

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмно-технічний засіб відео-декодера на базі FPGA Altera Cyclone V
Назва теми

КвРКІ.190188.14.14.ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва


Освітня програма «Комп'ютерна інженерія»
Назва

Виконав: студент IV курсу, група КІ2с-19-1


Підпис


О.О.Кухар
Ініціали, прізвище

Керівник


Підпис, дата

С.М. Лисенко
Ініціали, прізвище

Нормоконтролер


Підпис, дата

С.М. Лисенко
Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інформаційних
систем


Підпис

Т.О. Говоруценко
Ініціали, прізвище

« 1 » червня 2022 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 11 ” 01 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Кухару Олександрю Олеговичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб відео-декодера на базі FPGA Altera Cyclone V

Керівник проекту (роботи) Лисенко С. М., д.т.н., проф.

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

Затверджена наказом ректора університету від 06.01.2022 р. № 1

2. Строк подання студентом проекту (роботи) на кафедру 07.06.2022 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

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

Дослідження предметної області та постановка задачі

проективання програмно-технічного засобу

програмно-апаратна реалізація та тестування програмно-технічного засобу



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

Конфігурація FPGA

Інтерфейс проекту

Вікна компіляції, налагодження програми, а також перегляд вмісту регістрів

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КНП		
Антиплагіат	Нічепорук А.О., доцент кафедри КНП		

7. Дата видачі завдання « 06 » 09 2021 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	11.01.2022	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2022	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2022	виконано
4	Робота над розділом 2 - проектування програмно-технічного засобу	01.04.2022	виконано
5	Робота над розділом 3 програмно-апаратна реалізація та тестування програмно-технічного засобу	30.04.2022	виконано
6	Оформлення пояснювальної записки згідно вимог	31.05.2022	виконано
7	Попередній захист ВКР	02.06.2022	виконано
8	Захист ВКР на засіданні ЕК	Червень 2022 року	

Студент

Керівник проекту (роботи)
Засорнов


Підпис

О.О. Кухар
Ініціали, прізвище

С.М. Лисенко

Підпис

Ініціали, прізвище

№	Ф о р м а т	Позначення	Найменування	К і л ь о с т і в	№ ек з	П р и м і т к а
			Текстові документи			
		КВРКІ 190188. 14. 14 ПЗ	Пояснювальна записка	63		
			Графічні матеріали			
		КВРКІ 190188. 14. 14 E8	Конфігурація FPGA	1		
		КВРКІ 190188. 14. 14 E8	Інтерфейс проекту	1		
		КВРКІ 190188. 14. 14 E8	Вікна компіляції	1		
			налагодження програми, перегляд вмісту регістрів			

КВРКІ 190188. 14. 14 ВП

м	Ар к	№ докум	Підпис	Дата
Зробив		Кухар О.О.		
Перевір.		Лисенко С.М.		
І. контр.		Лисенко С.М.		
Загв.		Говорущенко		

Відомість проекту

Літера	Аркуш	Аркушів
У	1	1

ХНУ, КІ2с-19-1

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб відео-декодера на базі FPGA Altera Cyclone V».

Автор роботи: Кухар Олександр Олегович.

Керівник роботи: Лисенко Сергій Миколайович.

Пояснювальна записка: 65 с., 12 рис., 5 табл., 4 дод., 15 джерел.

Графічна частина: 15 презентаційних слайдів.

FPGA, ВІДЕОКОДУВАННЯ, ALTERA, QUARTUS, КОМП'ЮТЕРНА СИСТЕМА.

Метою роботи є розробка програмно-технічного засобу керування конфігурацією та інфраструктурою локального центру обробки даних.

У цій роботі розроблений засіб відео-декодера на базі FPGA Altera Cyclone V.

Підпис студента







Дата

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	3
ВСТУП	4
1.1 Поняття FPGA.....	5
1.2 Переваги та недоліки ASIC та FPGA	7
1.3 Еволюція FPGA	8
1.4 Висновок	10
2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ	12
2.1 Плата разработчика DE1-SoC	12
2.2 Засоби розробки.....	18
2.3 Linux BSP (пакет підтримки плат).....	19
2.4 Розробка програмно-апаратних засобів засобами Quartus	20
2.5 Редактор плат.....	32
2.6 Редактор компонентів	33
2.7 Взаємодія ECAD-MCAD	34
2.8 Висновки.....	39
3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ ВІДЕО-ДЕКОДЕРА НА БАЗІ FPGA ALTERA CYCLONE V	40
3.1 Створення прошивки FPGA	40
3.2 Створення Preloader	51
3.3 Розроблення програмного забезпечення.....	58
3.5 Висновки	64
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	67
Додаток А Лістинг розробленого програмного забезпечення	99
Додаток Б Інтерфейсні вікна програмного забезпечення.....	100

КвРКІ. 190188. 14. 14 ПЗ

Зм.	Арк.	Недокум.	Підпис	Дата				
Виконав		Кухар			Програмно-технічний засіб відео-декодера на базі FPGA Altera Cyclone V Пояснювальна записка	Літера	Аркуш	Аркушів
Перевір.		Лисенко С.М.				ХНУ, КІ2с-19-1		
Н.контр.		Аксенко С.М.						
Затвер.		Товстухин						

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ПЗ - програмне забезпечення

ОС - операційна система

ПКВМ (FPGA) – Програмована користувачем вентиляна матриця

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		3

ВСТУП

Важко уявити іншу технологію, яка настільки різнобічна як FPGA.

FPGA - Field-Programmable Gate Array, тобто програмована логічна матриця (ПЛМ), програмована логічна інтегральна схема (ПЛІС). Це технологія, за якої створюється мікросхема з набором логічних елементів, тригерів, іноді оперативної пам'яті та програмованих електричних зв'язків між ними. При цьому програмування FPGA виявляється схожим на розробку електричної схеми, а не програми. Користуюся цією технологією давно і спробую описати найкорисніші на мою думку застосування в міру їх ускладнення.

Частина мікросхем має інтерфейс 1.2В, інша 1.5, 1.8, 2.5, 3.3В, і всі ці мікросхеми можна підключити до однієї FPGA та забезпечити двосторонній обмін за рахунок того, що будь-яка FPGA має кілька банків введення-виводу, кожен з яких може мати своє опорне напруження сигналів. Наприклад, так:

FPGA досить дорогі, але надійні пристрої. Вони починають включатися при меншій нарузі, ніж номінальне, витримують імпульсні наведення, часто коротке замикання на ніжках ІО, швидко завантажуються і можуть використовуватися для контролю та управління процесорами та розумною периферією. Плюс можуть реалізовувати процесорів мають будь-якого виробника FPGA (від 8086 до ARM), дозволяють швидко створити процесор з певним набором периферії і вставити його в проект FPGA. До процесора додається компілятор та налагоджуваю та інтерфейсами до зовнішньої. Тобто проблема ефективної FPGA обмежена лише кількістю паралельних блоків та часом виконання однієї операції. Відлагодивши таким чином обчислення можна замовити ASIC, тобто замовну мікросхему, що виконує ті ж функції, але дешевші (при масовому виробництві) та з меншим енергоспоживанням. І ця ідея виявилася настільки перспективною та зручною, що вендори FPGA почали створювати спеціальне ПЗ, що дозволяє інтерактивно переносити частини обчислень із програми.

Таким чином, використання FPGA для вирішення прикладних задач є дуже актуальним.

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		4

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Поняття FPGA

FPGA - це мікросхема, яка запрограмована схемою. Кажуть, що він «імітує» цю схему. Ця емуляція працює повільніше, ніж реальна схема, якби вона була реалізована в ASIC — вона має нижчу тактову частоту і споживає більше енергії, але її можна перепрограмувати кожні кілька сотень мілісекунд.

У середині 1980-х Росс Фріман та його колеги купили технологію у Zilog і запустили Xilinx, орієнтуючись на ринки емуляції ASIC та освіти. (Zilog вийшов з Еххон, оскільки в 1970-х роках люди вже боялися, що нафта закінчиться через 30 років, що актуально і сьогодні). Паралельно, Altera була заснована з подібною технологією в своїй основі.

Люди, які виробляють ASIC, почали використовувати FPGA для емуляції своїх ASIC, перш ніж прикріпити їх до маски та відправити на фабрику для виробництва. Intel, AMD та багато інших компаній використовують FPGA для емуляції своїх чіпів перед їх виготовленням.

Телекомунікаційна галузь активно використовує FPGA. Стандарти телекомунікацій постійно змінюються, а створення телекомунікаційного обладнання є важким, тому компанія, яка постачає телекомунікаційні рішення, першою прагне захопити найбільшу частину ринку.

Оскільки створення ASIC займає багато часу, FPGA пропонують можливість швидкого шляху. FPGA почали використовуватися для перших версій телекомунікаційного обладнання, що поклало початок конфлікту цін на FPGA. Хоча ціна FPGA не має значення для ринку емуляції ASIC, ціна чіпа для телекомунікацій важлива.

AT&T і Lucent створили власні FPGA, які називаються ORCA (оптимізовані реконфігуровані клітинні масиви), але вони не були конкурентоспроможними з Xilinx або Altera з точки зору швидкості або розміру кремнію.

На початку FPGA використовувалися для SDR (програмно-визначених радіоприймачів), створюючи радіостанції для зв'язку за різними стандартами

					КвРКІ 190188. 14. 14 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		5

одночасно, по суті, маючи один телефон, який розмовляє багатьма мовами. Цього разу FPGA вразили величезний нерв.

Комерційні постачальники розробили економічно ефективні рішення, і сьогодні кожна базова станція на планеті має технологію SDR. У оборонній спільноті, з іншого боку, SDR були створені великими оборонними підрядниками з прибутковими застарілими продуктами для захисту. В результаті ціна радіопродуктів на основі FPGA була настільки високою, що частина оборонного ринку США отримала стійку алергічну реакцію на їх використання.

Далі FPGA спробували розвиватися на ринках DSP (процесор цифрових сигналів) і вбудованих. Почали з'являтися FPGA з маленькими жорсткими мікропроцесорами в кутку. Тиск щодо продажу цих нових FPGA був настільки високим, що якщо клієнти відмовлялися від нового сімейства мікросхем, їх заносили в чорний список, а іноді навіть відмовлялися від обслуговування на кілька місяців. Тиск для зростання ринку FPGA був і залишається величезним, як і масштаби невдач компаній FPGA у завоювання нових ринків, враховуючи неможливість зниження ціни на продукти FPGA через їх величезну площу поверхні та шари інтелектуальної власності.

Протягом останніх кількох років FPGA намагалися розвиватися на ринках НРС (високопродуктивних обчислень) і центрів обробки даних. У 2017 році Microsoft оголосила про використання FPGA Altera в центрі обробки даних, і Intel купила Altera. У 2018 році Xilinx оголосила про свою стратегію «Центр обробки даних перш за все», при цьому генеральний директор Xilinx заявив перед аудиторією аналітиків, що Xilinx більше не є компанією FPGA. Можливо, це була невелика драматизація, але історично є актуальність.

У НРС і центрах обробки даних FPGA головною перешкодою сьогодні є місце і маршрут — час, необхідний для запуску власного програмного забезпечення виробника FPGA, яке відображає схему на елементах FPGA. На великих FPGA і на швидкісному сервері ЦП місце і маршрут займають до трьох днів, і багато разів навіть через три дні програмне забезпечення не може знайти відображення.

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		6

У нафтогазовій сфері, однак, приблизно у 2007 році відкрилася ніша. Час, необхідний класичним комп'ютерам, щоб імітувати буріння отворів у землі, щоб знайти нафту, був довшим, ніж фактична споруда бурового майданчика та саме буріння. Використання прискорювачів FPGA різко змінило цей перевернутий час. Перші FPGA в центрі обробки даних нафтової компанії, що обчислюють сейсмічні зображення, були створені Maxeler Technologies.

