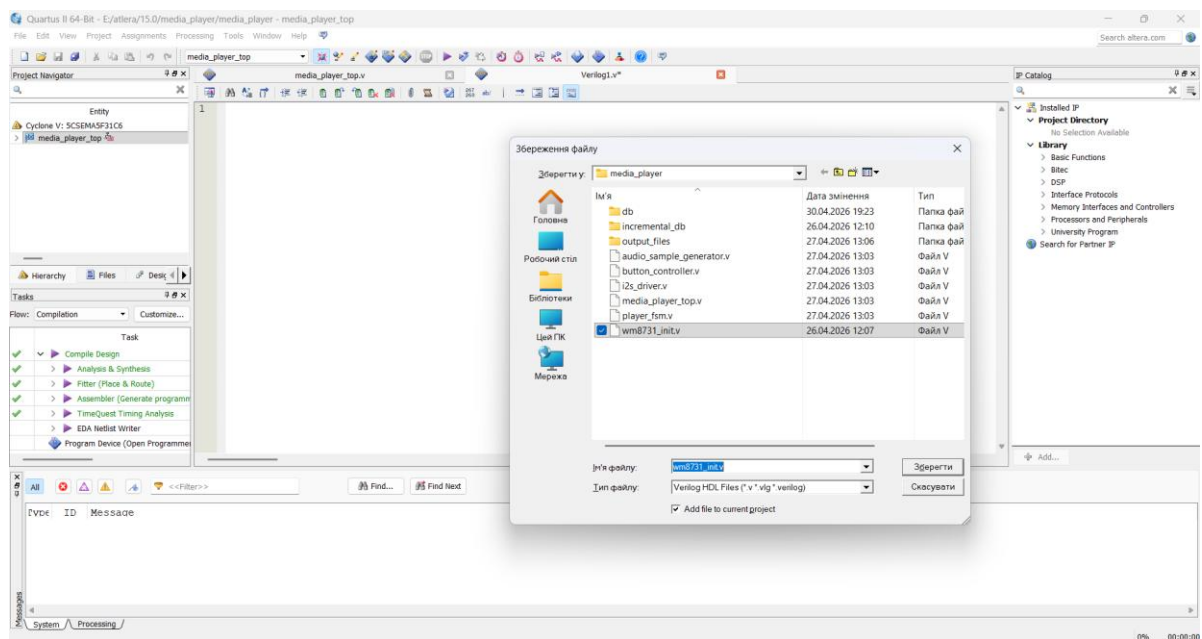


Рисунок 3.7 - Створення файлу `wm8731_init.v`

Для цього необхідно виконати його початкове налаштування шляхом запису конфігураційних параметрів у внутрішні регістри. Ця задача вирішується за допомогою окремого модуля що реалізує обмін даними через послідовний

інтерфейс керування. Без виконання цього етапу аудіокодек не зможе коректно обробляти вхідні цифрові дані та відтворювати звук. Структура модуля визначається наступним описом

```
module wm8731_init(input clk, output i2c_sclk, inout i2c_sdat );
```

У цьому фрагменті задається вхідний тактовий сигнал та сигнали інтерфейсу керування. Сигнал `i2c_sclk` використовується для синхронізації передавання даних а сигнал `i2c_sdat` виконує роль двонапрявленої лінії якій передаються конфігураційні значення. Така організація дозволяє реалізувати обмін даними між FPGA та аудіокодеком. Логіка роботи модуля базується на послідовному передаванні набору команд що визначають параметри функціонування аудіопідсистеми. У процесі ініціалізації формуються значення що відповідають необхідним режимам роботи таким як встановлення формату аудіоданих налаштування частоти дискретизації вибір активних каналів та рівень гучності. Кожна команда передається у вигляді послідовності бітів що синхронізуються із сигналом такту.

В реалізації використовується внутрішній механізм що поетапно формує сигнали керування. Спочатку генерується умова початку передавання після чого послідовно передаються адреса пристрою та дані що записуються у відповідний регістр. Після завершення передачі формується сигнал завершення що фіксує виконання операції. Такий підхід дозволяє забезпечити коректну взаємодію із аудіокодеком. Особливістю цього модуля є те що процес ініціалізації виконується автоматично після подачі тактового сигналу без необхідності додаткового втручання. Це означає що після завантаження конфігурації у FPGA аудіокодек одразу переходить у робочий режим із заданими параметрами. Така організація спрощує загальну структуру системи та дозволяє уникнути складних процедур керування.

У процесі передачі даних важливо забезпечити правильну послідовність сигналів та дотримання часових інтервалів. Для цього використовується синхронізація із тактовим сигналом що дозволяє уникнути помилок під час

обміну даними. У разі порушення цих умов аудіокодек може некоректно сприймати передані значення що призведе до неправильної роботи системи. Розроблений модуль забезпечує повну підготовку аудіокодека до роботи та є необхідною складовою всієї аудіосистеми. Він виконує роль проміжного елемента що з'єднує цифрову логіку FPGA із аналоговою частиною пристрою. У результаті його використання забезпечується правильна обробка аудіоданих та можливість їх відтворення через фізичні виходи плати.

3.7 Розроблення верхнього модуля системи media_player_top.v та інтеграція всіх компонентів

Завершальним етапом реалізації медіаплеєра стало створення верхнього модуля що об'єднує всі раніше розроблені компоненти у єдину цифрову систему.

На рисунку 3.8 показано створення головного файлу для роботи проекту.

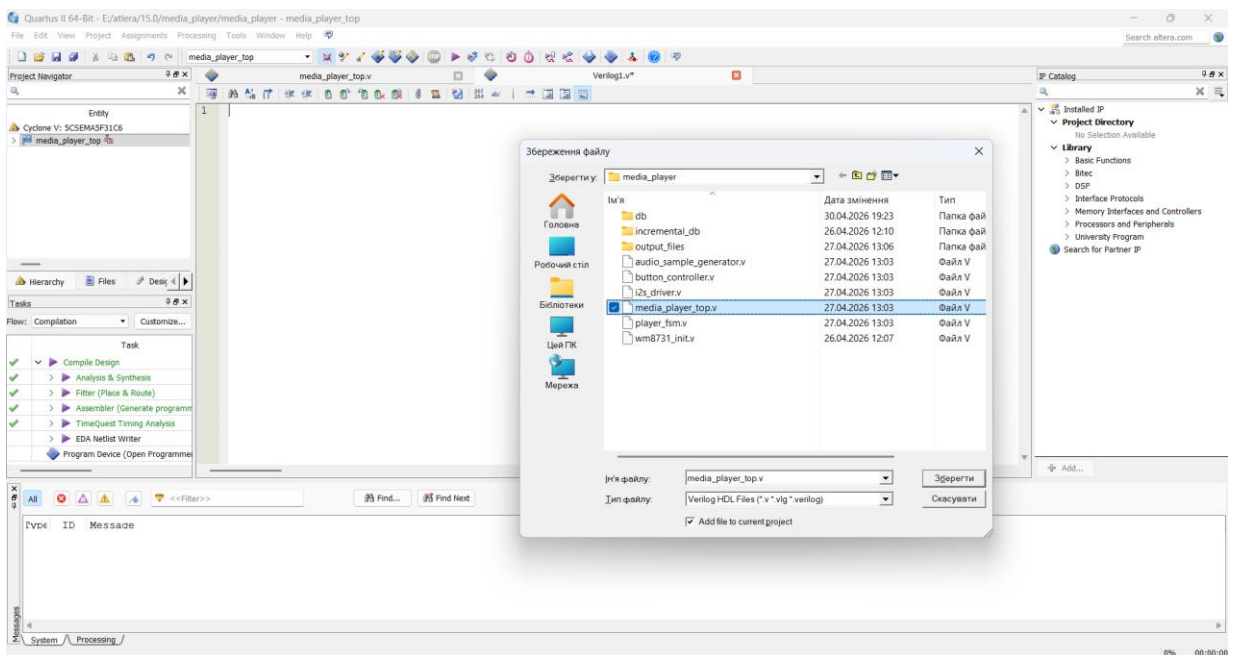


Рисунок 3.8 - Створення файлу media_player_top.v

Саме цей модуль визначає взаємодію між окремими функціональними блоками та забезпечує їх узгоджену роботу. У даного модуля виконано опис усіх

зовнішніх інтерфейсів а також організовано передавання сигналів між внутрішніми модулями.

Структура верхнього модуля починається із визначення вхідних та вихідних сигналів

```
module media_player_top( input CLOCK_50, input [3:0] KEY, output [7:0] LED, output AUD_XCK, output AUD_BCLK, output AUD_DACLK, output AUD_DACDAT, output AUD_I2C_SCLK, inout AUD_I2C_SDAT );
```

У цьому фрагменті задаються всі інтерфейси системи. Тактовий сигнал CLOCK_50 використовується як основа для синхронізації роботи всіх модулів. Вхід KEY містить сигнали від кнопок керування. Вихід LED використовується для індикації стану системи. Інші сигнали забезпечують взаємодію із аудіокодеком включаючи передавання аудіоданих та керування. Далі у модулі оголошуються внутрішні з'єднання що використовуються для обміну даними між компонентами

```
wire start; wire pause; wire next; wire back; wire play_enable; wire [2:0] tone_select; wire [15:0] audio_sample; wire [15:0] sample_out; wire bclk; wire lrclk; wire audio_data; wire i2c_clk; wire i2c_data;
```

Ці сигнали виконують роль каналів зв'язку між модулями. Вони дозволяють передавати керуючі імпульси аудіодані та сигнали синхронізації. Одним із перших кроків є подача тактового сигналу на аудіокодек

```
assign AUD_XCK = CLOCK_50;
```

Цей рядок забезпечує генерацію опорного сигналу що необхідний для роботи аудіопідсистеми. Без нього аудіокодек не зможе виконувати обробку даних. Після цього виконується підключення модуля обробки кнопок

```
button_controller buttons( .clk(CLOCK_50), .KEY(KEY), .start(start), .pause(pause), .next(next), .back(back) );
```

Цей блок перетворює фізичні натискання кнопок у керуючі сигнали що використовуються у подальшій логіці. Отримані імпульси передаються до модуля керування. Далі підключається модуль керування

```
player_fsm player( .clk(CLOCK_50), .start(start), .pause(pause),  
.next(next), .back(back), .play_enable(play_enable),  
.tone_select(tone_select) );
```

Цей модуль визначає режим роботи системи та вибір звукового сигналу. Сигнали play_enable та tone_select використовуються іншими компонентами для формування аудіопотоку. Наступним етапом є підключення генератора аудіосигналу

```
audio_sample_generator tone( .clk(CLOCK_50), .enable(1'b1),  
.tone_select(tone_select), .sample(audio_sample) );
```

Цей модуль формує цифровий сигнал на основі вибраного режиму. Отримане значення передається далі для обробки та відтворення.

```
assign sample_out = play_enable ? audio_sample : 16'd32768;
```

У випадку активного режиму передається сформований сигнал у протилежному випадку встановлюється значення що відповідає тиші. Це забезпечує простий та ефективний спосіб керування відтворенням. Далі виконується підключення модуля передачі аудіоданих

```
i2s_driver audio( .clk(CLOCK_50), .sample(sample_out), .bclk(bclk),  
.lrcclk(lrcclk), .data(audio_data) );
```

Цей модуль перетворює паралельний сигнал у послідовний потік що передається до аудіокодека. Сигнали синхронізації та даних формуються відповідно до вимог аудіоінтерфейсу. Наступним кроком є ініціалізація аудіокодека

```
wm8731_init codec( .clk(CLOCK_50), .i2c_sclk(i2c_clk),  
.i2c_sdat(i2c_data) );
```

Цей модуль забезпечує налаштування параметрів роботи аудіопідсистеми що є необхідним для коректного відтворення звуку. Індикація стану системи реалізована через світлодіоди

```
assign LED[2:0] = tone_select; assign LED[3] = play_enable; assign  
LED[7:4] = 4'b0000;
```

Ці рядки дозволяють відображати поточний режим роботи медіаплеєра що значно спрощує процес тестування та налагодження. Завершальним етапом є підключення сформованих сигналів до зовнішніх виводів

```
assign AUD_BCLK      = bclk; assign AUD_DACLK      = lrcclk; assign
AUD_DACDAT         = audio_data; assign AUD_I2C_SCLK = i2c_clk; assign
AUD_I2C_SDAT = i2c_data;
```

Цей фрагмент забезпечує передачу всіх необхідних сигналів до аудіокодека що дозволяє виконувати відтворення звуку.

3.8 Налаштування виводів FPGA та запуск проєкту на апаратній платформі

Після завершення розроблення всіх функціональних модулів та інтеграції їх у верхньому рівні наступним етапом є забезпечення фізичної взаємодії створеної цифрової системи із апаратними ресурсами плати.

На рисунку 3.9 показано налаштування пінів для плати.

itatu:	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✓	CLOCK_50	Location	PIN_AF14	Yes			
2	✓	KEY[0]	Location	PIN_AA14	Yes			
3	✓	KEY[1]	Location	PIN_AA15	Yes			
4	✓	KEY[2]	Location	PIN_W15	Yes			
5	✓	KEY[3]	Location	PIN_Y16	Yes			
6	✓	LED[0]	Location	PIN_V16	Yes			
7	✓	LED[1]	Location	PIN_W16	Yes			
8	✓	LED[2]	Location	PIN_V17	Yes			
9	✓	LED[3]	Location	PIN_V18	Yes			
10	✓	LED[4]	Location	PIN_W17	Yes			
11	✓	LED[5]	Location	PIN_W19	Yes			
12	✓	LED[6]	Location	PIN_Y19	Yes			
13	✓	LED[7]	Location	PIN_W20	Yes			
14	✓	AUD_BCLK	Location	PIN_A9	Yes			
15	✓	AUD...RCK	Location	PIN_C9	Yes			
16	✓	AUD...DAT	Location	PIN_B8	Yes			
17	✓	AUD...CLK	Location	PIN_B7	Yes			
18	✓	AUD...DAT	Location	PIN_A8	Yes			
19	✓	AUD_XCK	Location	PIN_Y18	Yes			
20	<<new>>	<<new>>	<<new>>					

Рисунок 3.9 - Налаштування пінів для плати

Для цього виконується налаштування виводів програмованої логічної інтегральної схеми що дозволяє зв'язати логічні сигнали описані у кодї із конкретними контактами мікросхеми. Без виконання цього етапу жоден із

розроблених модулів не зможе взаємодіяти із зовнішніми пристроями такими як кнопки світлодіоди та аудіокодек. Налаштування виводів здійснюється у середовищі Quartus через спеціалізований інструмент призначений для роботи із апаратними контактами мікросхеми. Інструмент кожного сигналу верхнього модуля призначається відповідний фізичний контакт згідно зі схемою плати Terasic DE1 SoC.

На цьому етапі необхідно уважно співвіднести всі сигнали із їх апаратними відповідниками оскільки помилки у призначенні можуть призвести до повної непрацездатності системи або некоректної роботи окремих її компонентів. Сигнали керування що надходять від кнопок плати підключаються до відповідних входів модуля через виводи мікросхеми. Аналогічно сигнали що відповідають за індикацію підключаються до світлодіодів що дозволяє відображати стан роботи медіаплеєра. Особливу увагу приділено підключенню аудіопідсистеми що включає сигнали тактування передавання даних та керування. Для кожного із цих сигналів необхідно визначити правильний контакт що відповідає підключенню аудіокодека на платі. Після завершення налаштування виконується повторна компіляція проєкту що враховує нові параметри прив'язки виводів.