Незважаючи на все це, FPGA залишаються швидким шляхом виходу на ринок, простим способом отримання конкурентних переваг і незамінною технологією для багатьох критично важливих ситуацій, незважаючи на те, що вони дорогі в порівнянні з ASIC. У HPC і центрах обробки даних, однак, FPGA мають значно нижчі експлуатаційні витрати в порівнянні з запуском програмного забезпечення на CPU або GPU. Необхідно менше FPGA, що вимагає набагато менше охолодження, ніж процесори і графічні процесори. FPGA створюють менші центри обробки даних, що вражає операторів, які бояться, що їхні центри обробки даних можуть зменшитися.

1.2 Переваги та недоліки ASIC та FPGA

Інший спосіб використання FPGA - це доповнення до ASIC. ASIC створені для забезпечення фіксованої функціональності з додаванням FPGA, щоб забезпечити деяку гнучкість для внесення змін в останню хвилину або адаптації продуктів до різних ринків.

Сучасні FPGA інтегрують все більше і більше жорстких функціональних можливостей і стають все більше схожими на ASIC, тоді як ASIC іноді додають трохи тканини FPGA у свій дизайн для налагодження, тестування, виправлення в польових умовах і гнучкості в додаванні невеликих частин функціональності за потреби. .

Одним із таких нових напрямів є автономна автомобільна промисловість. Оскільки алгоритми постійно змінюються, а закони можуть змінюватися, коли автомобілі знаходяться в полі, що вимагає оновлення драйверів, рішення має бути гнучким. FPGA мають нижчу тактову частоту, а отже, менші радіатори, що

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		7

призводить до меншого фізичного розміру, ніж процесори та графічні процесори. Нижче енергоспоживання та менший розмір роблять FPGA очевидним вибором. Тим не менш, графічні процесори легше програмувати і не вимагають триденного місця і маршруту.

Крім того, дуже важливо мати можливість запускати один і той же код в автомобілі та в хмарі (в першу чергу для моделювання та тестування), тому FPGA повинні бути доступні в хмарі, перш ніж їх можна буде використовувати в автомобілі. З цих причин багато розробників віддають перевагу графічним процесорам.

1.3 Еволюція FPGA

FPGA розвиваються. Сучасні інтерфейси намагаються зробити FPGA простішими для програмування, зробити їх більш модульними та більш кооперативними з іншими технологіями. FPGA підтримують шини AXI (Advanced Extensible Interface), що полегшує їх програмування, але також вносить величезну неефективність і робить FPGA менш продуктивними і, зрештою, набагато менш конкурентоспроможними. Академічні роботи, такі як стаття Еріка Чанга про динамічні мережі для FPGA, 1 допомагають вирішити проблему маршрутизації, але такі передові ідеї ще не були підхоплені промисловістю.

Для робочих навантажень НРС з великими потоками даних можна використовувати PCI Express і розгорнути методи приховування зв'язку. Але як щодо невеликих робочих навантажень, таких як NFV (віртуалізація мережевих функцій), які обслуговують велику кількість користувачів одночасно. Для NFV і прискорення віртуальних машин загалом, FPGA має підключатися безпосередньо до ЦП, можливо, використовуючи когерентність кешу як механізм зв'язку, як досліджується сьогодні VMware. Звичайно, ключовою особливістю є можливість аварійного завершення роботи FPGA без збою ЦП, і навпаки. Гіперскалярні технологічні компанії заново відкривають вимоги з часів мейнфреймів IBM, впроваджуючи все більшу складність у стандартизовані платформи.

					КвРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		8

Пропонуючи платформи FPGA, організації, які не мають бюджету на розробку ASIC і не знають останніх проблем і рішень виробництва кремнію, можуть розробляти схеми та створювати конкурентні переваги в своїх продуктах, наприклад, нові можливості для обчислень на краю Інтернету речей (Інтернет-речей) мережа, поблизу датчиків, дисплеїв або просто вбудовані в мережу, коли дані протікають.

Компанії FPGA висувують вертикально вгору по стеку і в сокет ЦП, де Intel домінує на ринку, включаючи, наприклад, спеціальні інструкції для NFV. Основними перешкодами для входу нових процесорів і FPGA в центр обробки даних є не тільки швидкість і вартість, а й наявність програмного забезпечення та драйверів для всіх можливих пристроїв введення-виводу.

Ключ до того, щоб FPGA працювали в центрі обробки даних, полягає в тому, щоб зробити їх простішими у використанні, наприклад, за допомогою автоматичних інструментів, які стимулюють використання FPGA без проблем з місцем і маршрутом. Корпорація Майкрософт піонером у використанні FPGA в гіперскалярному центрі обробки даних для прискорення алгоритмів Bing, NFV та AI. Microsoft також створила абстракції, мови для певної області та гнучкі апаратні інфраструктури. Комерційно головна проблема FPGA – це стратегія виходу на ринок.

Intel використовує FPGA для гнучкості. Це найуспішніша компанія, яка дотримується принципу створення апаратного забезпечення для роботи існуючого програмного забезпечення.

FPGA можуть бути швидшими, ніж процесори та графічні процесори, але важкий урок промисловості та інвестиційної спільноти полягає в тому, що більшість часу під час існування комп'ютера швидкість не має значення, а реальний час не має значення.

FPGA не мають стандартної моделі програмування з відкритим вихідним кодом і, отже, не мають стандартного ринку програм FPGA, які працюють на всіх мікросхемах FPGA або можуть бути легко перехресно скомпільовані.

Maxeler Technologies має рішення високого рівня для забезпечення такого інтерфейсу, але широке впровадження в галузі вимагає довіри. Щоб перейти від

					КвРКІ 190188. 14. 14 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		9

ранніх користувачів до користі для всіх, довіра вимагає узгодження та підтримки з боку відомих постачальників у центрі обробки даних.

Для обчислень у реальному часі існує виробництво для комп'ютерного зору на дронах це перевага FPGA у вазі та потужності. На супутнику оновлювати обладнання дуже дорого, тому FPGA забезпечують довгострокову гнучкість, яка може бути критичною. FPGA повинні знайти продукт, який резонує, і їх потрібно легко програмувати. Це не просто апаратне чи програмне забезпечення, це екосистема.

Одним із способів вийти за межі поточного ринку є компіляція в реальному часі та автоматичне генерування програм FPGA. Це легше сказати, ніж зробити, але можливості збільшуються, оскільки ШІ розриває простір додатків. Сьогодні все робиться за допомогою ШІ; навіть традиційні алгоритми, такі як сейсмографія для нафти і газу, включають ШІ. Для боротьби з блоками штучного інтелекту потрібне науково-технічне рішення. FPGA можуть бути гарною відправною точкою, можливо, спочатку для з'єднання блоків AI, а потім для включення їх у структуру FPGA, наприклад, чіпи наступного покоління від Xilinx — із структурою штучного інтелекту, процесорами, інтерфейсами 100G та елементами FPGA в одному 7-нм чіпі.

З іншого боку, з чіпами штучного інтелекту, які виробляють і споживають величезні обсяги даних, FPGA знадобляться, щоб годувати звіра та швидко переносити вихідні дані. З появою всіх нових ASIC для обробки штучного інтелекту, FPGA можуть забезпечити диференціацію компаній, що випускають мікросхеми AI.

1.4 Висновок

В розділі подано основні аспекти поняття FPGA.

Наведено основні переваги та недоліки застосування для вирішення задач.

Таким чином, зроблено висновок щодо можливості використання FPGA для розроблення програмно-апаратної реалізації відео-декодера.

					КвРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		10

В якості засобу створення пропонується залучення FPGA ALTERA CYCLONE V.

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		11

2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

2.1 Плата разработчика DE1-SoC

З метою виконання поставленого завдання розглянемо програмно-апаратні засоби, що використовуються для цього.

Плата разработчика DE1-SoC подана на рисунку 2.1.



Рисунок 2.1 - Плата разработчика DE1-SoC

Ресурсів плати ПЛІС DE0-nano не вистачає по ресурсах, а тому для вирішення задачі було обрано плату DE1-SoC від Terasic Technologies.

Крім того, що це більш нове покоління пліс - Cyclone V.

Між поколіннями не велика різниця між CycloneIV на DE0-nano, і Cyclone V, але потужніша, ніж Cyclone II.

Чіп має важливу перевагу - вбудований ARM процесор (Hardware Processor System у термінології Altera або HPS).

Що власне і відображено у назві плати SoC-system on chip.

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		12

Звичайно, у багатьох випадках можна обійтися FSM або тим самим NIOS II, але повноцінний dual core ARM із запущеним на ньому Linux звучить солідно.

Наявність процесора також є не маловажним фактором. Збільшено часу компіляції.

2.1.1 Комплект поставки

Комплект поставки подано на рисунку 2.2.



Рисунок 2.2 – Комплект поставки

Плата знаходиться досить великій транспортній коробці, розділеній на два відділення.

У першому знаходиться сама плата добре захищена від пошкоджень паралонової форми.

А під нею знаходяться: кабель USB (A to B), USB-кабель (A to Mini-Usb), блок живлення 12V 3.5A.

Штекер під американські розетки, так що знадобиться перехідник), Quick Start Guide і не багато рекламної інформаційної документації.

Дисків з ПЗ також немає, тому доведеться скачати з офіційного сайту.

Також комплект не містить це кабелю Ethernet та microSD картки

У великому відділенні присутня демо-плата DC934A - ЦАП і АЦП від Linear Technology.

2.1.2 Технічні характеристики

Технічні характеристики FPGA подано нижче:

- 1) пристрій Cyclone V SoC 5CSEMA5F31C6;
- 2) 85 000 програмованих логічних елементів;
- 3) 4450 Кбіт вбудованої пам'яті;
- 4) 6 дробових PLL;
- 5) 2 жорстких контролера пам'яті;

В нових чіпах Altera перейшли з 4-входових LE об'єднаних в блоки LAB, до 6-входових LE (об'єднаних в так звані об'єднані ALM), яка є суміщенням 2 4-входових LE.

У процесі розробки Quartus дозволяє будувати високопродуктивні схеми, де багато мультиплексорів тощо.

FPGA має двоядерний ARM Cortex-A9 (HPS).

Опис процесора займає 500 сторінок, так що до картинки можу додати хіба що те, що це повноцінний ARM процесор, і крім 2 ядер і розвиненої периферії має матрицю комутації.

Так що виходи периферійних пристроїв з деякими обмеженнями можна підключити як до різних висновків HPS, так і до логіки FPGA.

Для роботи з FPGA процесори HPS також мають спеціально виділені мости:

1. HPS-FPGA: конфігурований інтерфейс AMBA AXI 32, 64 або 128 біт.
2. FPGA-HPS: конфігурований інтерфейс AMBA AXI 32, 64 або 128 біт .

Вони використовуються для високошвидкісних програм (наприклад, передачі даних з пам'яті HPS в FPGA за допомогою DMA).

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		14

Зручною особливістю моста FPGA-to-HPS є те, що пристрої з FPGA можуть звертатися безпосередньо до периферії HPS.

Для управління більш простими і низькошвидкісними пристроями призначена шина Lightweight HPS-to-FPGA має 32 розряди і зручна для управління периферією FPGA (наприклад налаштування необхідного для вирішення конкретної задачі IP блоку).

HPS також має засоби для програмування FPGA, так що можна мати не одну, а кілька прошивок і змінювати їх при необхідності без зовнішніх програматорів.

2.1.3 Конфігурація

Конфігурація FPGA містить:

1. Пристрій Quad Serial Configuration – EPCQ256 на FPGA.
2. Вбудований USB Blaster II (звичайний USB-роз'єм типу B) .
3. Пам'ять 64 МБ (32Мx16) SDRAM на FPGA .
4. 1 ГБ (2x256Мx16) DDR3 SDRAM на HPS .
5. Гніздо для карти Micro SD на HPS.
6. Периферія.

Конфігурація FPGA подано на рисунку 2.2.

Вадливо відзначити, що плати від Terasic мають дуже розвинену практично периферію.

Але все ж таки, може коли вони прийдуть до модульності.

Наприклад, комусь потрібно більше I/O, а комусь навпаки активний акселерометр.

В такому разі можливим є адаптивна можливість до масштабованих рішень.

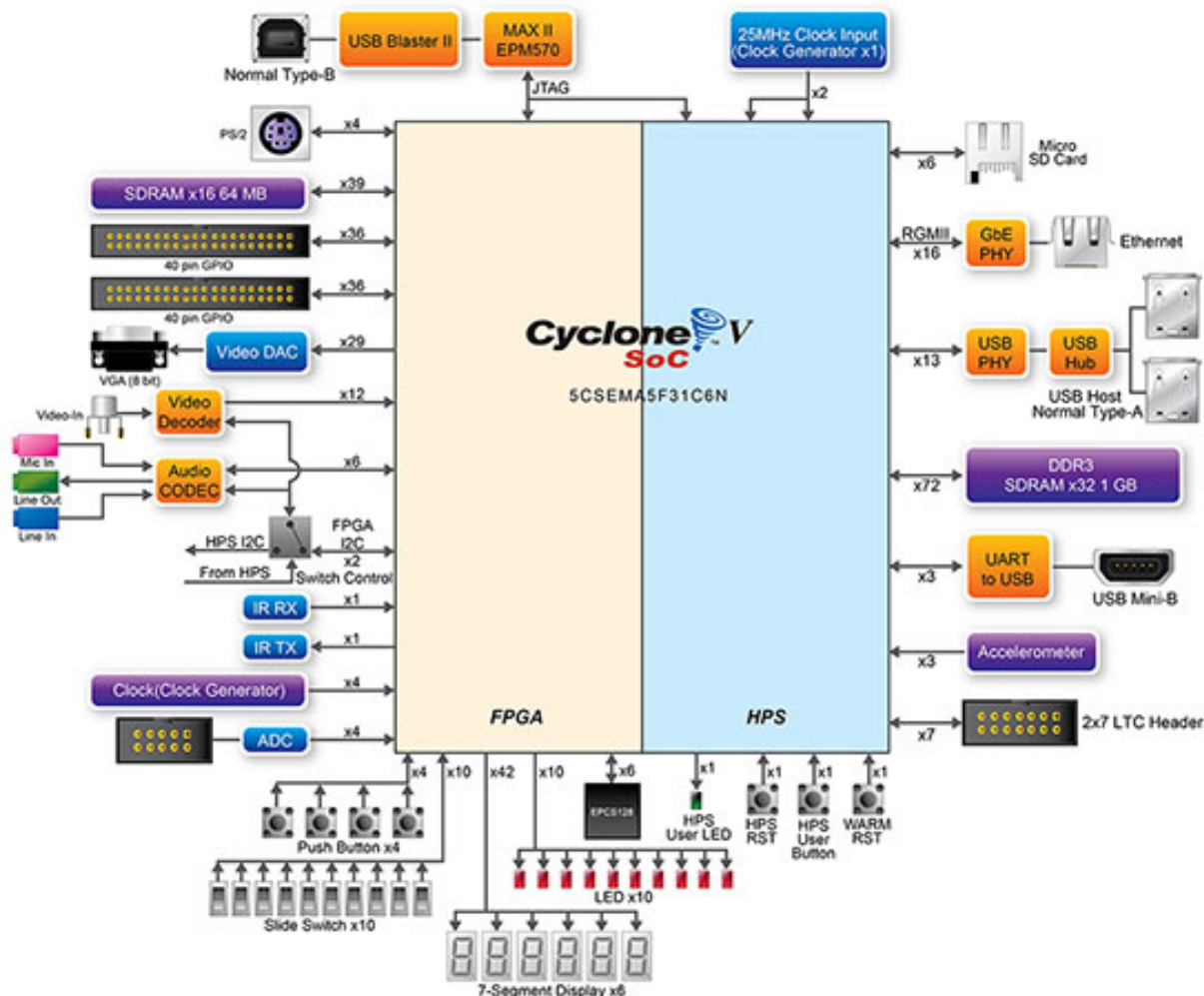


Рисунок 2.2. – Конфігурація FPGA

Конфігурація FPGA включає:

1. Двопортовий хост USB 2.0 (швидкісний інтерфейс ULPI з роз'ємом USB типу A).
2. Повноцінний USB хост з підтримкою usb-хабу (наприклад, існує можливість підключення клавіатури та миші, що дозволяє розширити використання даної FPGA).
3. USB до UART (роз'єм мікро USB типу B).

Стосовно системи дуже застосний компонент при використанні FPGA, виконана на базі FAT232.

4. 10/100/1000 Ethernet.
5. PS/2 миша/клавіатура. Остання річ т перекочувала з плати DE1.
6. ІЧ-випромінювач/приймач.

Зм..	Арк.	№докум.	Підпис	Дата

7. Два 40-контактні роз'єми розширення (рівні напруги: 3,3 В)
8. Безперечний вин.
9. Два IDC 40 конектори (36 GPIO доступно на кожному), крім того мають обмежувальні резистори і захисні діоди по кожному входу. Багато в чому схилили мене до цієї плати.
10. Один 10-контактний вхідний роз'єм АЦП.
11. Один роз'єм LTC (один головний послідовний периферійний інтерфейс (SPI), один інтерфейс I2C та один інтерфейс GPIO)

Про цей конектор має вбудований крок в 2мм.

Конектор йде з окремим комутатором ліній.

Це специфікація demo board від Linear Technologies, на відміну від GPIO від HPS.

12. Аудіо та відео.

13. 24-розрядний ЦАП VGA, 24-розрядний кодек, гнізда лінійного входу, лінійного виходу та мікрофону, декодер телевізора (NTSC/PAL/SECAM) і роз'єм для входу для ТВ.

14. АЦП .

До 1 MSPS, 8 каналів, 12 біт ЦАП.

15. Кнопки, перемикачі, індикатори.

FPGA має 4 клавіші користувача (FPGA x4), 10 перемикачів користувача (FPGA x10), 11 індикаторів користувача (FPGA x10; HPS x 1).

Також присутні важливі 2 кнопки скидання HPS (HPS_RST_n і HPS_WARM_RST_n).

16. Шість 7-сегментних дисплеїв.

Вони включають 42 дванадцять на семисегментник.

17. Датчики: G-Sensor на HPS .

18. Живлення: 12V DC input

На цій платі FPGA, на відміну від DE0-nano, не передбачено засобів вимірювання живлення струму.

					КВРКІ 190188. 14. 14 ПЗ	Арк.
						17
Зм..	Арк.	№докум.	Підпис	Дата		

2.2 Засоби розробки

З точки зору побудови програмно-апаратних засобів шляхом застосування FPGA, засоби розробки можна розділити на дві частини для дві частини: FPGA та HPS.

Для FPGA традиційно використовується середовище компанії від Altera для програмування та розробки під ПЛІС - Quartus II.

При цьому первинне налаштування матриці комутації HPS, тактування та робота з SDRAM також відбувається в Quartus II за допомогою Qsys.

Швидкий старт є як у мануалах компанії Терасик, так і від розробників рішень FPGA/

Однією з особливостей розробки є те, що більшість проєктів на сьогодні розробляються у 32-розрядній версії Quartus.

Але оскільки наступні версії Quartus II (починаючи з 14.0) будуть підтримують лише 64-розрядну ОС, то розробники виконали належну підтримку 64-розрядних рішень.

Для HPS вибір досить великий.

Але в першу чергу необхідно поставити SoC Embedded Design Suite заснований на ARM Development Studio 5 (DS-5) Altera Community Edition Toolkit навіть якщо їй не користуватися до неї в навантаження йде тулчейн для роботи з АРМ від Альтери.

Також необхідно встановити званий Embedded Command Shell, що дозволяє виконувати UNIX додатки під Windows.

Детальна інструкція з встановлення надана компанією Terasic як обов'язкова.

Таким чином, для розробки програм HPS застосовні такі способи:

1. Можливість застосування блокнот для написання мейк-файлів для плати (основний спосіб пропонований у мануалах terasica) при цьому на платі має бути встановлений Лінукс.

2. Розробка під Altera University Program. Такий спосіб передбачає застосування образ системи, дозволяє проводити налагодження за асемблерним

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		18

кодом, а також має ряд прикладів і безкоштовних IP для конфігурації плати та обробки відео. .

3. ARM Development Studio 5 (DS-5) – потужний інструмент, що дозволяє робити все від написання «голих» додатків (Bare Metal), так і написання та налагодження драйверів під Linux, складання системи та ще купу плюшек. У безкоштовній версії доступне лише написання програмного забезпечення під ОС Linux.

4. Можлива розробка додатків Qt на C++. Докладно як це зробити, також розписано в мануалі від компанії Terasic.

Щоправда, там вони пропонують розробляти під Ubuntu у віртуальній машині.

2.3 Linux BSP (пакет підтримки плат)

Як можна зрозуміти, з наведеного вище тексту, написання програмного забезпечення для плати, без покупки ліцензій, можливе лише під ОС Linux.

Terasic пропонує вже готовий образ системи, тільки на флешку записати, та в слот на платі вставити.

Компанією Terasic реалізовано викотили 5 образів для розробки.

Образи засновані на ядрі версії 5.12, в якому або спеціально демонтовано частина драйверів для роботи із зовнішньою периферією.

Наприклад, згадана плата DC934A не працює, тому що вимагає наявності /sys/class /gpiu якого немає, але який є у версії ядра 5.13, що поставляється з DS-5 Studio.

Наразі актуальні для розробки образи:

1. Linux Console – базові функції та консольне введення/виведення. Вимагає флеш-накопичувача 4ГБ.

2. Linux Console with framebuffer - теж саме, але з фреймбуфер підходить для додатків на Qt, розмір той же.

3. Linux LXDE Desktop – підтримує графічну систему LXDE, потребує 8ГБ флеш-накопичувача.

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		19

4. Linux Ubuntu Desktop - так на плату можна поставити Ubuntu, теж не менше 8ГБ флеші.

5. DE1-SoC OpenCL BSP - вийшла зовсім недавно - це Linux для використання при розробці OpenCL додатків.

OpenCL дозволяє розробляти програмне забезпечення на Сі з векторною обробкою, при цьому їх компілятор намагатиметься створити відповідну схему на FPGA і таким чином значно прискорити обчислення.

При цьому програмісту не потрібно детально вникати в архітектуру ссамої ПЛІС.

Незважаючи на наявність інструкцій від компанії Terasic із запуску цих образів, існує суттєва відмінність між командами LS і CD

2.4 Розробка програмно-апаратних засобів засобами Quartus

Quartus має ряд функціональних можливостей для поліпшення управління проектами плат та їх файлової структури.

Ці можливості включають повністю синхронізовану зміну імені проекту і проектних файлів, інтелектуальну обробку порушень зв'язку локального і віддаленого проекту і опції розумного очищення даних, які представляють концепцію оборотного видалення.

Перейменування проекту та проектних файлів.

У контекстному меню панелі Projects доступна команда Rename для перейменування файлів, яка замінює команду Save As .

Вона дозволяє змінити ім'я проекту плати (*.PrjPcb) або будь-якого з файлів, що входять до його складу (*.PcbDoc, *.SchDocі т.д.) і потім зберегти зміну локально або у віддаленому сховищі системи контролю версій (File » Save to Server).

Важливо звернути увагу, що команда Save As залишається доступною в головному меню File, а проектний файл також можна перейменувати за допомогою контекстного меню на панелі Storage Manager (рисунок 2.3).