На цьому етапі система перевіряє коректність призначення контактів та виконує остаточне формування конфігураційного файлу. У разі відсутності помилок формується файл що містить повний опис логічної структури разом із інформацією про фізичні з'єднання. Наступним кроком є підключення апаратної платформи до персонального комп'ютера із використанням відповідного інтерфейсу програмування. Після встановлення з'єднання відкривається інструмент для завантаження конфігурації у FPGA. У цьому інструменті обирається сформований файл після чого запускається процес програмування. У процесі запису відбувається передавання даних до пам'яті мікросхеми що визначає її подальшу поведінку. Після завершення завантаження конфігурації система починає працювати у відповідності до розробленого проєкту. На цьому

					КВРКІ.22045.22.03.15 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

етапі можна спостерігати реакцію системи на натискання кнопок а також перевірити роботу індикації. При натисканні кнопки запуску активується режим відтворення що супроводжується зміною стану світлодіодів. Перемикання між звуковими сигналами також відображається через індикацію що дозволяє контролювати роботу системи.

Особливе значення має перевірка роботи аудіопідсистеми. Після запуску медіаплеєра аудіокодек отримує необхідні сигнали керування та починає приймати цифрові аудіодані що формуються FPGA. У результаті відбувається відтворення звукових сигналів через аудіовихід плати. Зміна частоти сигналу при перемиканні режимів підтверджує правильність роботи генератора аудіосигналу та модуля керування. У процесі тестування також перевіряється коректність роботи режиму паузи що дозволяє припинити відтворення без порушення функціонування інших компонентів системи.

На рисунках з 3.10 до 3.14 показано результат проекту.