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		20

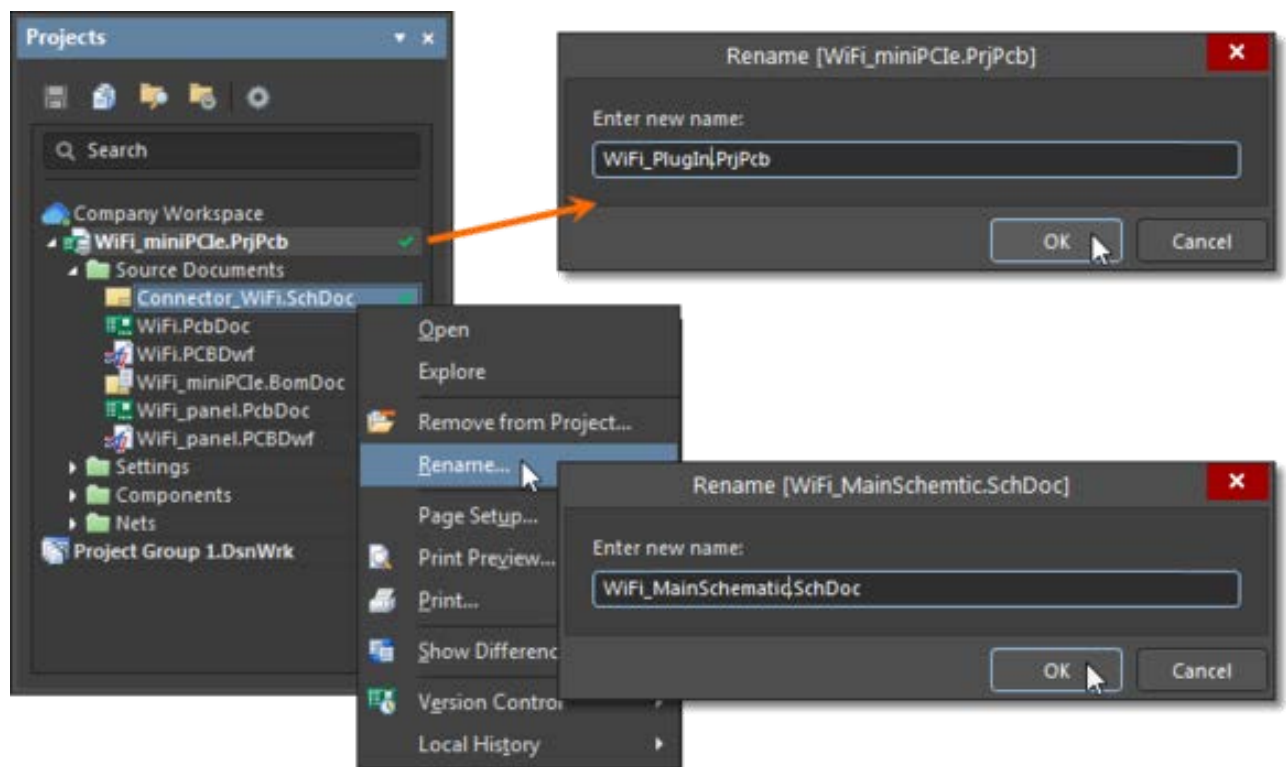
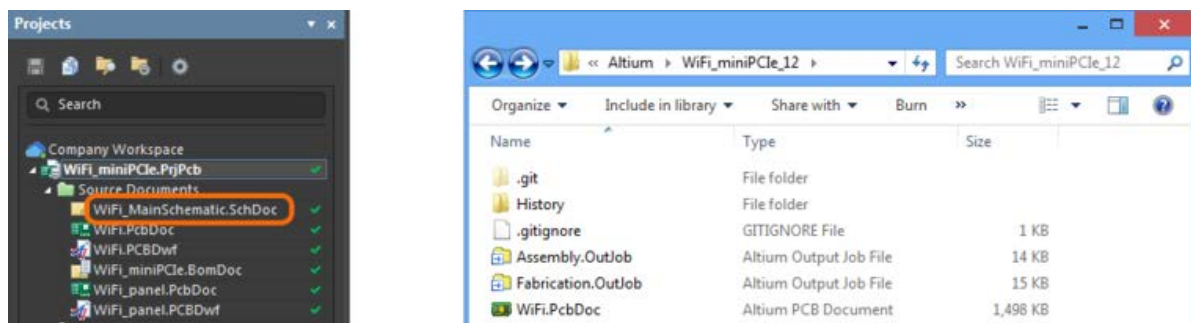


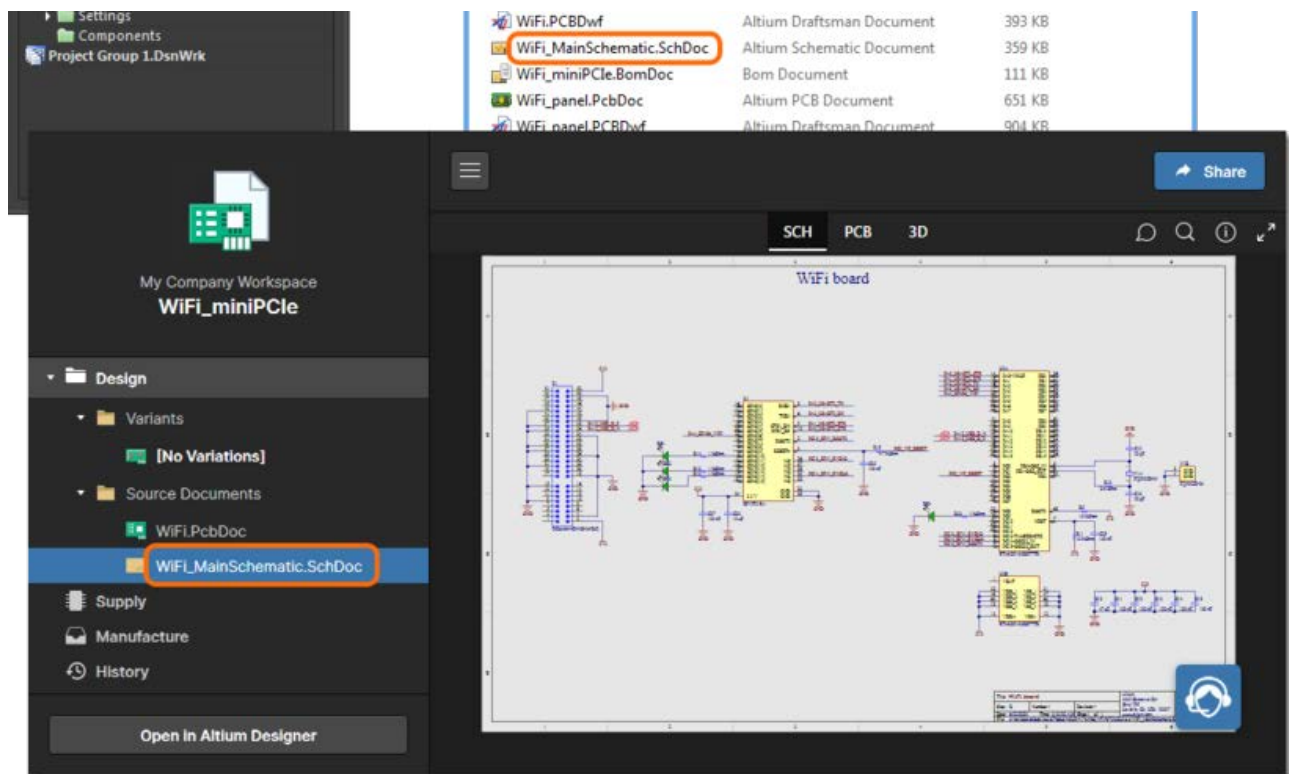
Рисунок 2.3 – Інтерфейс збереження проєкту

Підтримується повна синхронізація зміни імені файлів між локальним та віддаленим серверним сховищем.

Крім того, при перейменуванні файлу проєкту буде змінено ім'я проєкту на стороні сервера, що вказано в наступному розділі (рисунок 2.4).



а)



б)

Рисунок 2.4 – Інтерфейс проекту

Перейменування керованого проекту.

Пряма зміна імені файлу проекту (*.PrjPcb) на стороні системи проектування також підтримує автоматичну зміну імені проекту на стороні сервера.

Це означає, що при синхронізації зміненого імені проекту з сервером сервер автоматично змінює параметр Name керованого проекту для забезпечення відповідності.

При зміні імені керованого проекту в серверному просторі (за допомогою режиму Edit на сторінці Projects сервера або на панелі Explorer клієнтської програми).

Ця зміна передається на бік клієнтської програми при наступному відкритті проекту (рисунок 2.5).

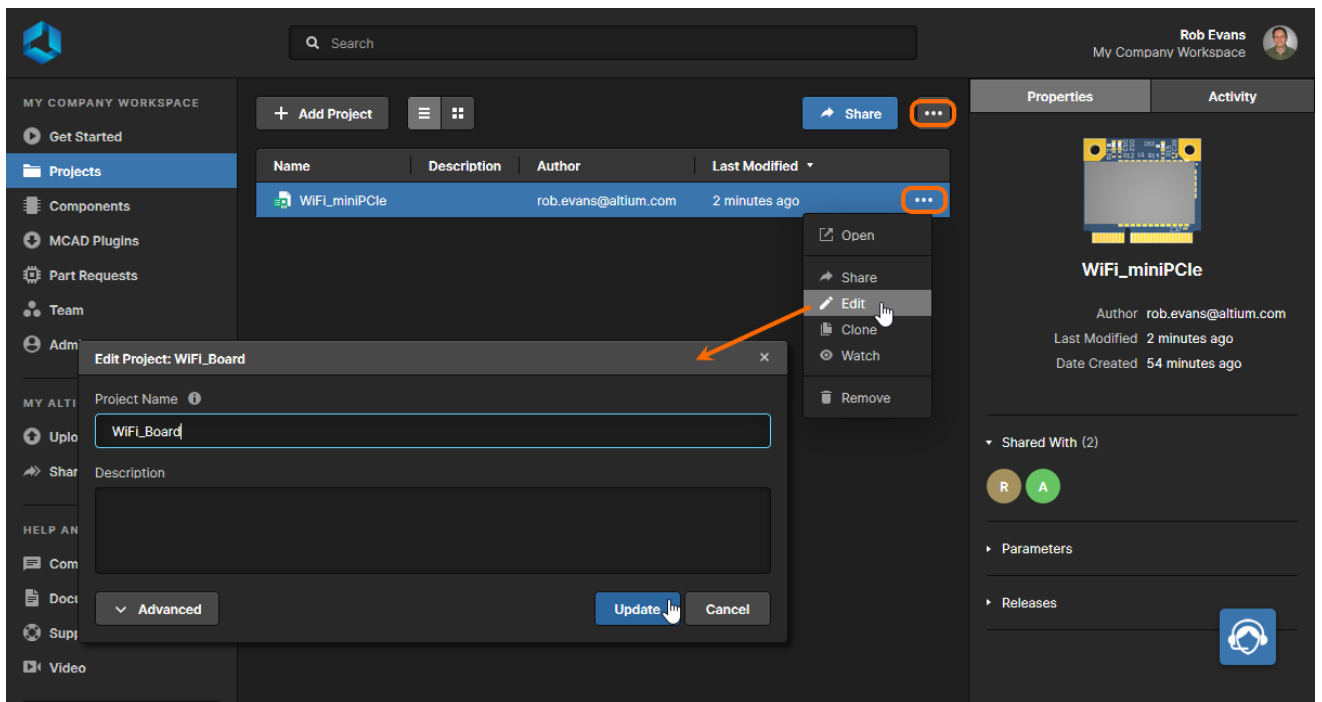


Рисунок 2.5 – Інтерфейс проекту

При відкритті зміненого проекту (File » Open Project) буде відкрито діалогове вікно, що пропонує опції присвоєння файлу проекту нового значення параметра Name проекту, присвоєння параметру Name проекту існуючого імені файлу проекту та збереження різних імен проекту.