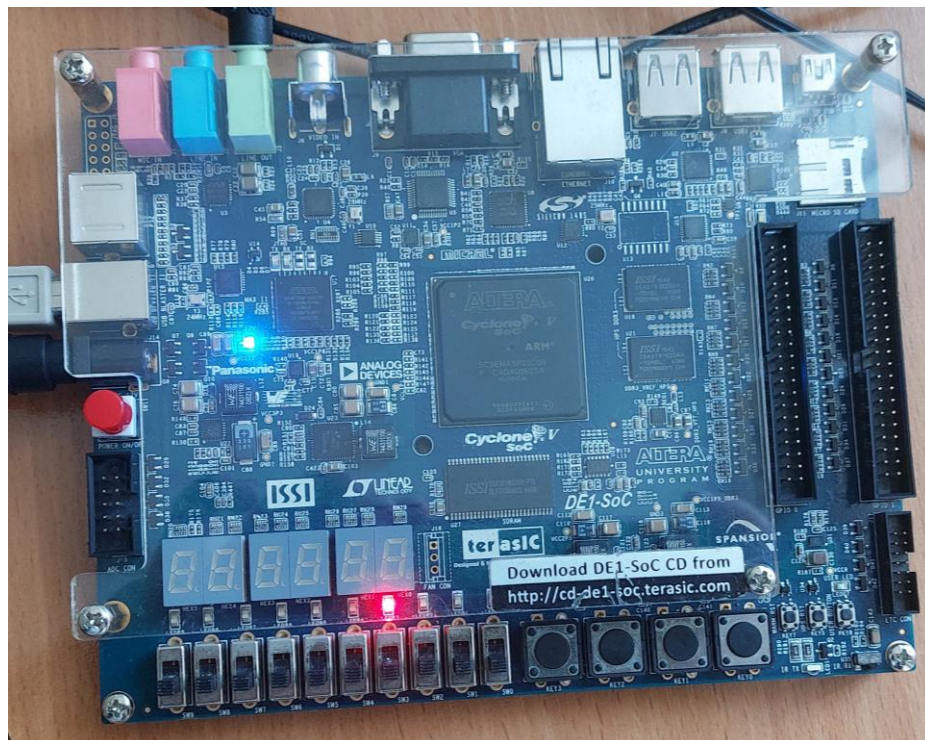


Рисунок 3.10 - Медіаплеєр запущений

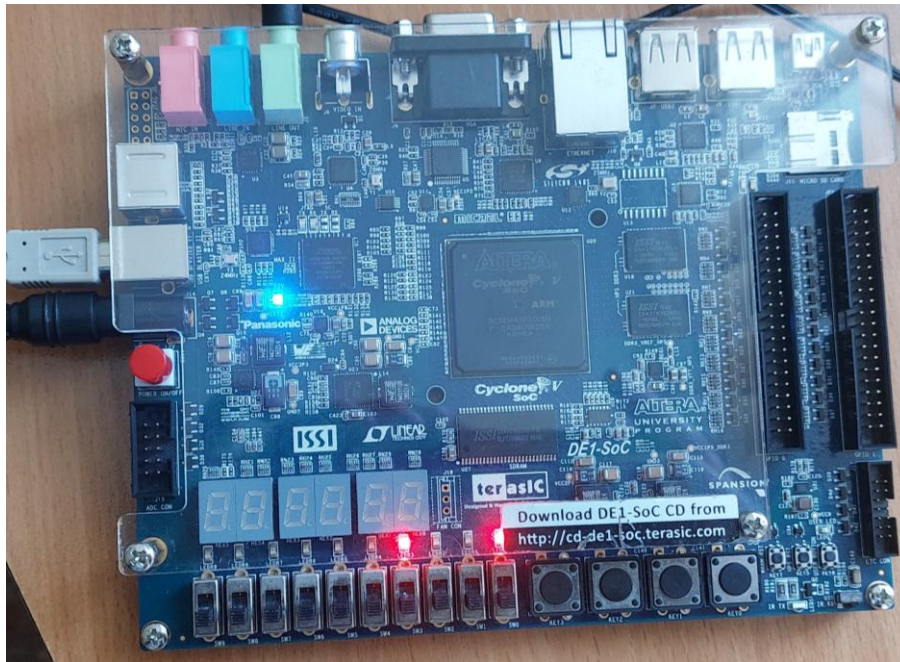


Рисунок 3.11 - Медіаплеєр запуснений і відворює звук номер 1

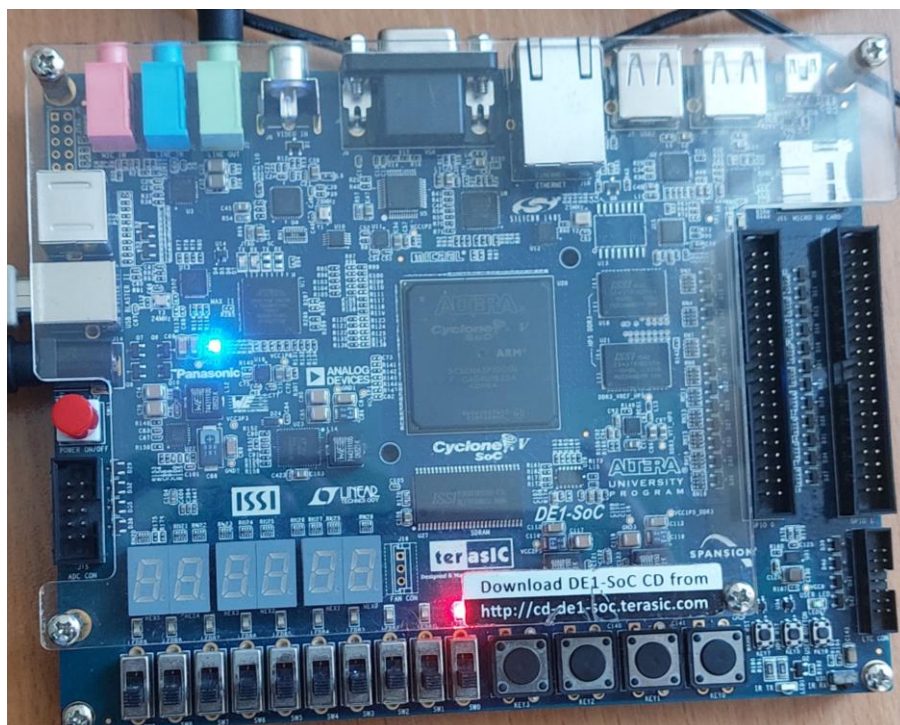


Рисунок 3.12 - Медіаплеєр на паузі звук досі номер 1

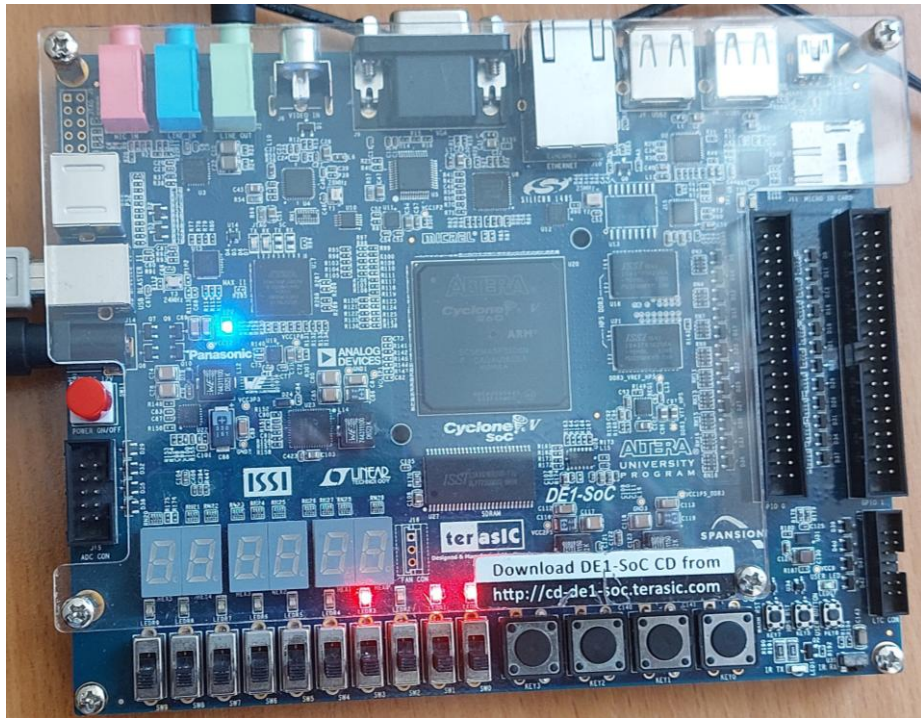


Рисунок 3.13 - Медіаплеєр запущений звук номер 3

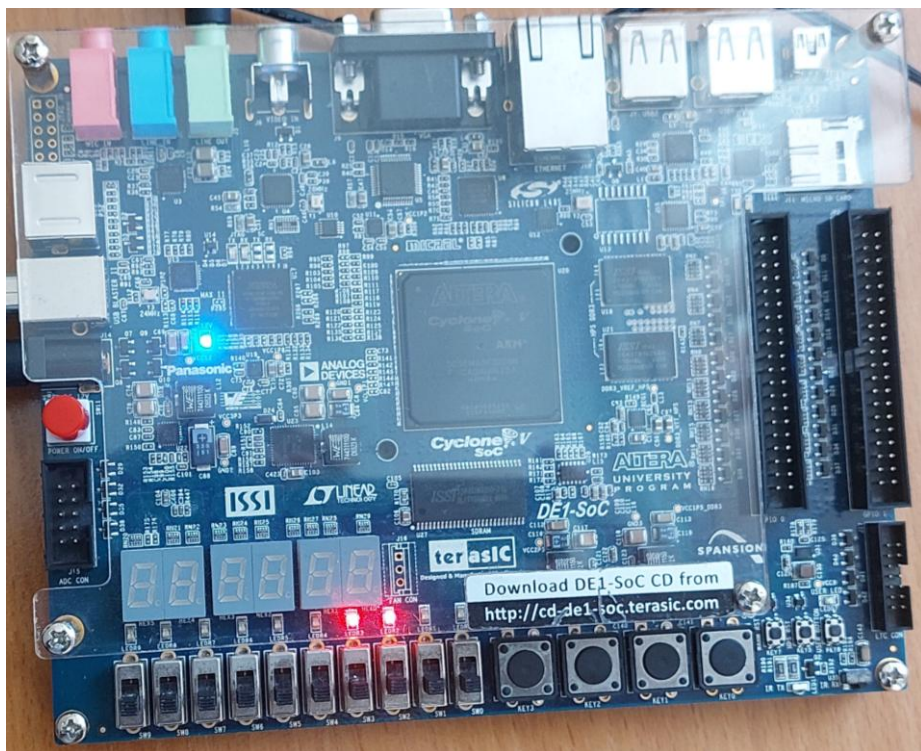


Рисунок 3.14 - Медіаплеєр запущений звук номер 4

Зм.	Арк.	№ докум.	Підпис	Дата

На рисунку 3.15 показані колонки які виводять звук.



Рисунок 3.15 - Колонки для виведення звуку

Виконано завершальний етап реалізації програмно технічного засобу медіаплеєра на FPGA що включає налаштування виводів компіляцію проєкту та його запуск на реальній апаратній платформі. Отримані результати підтверджують працездатність розробленої системи та відповідність її функціональних характеристик поставленим вимогам. Це свідчить про успішну реалізацію медіаплеєра із використанням програмованої логічної інтегральної схеми та можливість подальшого розвитку даного проєкту.

На рисунках 3.16 і 3.17 показано схему електричну і принципову проєкта.