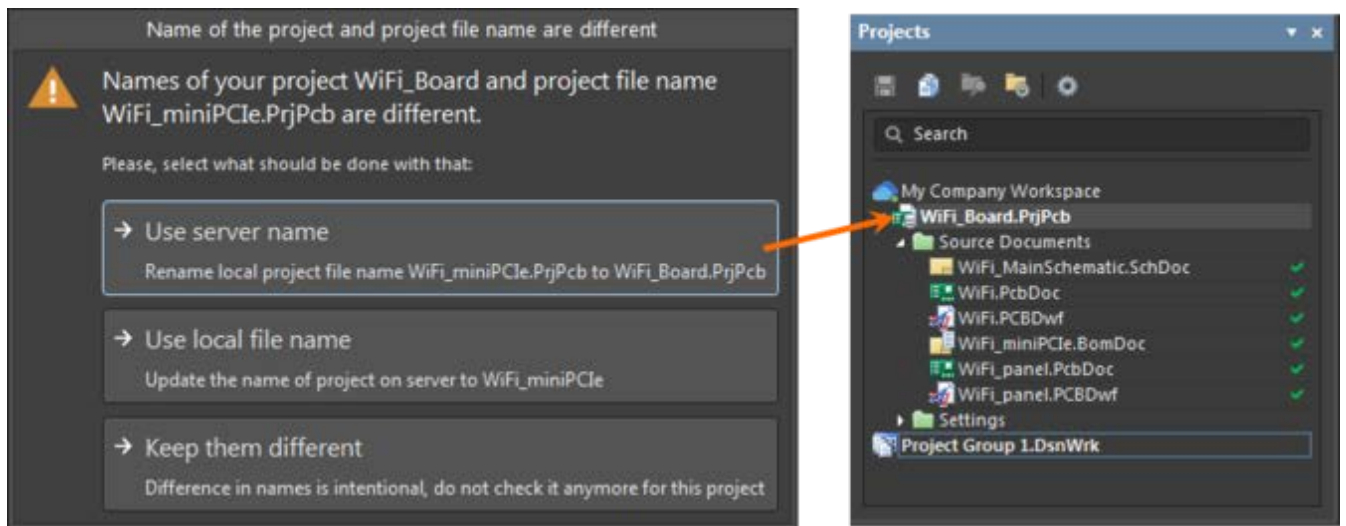


Рисунок 2.6 – Інтерфейс відкриття проекту

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

Історія керованого проекту.

При роботі з серверними керованими проектами Altium Designer надає прямий доступ до нових функціональних можливостей відстеження історії проекту в Altium 365 Workspace.

Інтерактивна графічна шкала представляє історію файлу проекту у вигляді набору хронологічних версій на основі подій, які, залежно від типу події, можна переглянути, до яких можна зробити відкат або доступ до даних яких можна отримати доступ. Повні можливості тимчасової шкали історії доступні для серверних проектів під управлінням власного (внутрішнього) контролю версій Git у Workspace.

Для отримання більш детальної інформації перейдіть на сторінку Історія проекту у Workspace .

Щоб отримати доступ до історії серверного проекту з Altium Designer, виберіть команду History з меню Projects або клацніть ПКМ на ім'я проекту в панелі Projects та виберіть команду History з контекстного меню (дивись рисунок 2.7).

Щоб відкрити історію проекту безпосередньо у веб-інтерфейсі Workspace, відкрийте проект на сторінці Projects Management (Projects) та виберіть опцію History у бічному меню.

Публікація проекту.

Поряд із публікацією проектів на основі прав доступу для членів команди Altium 365 Workspace , тепер доступна публікація проекту учасникам, які не є членами команди Workspace або які не зареєстровані на даний момент в AltiumLive .

Це дозволяє зовнішнім та внутрішнім співробітникам переглядати або редагувати (за наявності відповідних прав доступу) певний проект.

Ті, хто знаходяться поза вашою командою Workspace, можуть отримати керований доступ до перегляду динамічного проекту (менеджери та інші зацікавлені особи) або редагування самого проекту (зовнішні підрядники і т.д.).

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		24

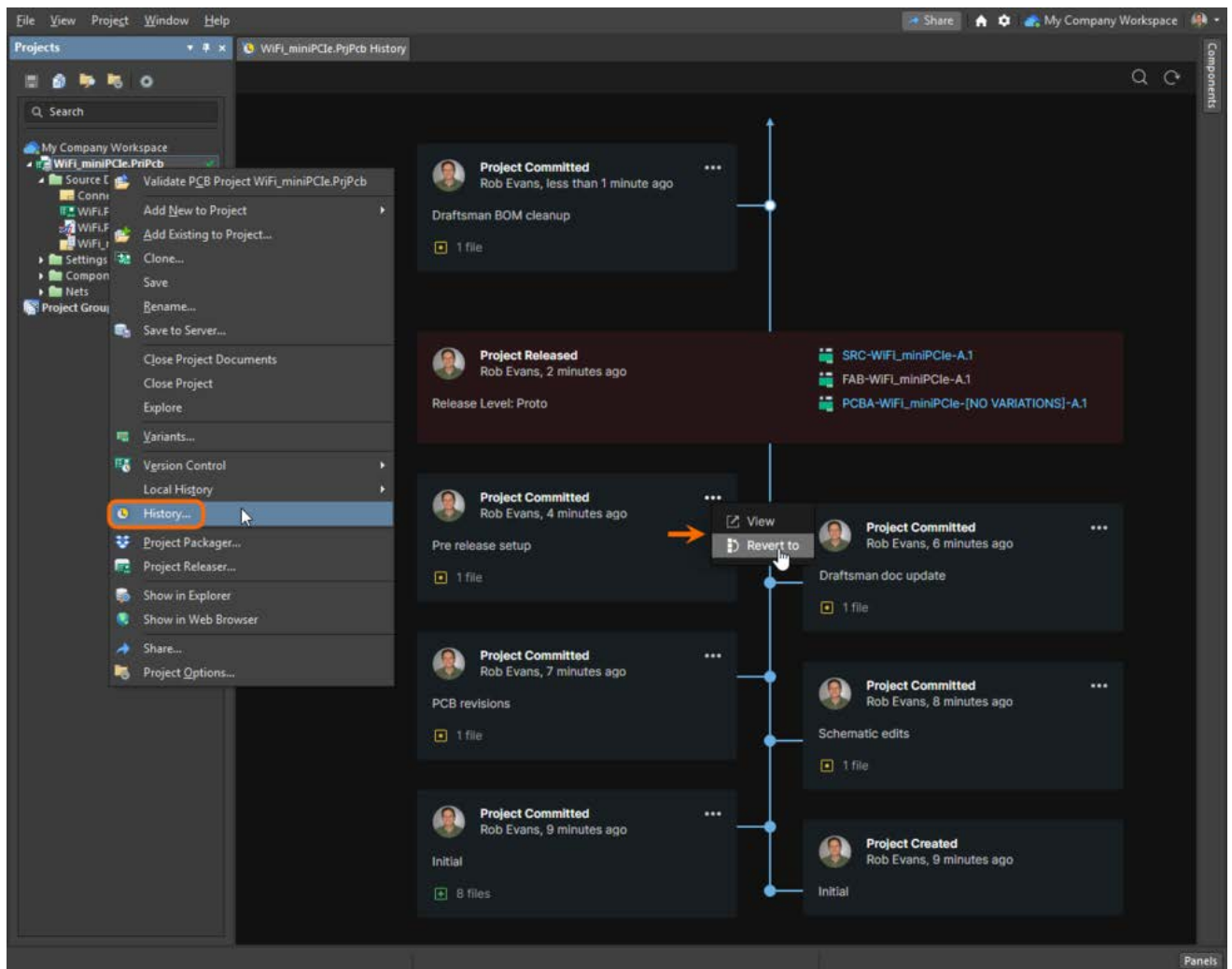


Рисунок 2.7 – Інтерфейс відкриття проєкту

Щоб отримати докладнішу інформацію, перейдіть на сторінку Глобальна публікація.

Проекти, опубліковані з Altium Designer або через веб-інтерфейс Workspace, стають доступними запрошеному користувачеві або групі на винятковій основі, без доступу до будь-яких інших даних Workspace.

Користувачам, які не мають облікового запису AltiumLive (який необхідний), надаються кошти для його обов'язкової реєстрації через email-запрошення.

Доступ до проєкту, опублікованого користувачеві Altium Designer, здійснюється за допомогою опції Shared With Me у діалоговому вікні Open Project. Користувачеві не потрібно бути членом команди Workspace (рисунок 2.8).

Зм..	Арк.	№докум.	Підпис	Дата

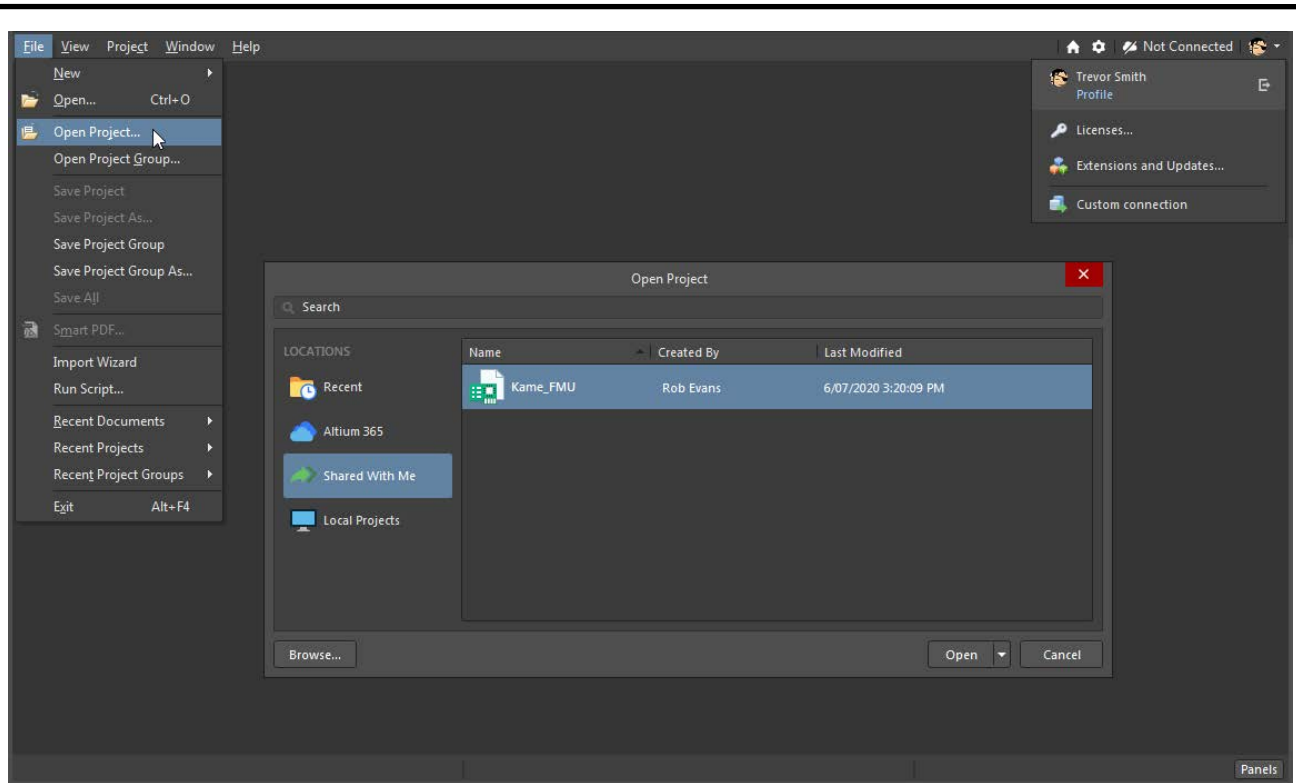


Рисунок 2.8 – Інтерфейс Workspace проєкту

У панелі Projects, проєкт, який був відкритий із загального доступу, позначається відповідним текстом Shared with me .

Проєкт може бути збережений та зафіксований на сервері, якщо були надані права на редагування опублікованого проєкту, або збережено/зафіксовано локально, якщо проєкт був опублікований з доступом лише на перегляд змісту проєкту.

Щоб відкрити проєкт у форматі режиму веб-перегляду Altium, необхідно обрати Show in Web Browser у контекстному меню поточного проєкту на панелі Projects (рисунок 2.9).

Також вадливо звернути увагу, що для опублікованого проєкту, відкритого в Altium Designer, є коментування документів , що дозволяє здійснювати пряму динамічну взаємодію між користувачем, який опублікував проєкт, і користувачем, кому цей проєкт опублікований.

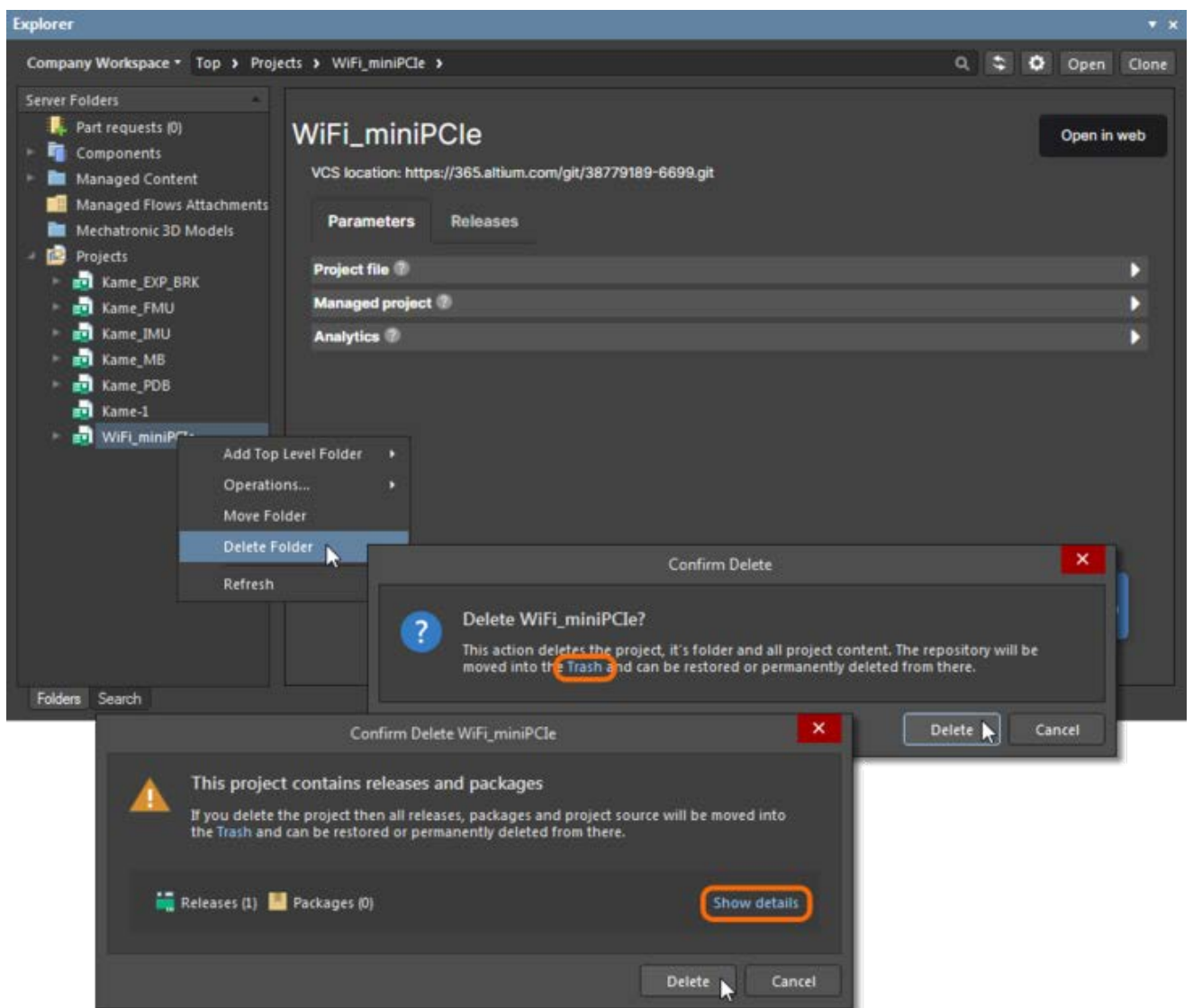


Рисунок 2.10 – Інтерфейс зворотного видалення проєкту

У Workspace віддалені об'єкти переміщуються в спеціальну область Trash , звідки їх можна відновити (Restore) або повністю видалити (Permanently Delete) за допомогою опцій меню віддаленої сутності (...).

Інтерфейс Workspace проєкту подано на рисунку 2.11.

Панель Comments доступна без входу в обліковий запис Altium або підключення до сервера, і тут надається посібник з наступних кроків для включення коментування проекту.

Панель Comments подано на рисунку 2.13.

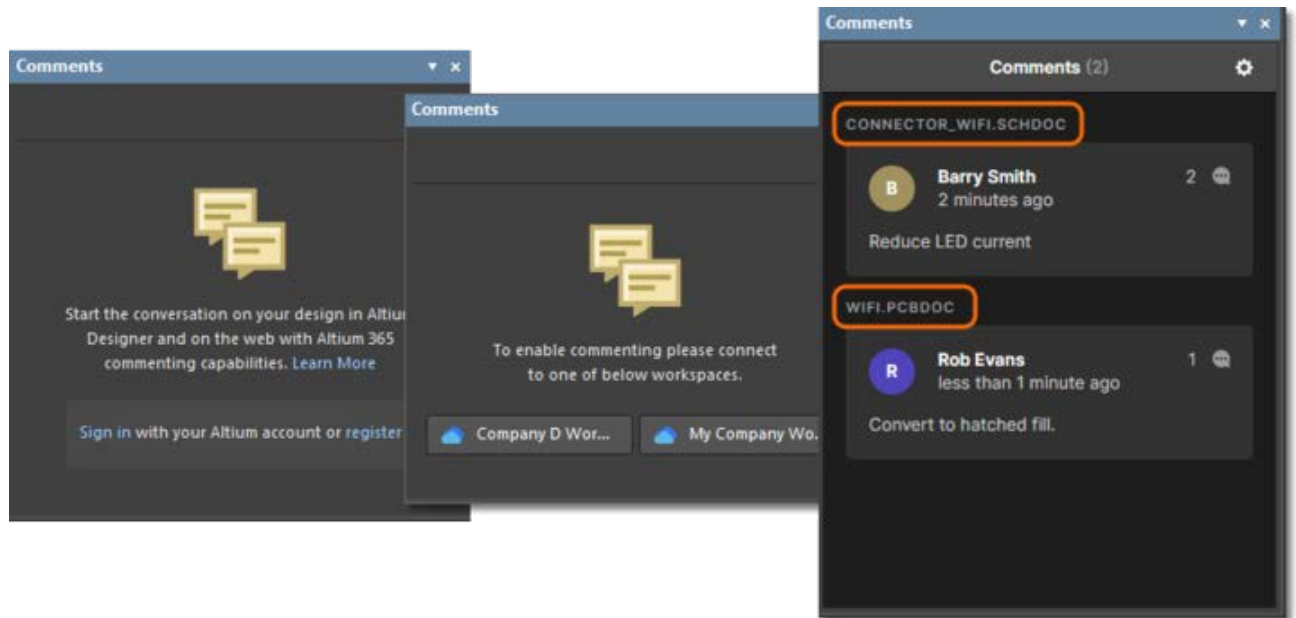


Рисунок 2.13 – Панель Comments

Синхронізація робочої копії проекту із сервером.

Використання повністю керованих проектів плат, що знаходяться під управлінням контролю версій, ґрунтується на тісній синхронізації між даними у локальній робочій папці проекту із сховищем контролю версій сервера.

Якщо цей зв'язок порушено змінами, внесеними нестандартними процесами, структура керованого проекту може бути пошкоджена.

Зміни, які можуть порушити синхронізацію локального та віддаленого сховища, включають ручні операції, такі як перейменування, переміщення, копіювання/клонування робочої папки проекту.

Такі проблеми можуть бути виявлені та усунені за допомогою діалогового вікна з варіантами вибору, де виділено рекомендовану дію на основі ситуації.

Ці опції дозволяють зробити папку проекту новим керованим проектом, повторно синхронізувати папку проекту як поточний керований проект, видалити

зв'язок проекту із сервером (зробити проект некерованим) або ігнорувати поточну невідповідність.

Процес внесення змін в поточний проект для коректної роботи подано на рисунку 2.14.

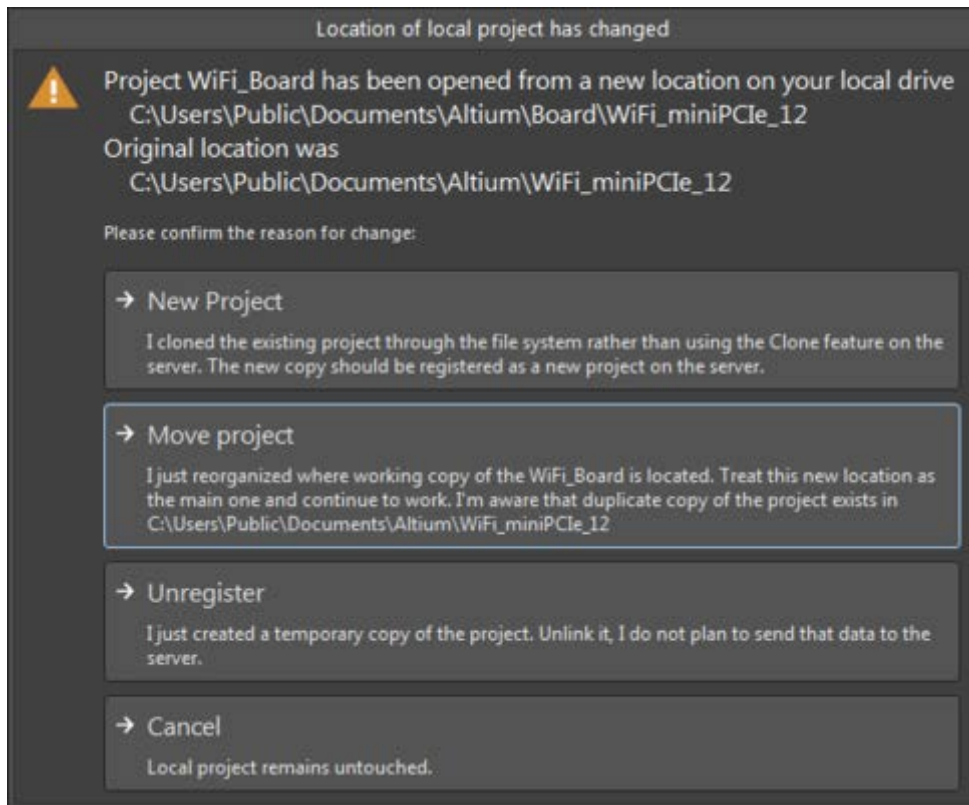


Рисунок 2.14 – Процес внесення змін в поточний проект для коректної роботи

Інші складніші порушення синхронізації можуть бути викликані змінами серверної сутності, наприклад, коли сам сервер був перейменований або переміщений, коли локальний файл проекту був перезаписаний файлом, що містить інші параметри серверної сутності або коли цільовий репозиторій проекту був змінений.

Такі проблеми створюють невідповідність між серверними та локальними репозиторіями проекту, і звіт по них наводиться в діалогових вікнах із попередженнями або опціями для усунення проблеми.

Робота з репозитарієм показна на рисунку 2.15.