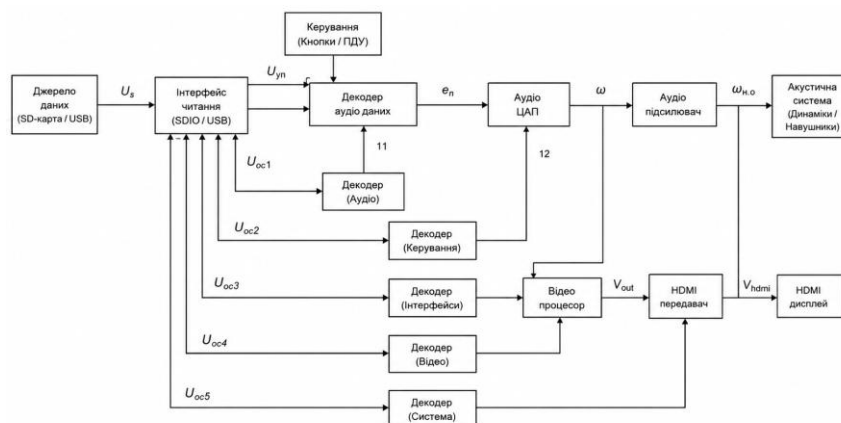


Рисунок 3.16 - Схема електрична проєкту

Зм.	Арк.	№ докум.	Підпис	Дата

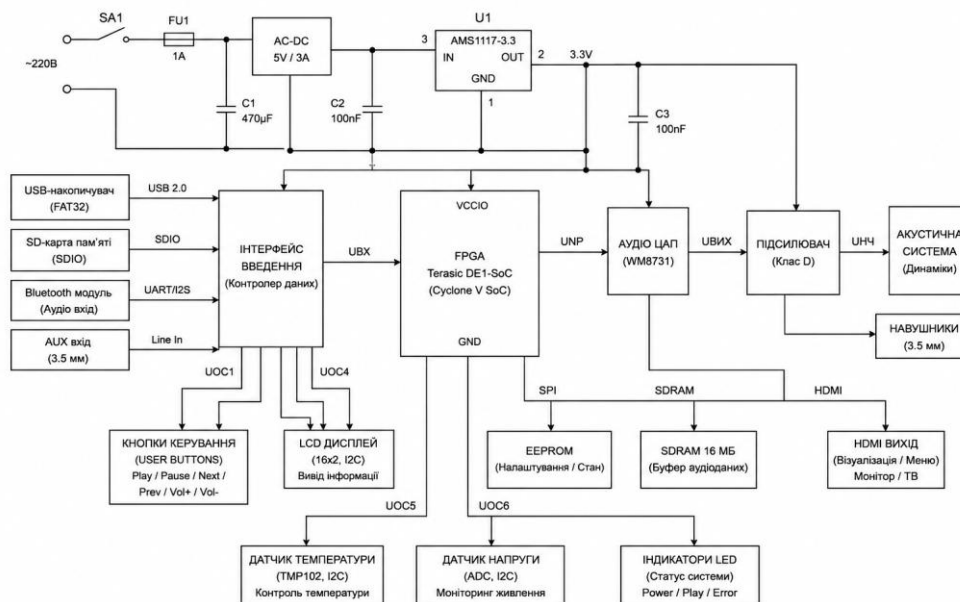


Рисунок 3.17 - Схема принципова проекту

3.9 Висновки до третього розділу

У третьому розділі було виконано повний цикл реалізації програмно технічного засобу медіаплеєра на FPGA починаючи зі створення проекту у середовищі розроблення та завершуючи запуском системи на апаратній платформі. Послідовно розроблено всі функціональні модулі що забезпечують обробку керуючих сигналів генерацію аудіосигналу формування та передавання цифрового аудіопотоку а також ініціалізацію аудіокодека. Кожен із модулів реалізує окрему частину загальної логіки що дозволило побудувати структуровану та зрозумілу архітектуру системи. Особливу увагу приділено інтеграції компонентів верхнього модуля де організовано взаємодію між усіма блоками та забезпечено коректне функціонування медіаплеєра. Реалізовано механізм керування за допомогою кнопок що дозволяє виконувати запуск зупинку та перемикання звукових сигналів. Генерація аудіоданих здійснюється засобами цифрової логіки що підтверджує можливість створення аудіосистем без використання зовнішніх джерел сигналу. Завершальним етапом стало налаштування виводів та завантаження конфігурації у FPGA.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень було розроблено програмно технічний засіб реалізації медіаплеєра на FPGA із використанням апаратної платформи на базі програмованої логічної інтегральної схеми. Отримано цілісну систему що забезпечує генерацію оброблення та відтворення аудіосигналів у реальному часі із можливістю керування режимами роботи.

У першому розділі проведено аналіз основних понять пов'язаних із медіаплеєрами та технологією FPGA розглянуто принципи їх функціонування області застосування а також особливості побудови подібних систем. Описано базові підходи до реалізації цифрових аудіопристроїв та визначено ключові характеристики що впливають на їх ефективність.

У другому розділі виконано проектування структури медіаплеєра на базі плати Terasic DE1 SoC розглянуто апаратні можливості платформи її інтерфейси та ресурси що використовуються у процесі реалізації. Описано принцип роботи аудіопідсистеми організацію передавання даних а також особливості конфігурації та програмування FPGA. Проведено аналіз переваг та обмежень використання даної технології для створення мультимедійних систем.

У третьому розділі реалізовано власний проєкт медіаплеєра що включає створення проєкту у середовищі розроблення розроблення окремих функціональних модулів та їх інтеграцію у єдину систему. Виконано налаштування виводів та завантаження конфігурації у FPGA після чого проведено перевірку роботи системи на реальній апаратній платформі. Отримані результати підтверджують працездатність розробленого рішення та правильність обраних підходів до його реалізації. У підсумку розроблений програмно технічний засіб демонструє ефективність використання FPGA для створення цифрових аудіосистем та може бути використаний як основа для подальшого розширення функціональних можливостей медіаплеєра.

					КВРКІ.22045.22.03.15 ПЗ	Арк. 67
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. AbuShanab S. FPGA-based implementation and comparative analysis of Robinson and Kirsch compass edge detectors. *Journal of VLSI Circuits and Systems*. 2025. Vol. 7(2). P. 99-110.
2. Altman M. B., Wan W., Hosseini A. S., Nowdeh S. A., Alizadeh M. Machine learning algorithms for FPGA implementation in biomedical engineering applications: a review. *Heliyon*. 2024. Vol. 10(4).
3. Amonoy J. M. C., Caya M. V. C. Gesture-controlled device for speakers using machine learning. *2025 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAJET)*. 2025. P. 90-95.
4. Arkannezhad F., Sehatbakhsh N. Attest like software: formally verified software programmable proof of execution architecture using SoC FPGAs. *Proceedings of the 2026 ACM SIGDA International Symposium on Field Programmable Gate Arrays*. 2026. P. 34-34.
5. Azzouzi O., Anane M., Ghanem M. C., Himeur Y., Wojtczak D. *Flexible and area-efficient codesign implementation of AES on FPGA. Cryptography*. 2025. Vol. 9(4). P. 78.
6. Bircanin F., Nevsky A., Cruice M. N., Markovic O., Neate T. A sound understanding an in situ deployment of an accessible audio media player with people living with aphasia. *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*. 2026. P. 1-20.
7. Cope B. та ін. Open standard software defined AV over IP framework. *SID Symposium Digest*. 2023. Vol. 54(1). P. 425-428.
8. Desoky M., Elmezayen M. R., Naguib A. Design and implementation of a digital phase frequency detector for ADPLL. *2025 IEEE International Microwave and Antenna Symposium (IMAS)*. 2025. P. 1-5.
9. . Diachok R., Lysiak V., Klym H. FPGA-based mathematical special computer using NIOS2 core. *Electronics and Information Technologies*. 2024. № 26.

					КВПКІ.22045.22.03.15 ПЗ	Арк. 68
Зм.	Арк.	№ докум.	Підпис	Дата		

10. dos Santos Cardoso D., Júnior H. P. L. Comunicação entre processador HPS e FPGA em um chip SoC Cyclone V. *Notas Técnicas*. 2025. Vol. 15(3).
11. Erdem O., Soylu T., Carus A. Pipelined decision trees for traffic classification on FPGAs. *The Computer Journal*. 2024. Vol. 67(3). P. 825-839.
12. Farkas C. та ін. Sustainable PCB substrates based on PLA flax composites. *Sustainable Materials and Technologies*. 2024. Vol. 40. e00902.
13. Firmansyah I., Setiadi B., Kurniawan D., Indrawijaya R., Subekti A., Rohman B. P. A. U. General Matrix Multiplication GEMM evaluation on Cyclone V SoC FPGA using OpenCL. *2023 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications*. 2023. P. 12-17.
14. Ibrahim S. H., Hameed I. S., Ali H. H. High performance IIR filter design based on fast multiplier. *Diyala Journal of Engineering Sciences*. 2025. P. 192-202.
15. Jamil S., Kwon O. J., Lee J., Ullah F., Yaseen Afnan. A novel multimedia player for international standard JPEG Snack. *Journal of Imaging*. 2023. Vol. 9(3). P. 58.
16. Karthikhaa Shree K. T., Lakshna Shree R., Lakshmi S. A vision based hand gesture interface for controlling VLC media player. *International Conference on Computational Intelligence in Data Science*. 2025. P. 258-269.
17. Kavitha S., Swetha M., Thanishka K. An efficient low power design of multiplierless discrete cosine transform. *IEEE ICBSII*. 2026. P. 1-6.
18. Khong T. T. T., Tran V. L., Phan H. P., Duong D. H. Crowd counting using deep learning model on FPGA card. *Hue University Journal of Science*. 2024. Vol. 133(2B). P. 37-48.
19. Khudhur Z. M., Abdulsattar A., Mahmood T. A., Dhulkefl E. J. Control of a 3-axis mechanical system using FPGA programming techniques. *Journal of Research in Engineering and Computer Sciences*. 2025. Vol. 3(03). P. 48-57.
20. Kim T. H., Hong W. C., Pyo S. H., An B. H., Park T. S. Study on HILS implementation of FPGA-based PFC circuits using sub-cycle average models. *Energies*. 2025. Vol. 18(24). P. 6443.