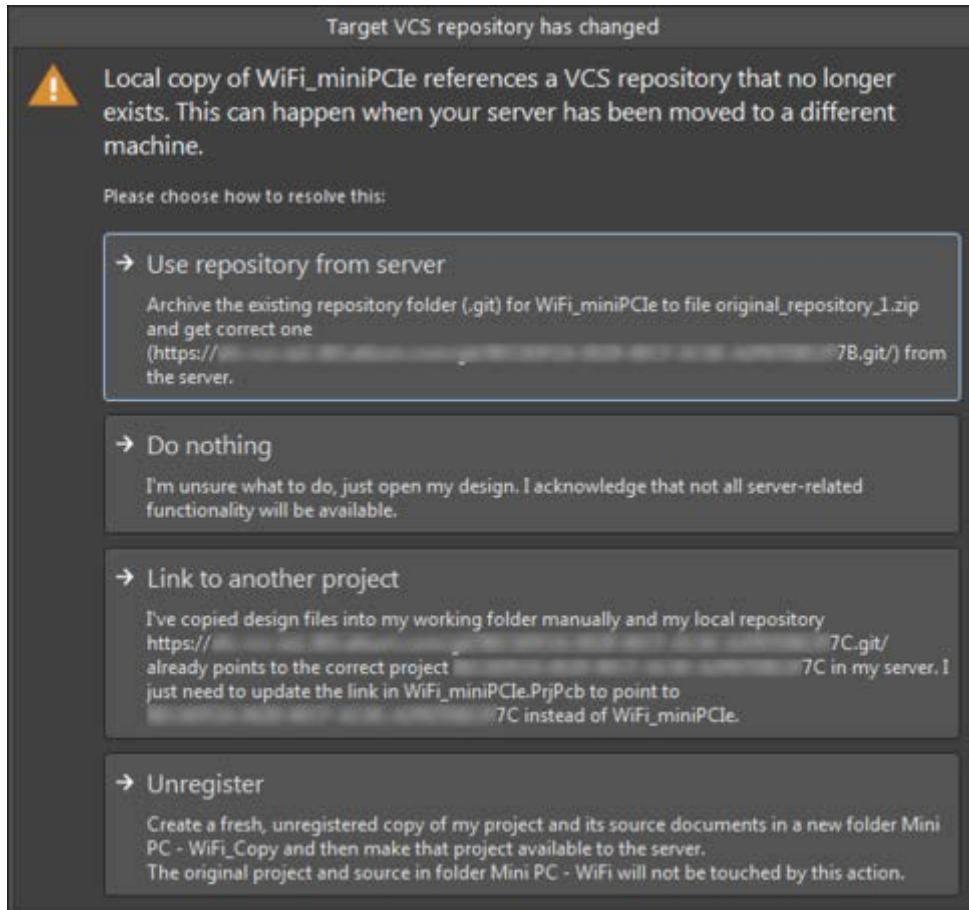


Рисунок 2.15 – Робота з репозитарієм показна на рисунку

2.5 Редактор плат

Розрахунок часу розповсюдження сигналу в перехідному отворі.

Розрахунок часу поширення сигналу в ланцюгах друкованої плати тепер враховує довжину підключених перехідних отворів, що використовується, виходячи з пропорції часу, зазначеного у властивості Propagation Delay.

Це означає, що довжина перехідного отвору визначається початковим і кінцевим шаром, в який входить і з якого виходить ланцюг відповідно.

Цей розрахунок часу поширення сигналів узгоджується з розрахунками довжини ланцюгів, які враховують частковий шлях проходження сигналу через підключені перехідні отвори.

Інтерфейс роботи з редактором плат подано на рисунку 2.16.

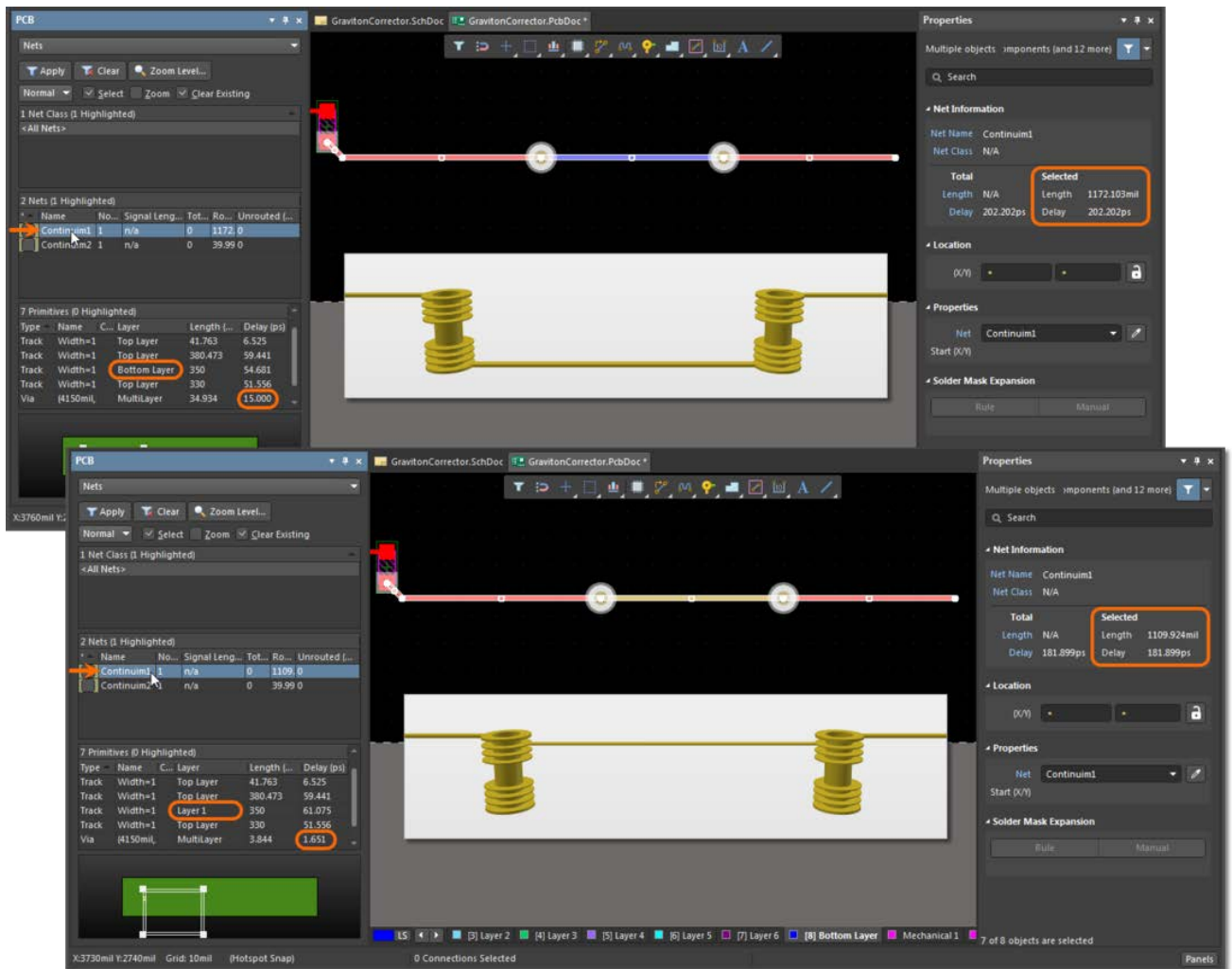


Рисунок 2.16 – Інтерфейс роботи з редактором плат

2.6 Редактор компонентів

Перехресний перехід у редакторі компонентів.

Тепер під час редагування серверного керованого компонента в редакторі компонентів у режимі Single Component Editing доступне підсвічування висновків символу та посадкового місця.

Редактор може бути викликаний вибором опції Edit із контекстного меню рядка компонента на панелі Components.

Редактор компонентів подано на рисунку 2.17.

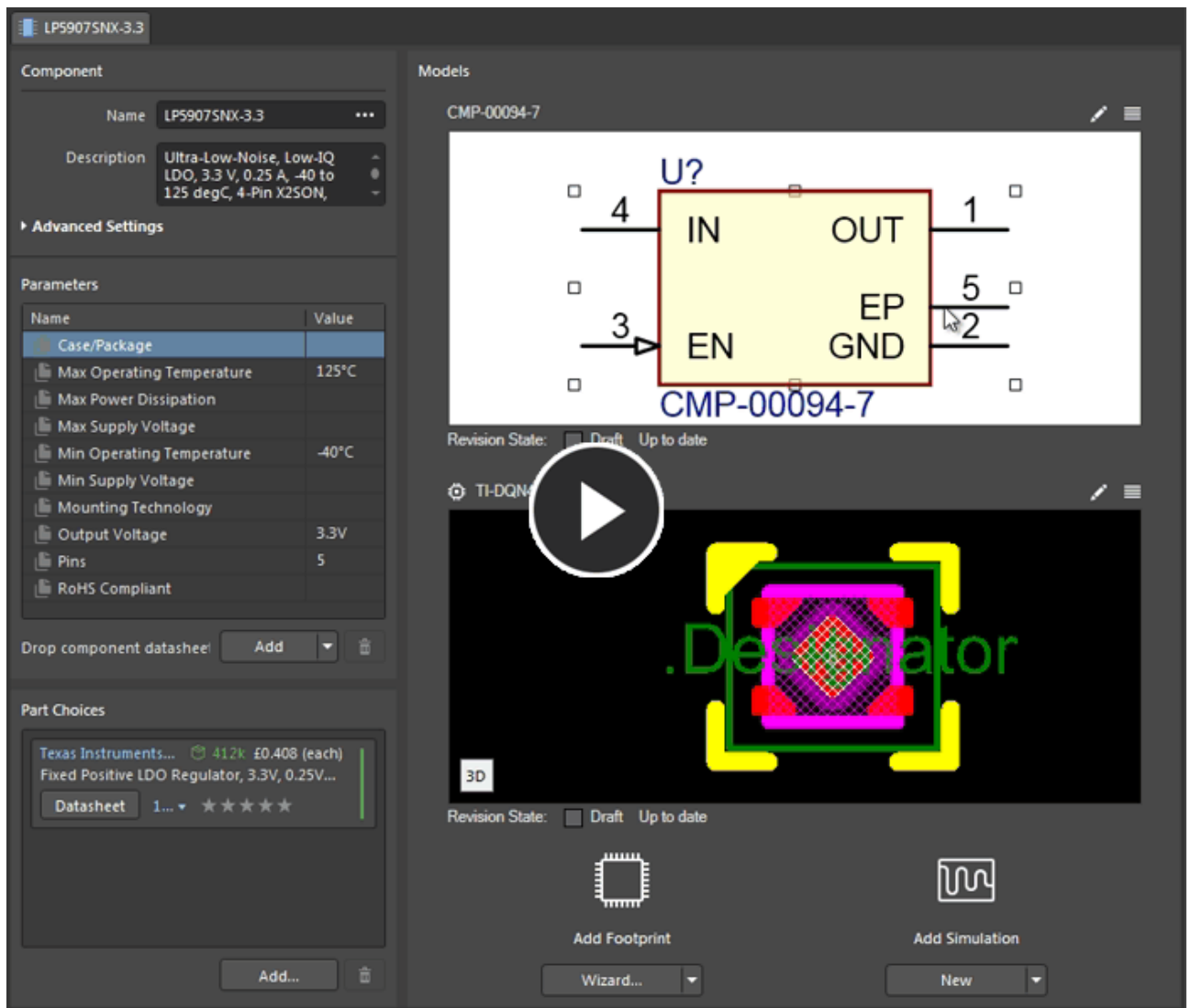


Рисунок 2.17 – Редактор компонентів подано

2.7 Взаємодія ECAD-MCAD

Технологія CoDesigner від Altium пов'язує галузі проектування механіки та електроніки, що дозволяє інженерам механічної та електронної частин виробу передавати плату між цими областями.

CoDesigner продовжує свій розвиток у рамках нової версії:

1. Визначення гнучко-жорстких плат у MCAD-системі – було внесено суттєві поліпшення підтримки гнучко-жорстких плат. Форма плати та місця згину тепер можуть бути задані та змінені в SOLIDWORKS®, і компоненти тепер можуть бути розміщені та переміщені у SOLIDWORKS.

2. Розширена підтримка складних форм плати – було покращено підтримку визначень форми плати. Мікросегменти та самоперетинання в ескізі плати в ECAD-системі тепер виявляються під час відправлення. Наводиться інформація щодо їх розташування, і вони автоматично виправляються в MCAD-системі, де це можливо.