21. Kwon W., Cheon S. Real-time holography processor architecture with HBM. *SID Symposium Digest*. 2024. Vol. 55(1). P. 1563-1566.
22. Lazaro J. B. та ін. FPGA-based pulse rate detection system using DE0-Nano. *2025 IEEE ICBET*. 2025. P. 45-49.
23. Léonard C., Stober D., Schulz M. FPGA-enabled machine learning applications in earth observation: a systematic review. *ACM Computing Surveys*. 2026.
24. Li Z. та ін. CPLoRa: parallel LoRa backscatter communications. *IEEE VTC 2025*. 2025. P. 1-5.
25. Loh S. H., Tan G. Y., Sim J. J. VLSI design course with FPGA implementation of pipelined RISC-V processor. *ICCSCE 2025*. 2025. P. 36-41.
26. Maklad Y. та ін. Reconfigurable duplication for multi-die FPGA space applications. *STA 2025*. 2025. P. 200-204.
27. Melo W. та ін. A high-resolution 64-multi-phased time-to-digital converter architecture implemented on Cyclone V FPGA. 2025.
28. Minsky H., Khanna O., Fang S., Zhu A. Real-time keyword spotting: FPGA-based inference accelerator. 2026.
29. Nagamine N., Watanabe M., Watanabe N. Radiation-hardened JTAG interface for ORGA VLSIs. *IEEE ICCE 2026*. 2026. P. 1-6.
30. Pantula M. A study on media players from accessibility perspective. *International Journal of Computer Aided Engineering and Technology*. 2023. Vol. 18(4). P. 263-289.
31. Pinjare S. L., Rajeev B. R., Awasthi K., Vikas M. B. Implementation of STFT for auditory compensation on FPGA. *ERCICA 2022*. 2022. P. 483-497.
32. Poonguzhali S. D., Prabakar T. N., Elamaran V. FPGA-based cascaded digital filters for stethoscopes. *Circuits, Systems and Signal Processing*. 2025. P. 1-30.
33. Rajendra Prasad K. та ін. Application controlling using hand gestures through YOLOv5s. 2023. P. 857-864.

					КВПКІ.22045.22.03.15 ПЗ	Арк. 70
Зм.	Арк.	№ докум.	Підпис	Дата		

34. Saini S. S., Sharma L. S. Investigation of HTTP Live Streaming protocol adaptability. *Journal of The Institution of Engineers India*. 2025. Vol. 106(4). P. 1081-1089.
35. Sanmathi T. та ін. Optimized video streaming with FFmpeg and MediaCodec. *CRC Press*. 2025. P. 407-420.
36. Sekioka U. та ін. Radiation-hardened TMR serial communication system. *IEEE ICCE 2025*. 2025. P. 1-4.
37. Sharihan N. M. та ін. Bitstream-level IO tampering in FPGA systems. *TENCON 2025*. 2025. P. 454-458.
38. Siriwardana S. M. D. N. Using media player software as stealth attack vectors. *IEEE GlobConET*. 2023. P. 1-13.
39. Sola-Thomas E., Sarker M. A. B., Imtiaz M. FPGA-controlled AI vision for prosthetics hand. *IEEE AIIoT 2023*. 2023. P. 520-524.
40. Strogonov A. V., Vinokurov A., Arsentiev A. Development of a multi-cycle RISC-V microprocessor core on Cyclone V FPGA. *Elektronika*. 2025. № 1. P. 96-100.
41. Tam S. Y. та ін. Processing-in-memory architectures for MAC operations in FPGA. *IEEE ECE 2025*. 2025. P. 36-40.
42. Tasci M. та ін. FPGA-QNN: quantized neural network acceleration. *Applied Sciences*. 2025. Vol. 15(2). P. 688.
43. Tettey D. K., Elamassie M., Uysal M. Dual-channel VLC system for V2V video streaming. *ICSPIS 2024*. 2024. P. 1-6.
44. Vaithianathan M., Patil M., Ng S. F., Udkar S. Low-power FPGA design techniques for mobile devices. *ESP Journal*. 2024. Vol. 2(2). P. 82-93.
45. Villamizar J., Borra V., Li F. FPGA and Nios processor for induction heating controller. *Discover Electronics*. 2024. Vol. 1(1). P. 27.
46. Xu F. та ін. A custom parallel hardware architecture on FPGA. *IEEE Transactions on Industrial Electronics*. 2021. Vol. 69(11). P. 11569-11579.
47. Sanju B. N., Sandeep S. V., Manjunatha G., Manojkumar S. B. *Energy-efficient low-dropout regulator architecture on Altera Cyclone II FPGA for ultralow*

power SoC applications. 2025 Third International Conference on Networks, Multimedia and Information Technology (NMITCON). 2025. P. 1-6.

48. Yang Q. Implementierung einer MPEG Codierung auf FPGAs. *Master's thesis*. 2024.

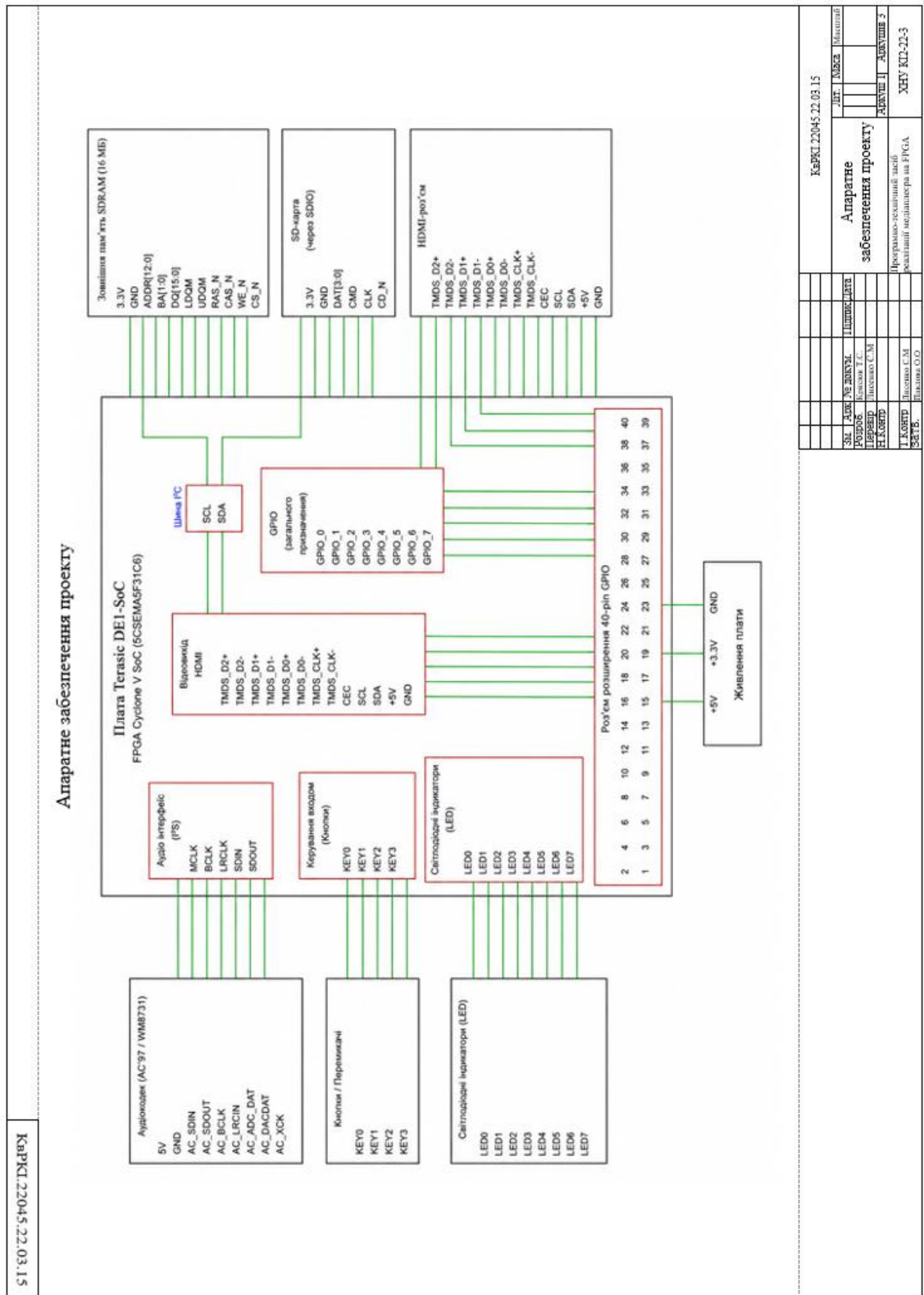
49. Zakharov V. M., Shalagin S. V. Random variable generator based on a d-ary tree. *Kazan University Journal*. 2025. № 508.

50. Хижняк А., Лисенко О. Розроблення архітектури багатоканального конвеєрного радіоприймача на основі FPGA. *Herald of Khmelnytskyi National University*. 2025. № 355(4). С. 686-694.

					КВРКІ.22045.22.03.15 ПЗ	Арк. 72
Зм.	Арк.	№ докум.	Підпис	Дата		

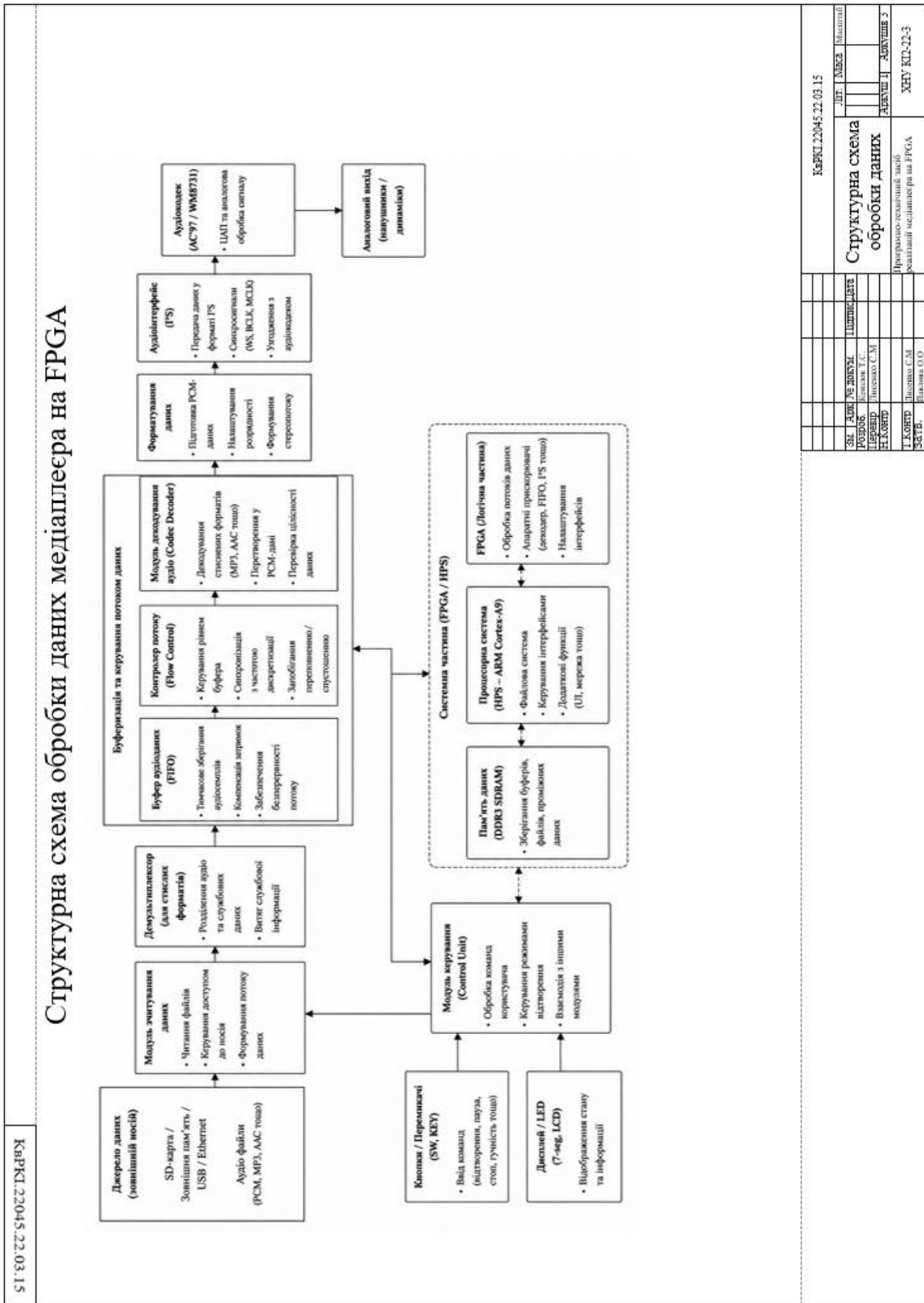
ДОДАТОК Б (обов'язковий)

Копія креслення «Апаратне забезпечення проекту»



ДОДАТОК В (обов'язковий)

Копія креслення «Структурна схема обробки даних»



ДОДАТОК Г
(обов'язковий)

Копія креслення «Лістинг коду програми»

Файл media_player_top.v

```
module media_player_top(
```

```
    input CLOCK_50,  
    input [3:0] KEY,
```

```
    output [7:0] LED,
```

```
    output AUD_XCK,  
    output AUD_BCLK,  
    output AUD_DACLK,  
    output AUD_DACDAT,
```

```
    output AUD_I2C_SCLK,  
    inout  AUD_I2C_SDAT
```

```
);
```

```
wire start;
```

```
wire pause;
```

```
wire next;
```

```
wire back;
```

```
wire play_enable;
```

```
wire [2:0] tone_select;
```

```
wire [15:0] audio_sample;
```

```
wire [15:0] sample_out;
```

```
wire bclk;
```

```

wire lrclk;
wire audio_data;

wire i2c_clk;
wire i2c_data;

// MASTER CLOCK
assign AUD_XCK = CLOCK_50;

// BUTTONS
button_controller buttons(

    .clk(CLOCK_50),
    .KEY(KEY),

    .start(start),
    .pause(pause),
    .next(next),
    .back(back)

);

// FSM
player_fsm player(

    .clk(CLOCK_50),

    .start(start),
    .pause(pause),
    .next(next),
    .back(back),

```

```

        .play_enable(play_enable),
        .tone_select(tone_select)

);

// SOUND GENERATOR
audio_sample_generator tone(

    .clk(CLOCK_50),
    .enable(1'b1),
    .tone_select(tone_select),

    .sample(audio_sample)

);

// PAUSE = TИЩА
assign sample_out = play_enable ? audio_sample : 16'd32768;

// I2S
i2s_driver audio(

    .clk(CLOCK_50),
    .sample(sample_out),

    .bclk(bclk),
    .lrclk(lrclk),
    .data(audio_data)

);

```

```

// CODEC INIT
wm8731_init codec(

    .clk(CLOCK_50),

    .i2c_sclk(i2c_clk),
    .i2c_sdat(i2c_data)

);

// LED
assign LED[2:0] = tone_select;
assign LED[3]   = play_enable;
assign LED[7:4] = 4'b0000;

// AUDIO
assign AUD_BCLK      = bclk;
assign AUD_DACLK     = lrclk;
assign AUD_DACDAT    = audio_data;

assign AUD_I2C_SCLK = i2c_clk;
assign AUD_I2C_SDAT = i2c_data;

endmodule

Файл audio_sample_generator.v
module audio_sample_generator(

    input clk,

```

```

    input enable,
    input [2:0] tone_select,

    output reg [15:0] sample = 0

);

reg [15:0] counter = 0;
reg [15:0] limit;

always @(*)
begin

    case(tone_select)

        3'b001: limit = 12500; // 2000 Hz
        3'b010: limit = 6250;  // 4000 Hz
        3'b011: limit = 4166;  // 6000 Hz
        3'b100: limit = 3125;   // 8000 Hz
        3'b101: limit = 2500;   // 10000 Hz
        3'b110: limit = 2083;   // 12000 Hz
        3'b111: limit = 1785;   // 14000 Hz

        default: limit = 12500;

    endcase

end

always @(posedge clk)
begin

```

```

    if(enable)
    begin
        if(counter >= limit)
        begin
            counter <= 0;
            sample <= ~sample;
        end
        else
            counter <= counter + 1;
        end
    end

end

endmodule

```

Файл button_controller.v

```

module button_controller(

    input clk,
    input [3:0] KEY,

    output reg start = 0,
    output reg pause = 0,
    output reg next = 0,
    output reg back = 0

);

reg [3:0] prev = 4'b1111;

always @(posedge clk)
begin

```

```

start <= 0;
pause <= 0;
next <= 0;
back <= 0;

if(KEY[0] == 0 && prev[0] == 1)
    start <= 1;

if(KEY[1] == 0 && prev[1] == 1)
    pause <= 1;

if(KEY[2] == 0 && prev[2] == 1)
    next <= 1;

if(KEY[3] == 0 && prev[3] == 1)
    back <= 1;

prev <= KEY;

end

endmodule

```

Файл i2s_driver.v

```

module i2s_driver(

    input clk,
    input [15:0] sample,

    output reg bclk = 0,
    output reg lrclk = 0,
    output reg data = 0

```

```

);

reg [4:0] bit_counter = 0;

always @(posedge clk)
begin

    bclk <= ~bclk;

    if(bclk)
    begin

        data <= sample[15 - bit_counter];

        bit_counter <= bit_counter + 1;

        if(bit_counter == 15)
        begin
            bit_counter <= 0;
            lrclk <= ~lrclk;
        end

    end

end

end

endmodule

```

Файл player_fsm.v

```

module player_fsm(

```

```

    input clk,

```

```

input start,
input pause,
input next,
input back,

output reg play_enable = 0,
output reg [2:0] tone_select = 3'b001

);

always @(posedge clk)
begin

    if(start)
        play_enable <= 1;

    if(pause)
        play_enable <= 0;

    if(next)
    begin
        if(tone_select == 3'b111)
            tone_select <= 3'b001;
        else
            tone_select <= tone_select + 1;
    end

    if(back)
    begin
        if(tone_select == 3'b001)
            tone_select <= 3'b111;
    end

```

```

        else
            tone_select <= tone_select - 1;
        end

    end

end

endmodule

```

Файл wm8731_init.v

```

module wm8731_init(

    input clk,

    output reg i2c_sclk = 1,
    inout i2c_sdat

);

reg [15:0] clk_div = 0;
reg i2c_clk = 0;

reg [7:0] state = 0;
reg [5:0] bit_cnt = 0;

reg sdat_reg = 1;
assign i2c_sdat = sdat_reg;

always @(posedge clk)
begin

    clk_div <= clk_div + 1;

```

```

    if(clk_div == 250)
    begin
        clk_div <= 0;
        i2c_clk <= ~i2c_clk;
    end

end

always @(posedge i2c_clk)
begin

    case(state)

    0: begin
        sdat_reg <= 0;
        state <= 1;
    end

    1: begin
        sdat_reg <= 1;
        state <= 2;
    end

    // RESET

    2: begin
        send_word(16'h0F00);
        state <= 3;
    end

end

```

```
// POWER UP

3: begin
    send_word(16'h0C00);
    state <= 4;
end

// DIGITAL AUDIO FORMAT

4: begin
    send_word(16'h0702);
    state <= 5;
end

// SAMPLING CONTROL

5: begin
    send_word(16'h0812);
    state <= 6;
end

// ACTIVATE

6: begin
    send_word(16'h0901);
    state <= 7;
end

default:
    state <= 7;
```

```
        endcase

    end

    task send_word;

    input [15:0] data;

    integer i;

    begin

        for(i = 15; i >= 0; i = i - 1)
            begin

                sdat_reg <= data[i];
                i2c_sclk <= 0;
                i2c_sclk <= 1;

            end

        end

    end

endtask

endmodule
```

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Тимур КРИСЮК

Співавтор:

Назва: Програмно-технічний засіб реалізації медіаплесра на FPGA

Експерт: Сергій ЛИСЕНКО

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 1.6%

Коефіцієнт подібності 2: 0.89%

Мікропробіли: 3

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2026-05-20 18:19:20.0

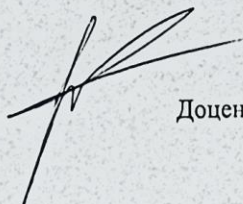
Після аналізу Звіту подібності констатую наступне:

- Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.
- Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.
- Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2026-05-20

Дата



Доцент Андрій Нічепорук

експерт

Anti-Plagiarism (<http://ap.km.ua>) v-15.701

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 14%

ID: 271823 Назва: БКР Програмно-технічний засіб реалізації медіаллеера на FPGA Додано в БД: 2026-05-20 Автора: Тимур КРИСЮК Керівники: Сергій ЛИСЕНКО Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	101184	733	868 (1%)	12 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Крисюк Тимур Сергійович

Тема: Програмно-технічний засіб реалізації медіаплеєра на FPGA

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 64

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи була розробка програмно-технічного засобу реалізації медіаплеєра на FPGA
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі було проведено аналіз основних принципів побудови програмно технічних засобів реалізації медіаплеєра на базі програмованих логічних інтегральних схем FPGA та розглянуто особливості їх застосування у складі сучасних цифрових мультимедійних систем. У другому розділі було детально розглянуто процес проектування програмно технічного засобу реалізації медіаплеєра на FPGA із використанням апаратної платформи Terasic DE1 SoC що базується на програмованій логічній інтегральній схемі Intel Cyclone V. Проведений аналіз дозволив визначити основні характеристики обраної платформи її архітектурні особливості та можливості застосування для побудови цифрової системи відтворення аудіоінформації у режимі реального часу. У третьому розділі було виконано повний цикл реалізації програмно технічного засобу медіаплеєра на FPGA починаючи зі створення проєкту у середовищі розроблення та завершуючи запуском системи на апаратній платформі. Послідовно розроблено всі функціональні модулі що забезпечують обробку керуючих сигналів генерацію аудіосигналу формування та передавання цифрового аудіопотоку а також ініціалізацію аудіокодека. Кожен із модулів реалізує окрему

частину загальної логіки що дозволило побудувати структуровану та зрозумілу архітектуру системи.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага аналізу предметної області; недостатньо чітко описано процес складання програмно-технічного засобу.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: робота виконана на достатньому технічному рівні.


8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільно (D / 70)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

*Мартишюк Валерій Володимирович, р.т.ч.,
проф., професор кафедри ІНІТ та РІС*

"01" червня 2026 р.

 (підпис)

Зав. кафедри КІС
д-р. філософії Ользі ПАВЛЮВІЙ

Тимур КРИСЮК

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-3

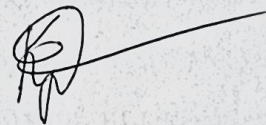
ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Програмно-технічний засіб реалізації медіаплеєра на FPGA
 Автор Тимур КРИСЮК
 Освітня програма Комп'ютерна інженерія та програмування
 Рівень вищої освіти перший (бакалаврський)
 Спеціальність 123 Комп'ютерна інженерія
 Науковий керівник: д.т.н., проф Сергій ЛИСЕНКО

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.
- 4) значна частина знайденого плагіату відноситься до списку використаних джерел


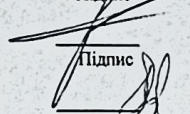

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 1.6%; та системою Anti-Plagiarism складає 1.0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

01.06.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи


 Підпис

 Підпис

 Підпис

Ольга ПАВЛОВА
Ім'я, ПРІЗВИЩЕ

Андрій НІЧЕПОРУК
Ім'я, ПРІЗВИЩЕ

Сергій ЛИСЕНКО
Ім'я, ПРІЗВИЩЕ