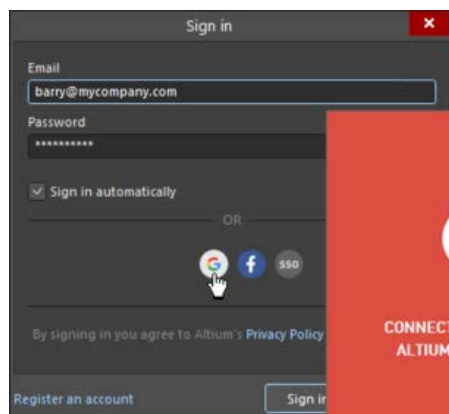
3. Покращена синхронізація компонентів – було спрощено обробку заблокованих компонентів. Зафіксовані у MCAD-системі компоненти тепер транслюються у заблоковані компоненти в ECAD-системі, і заблоковані у ECAD-системі компоненти стають заблокованими у MCAD-системі. Зміни стану блокування тепер синхронізуються між MCAD- та ECAD-системами.

4. Покращена передача властивостей компонентів ECAD – всі параметри компонентів на платі в ECAD-системі передаються в компоненти в MCAD-системі.

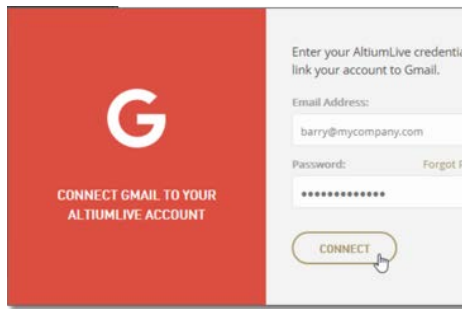
Для надання більшої гнучкості при підключенні до платформи Altium 365 Altium Designer тепер дозволяє здійснювати сторонній вхід в AltiumLive.

Щоб зв'язати опцію альтернативного входу зі своїм обліковим записом, клацніть ЛКМ по потрібному підключенню (тут показано для Google ®) і увійдіть до цього сервісу, якщо потрібно.

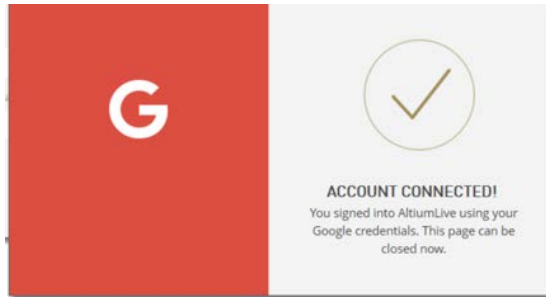
Введіть існуючі облікові дані AltiumLive на наступній сторінці, щоб встановити сторонній зв'язок з вашим обліковим записом (рисунок 2.18).



a)



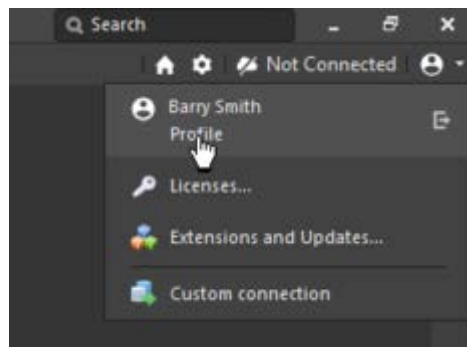
б)



в)

Рисунок 2.18 – Облікові дані AltiumLive на сторінці, щоб встановити сторонній зв'язок з обліковим записом

Підтвердження зв'язку облікового запису можна побачити в обліковому записі Altium, де також можна видалити цей зв'язок, якщо необхідно (Unlink).



а)

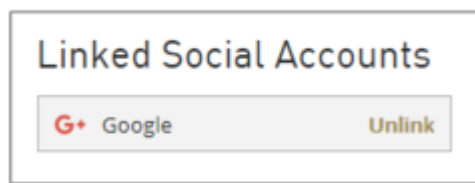


Рисунок 2. 19 – Підтвердження зв'язку облікового запису

Зм..	Арк.	№докум.	Підпис	Дата

Єдиний вхід.

Для покращення сумісності з існуючими системами компанії, можете здійснювати вхід до Altium Designer за допомогою системи єдиного входу (Single Sign-On, SSO) своєї організації.

Крім переваг підвищеної безпеки та більш ефективного з точки зору ІТ управління доступом користувачів, застосування схем SSO забезпечує зручність входу до ПЗ Altium за допомогою тих самих облікових даних.

Вони використовуються в інших системах та службах компанії, що використовує даний сервіс.

Налаштування облікового запису Altium компанії для SSO здійснюється адміністратором облікового запису на сторінці системи Authentication в Altium Dashboard .

Використання облікового запису Altium подано на рисунку 2.20.

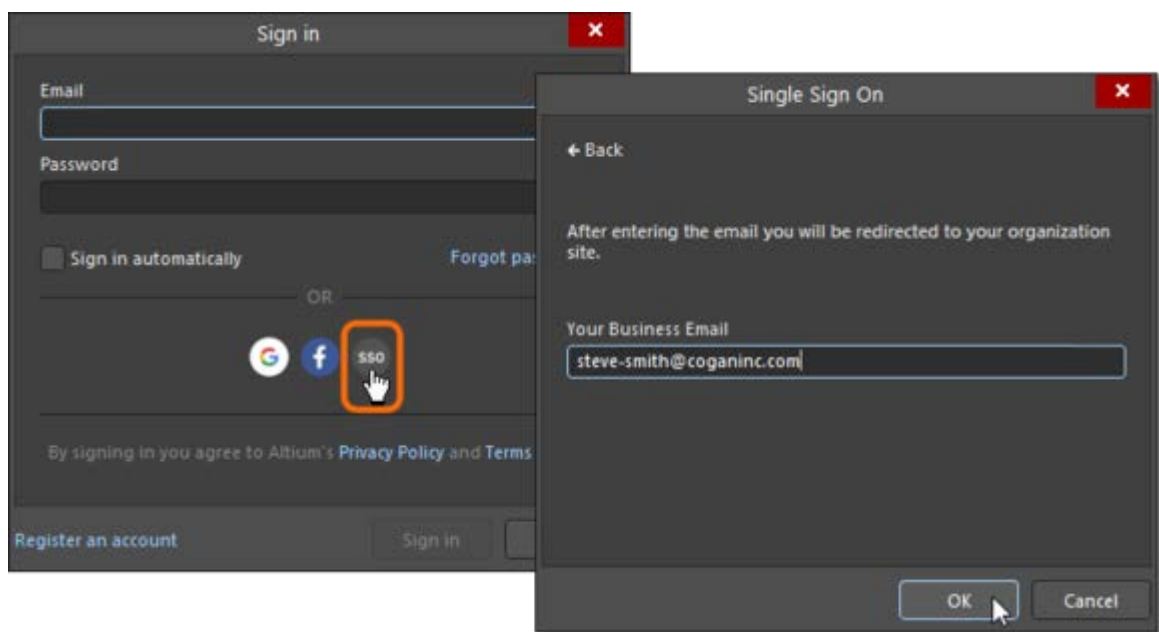


Рисунок 2.20 – Використання облікового запису Altium

Щоб отримати докладнішу інформацію, перейдіть до розділу Єдиний вхід сторінки ліцензування.

Крім Altium Designer, можна створити сеанс SSO при вході в AltiumLive або Altium 365 Workspace .

Оновлення Library Migrator.

В системі існує ряд можливостей засобами інтерфейсу Library Migrator та його функції обробки бібліотек, у тому числі:

1. Загальні поліпшення зручності інтерфейсу користувача.
2. Розширення логіки виявлення ключових слів параметрів для покращення автоматичного поділу бібліотек.
3. Покращена обробка відсутніх символів/посадкових місць компонентів.

Проблема відсутніх посадкових місць тепер усувається автоматично додаванням бібліотеки, яка містить потрібні посадкові місця.

Відсутні посадкові місця тепер коректно виявляються процесом валідації, де також надається опція Choose a Library у звіті про наявну помилку (у розділі Details).

Якщо немає одного посадкового місця, діалогове вікно Apply PcbLib Selection надає опції для використання певної бібліотеки для цього або всіх компонентів з відсутніми посадковими місцями.

Робота з діалоговим вікном Apply PcbLib Selection представлено на рисунку 2.21.

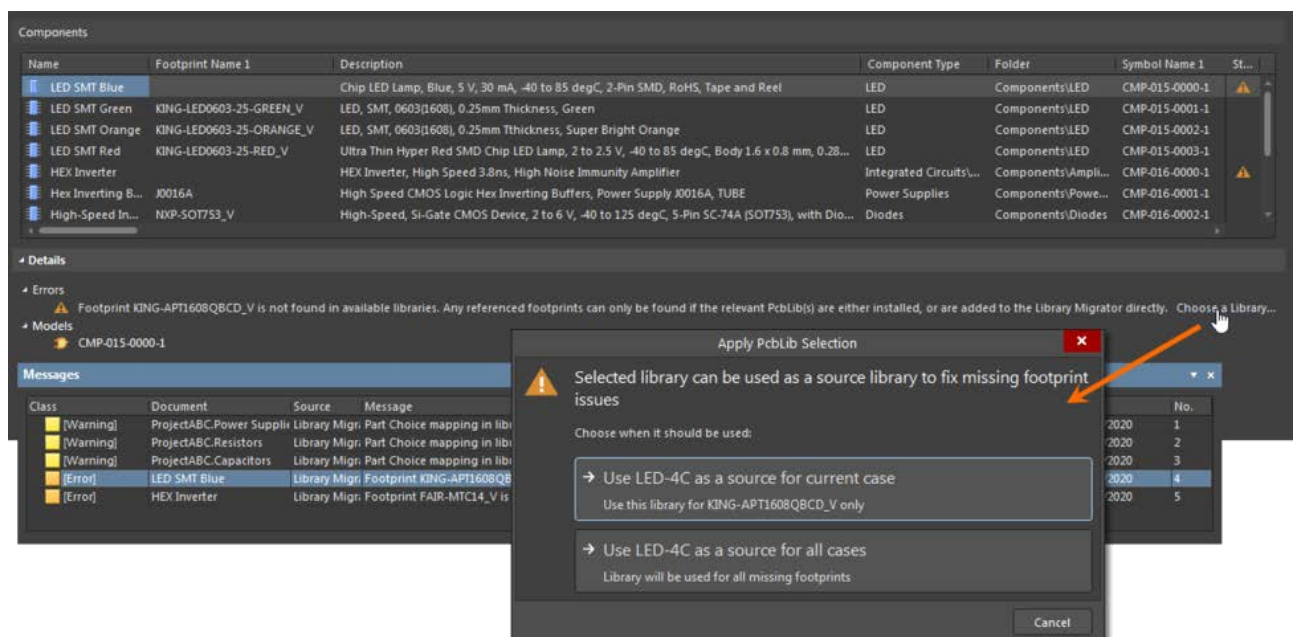


Рисунок 2.21 – Робота з діалоговим вікном Apply PcbLib Selection

Оновлення безпеки Altium Designer.

Altium Designer включає безпеку віддалених підключень завдяки наступним оновленням Altium Designer:

1. Збережені паролі для входу тепер зашифровані.
2. Сертифікати HTTPS тепер підтверджені.
3. Забезпечена підтримка TLS 1.2 .

Важливо звернути увагу на те, що в рамках цих оновлень було змінено адресу порталу облікових записів за замовчуванням.

2.8 Висновки

В розділі подано аналіз функційних можливостей плати розробника DE1-SoC.

В розділі наведено комплект поставки, технічні характеристики та основана конфігурація.

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		39

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ ВІДЕО-ДЕКОДЕРА НА БАЗІ FPGA ALTERA CYCLONE V

3.1 Створення прошивки FPGA

Для створення прошивки було створено проект Quartus.

Але створино його було не стандартним способом (через project wizard), а через утиліту, яка йшла в комплекті з платою DE1-SoC (рисунок 3.1).

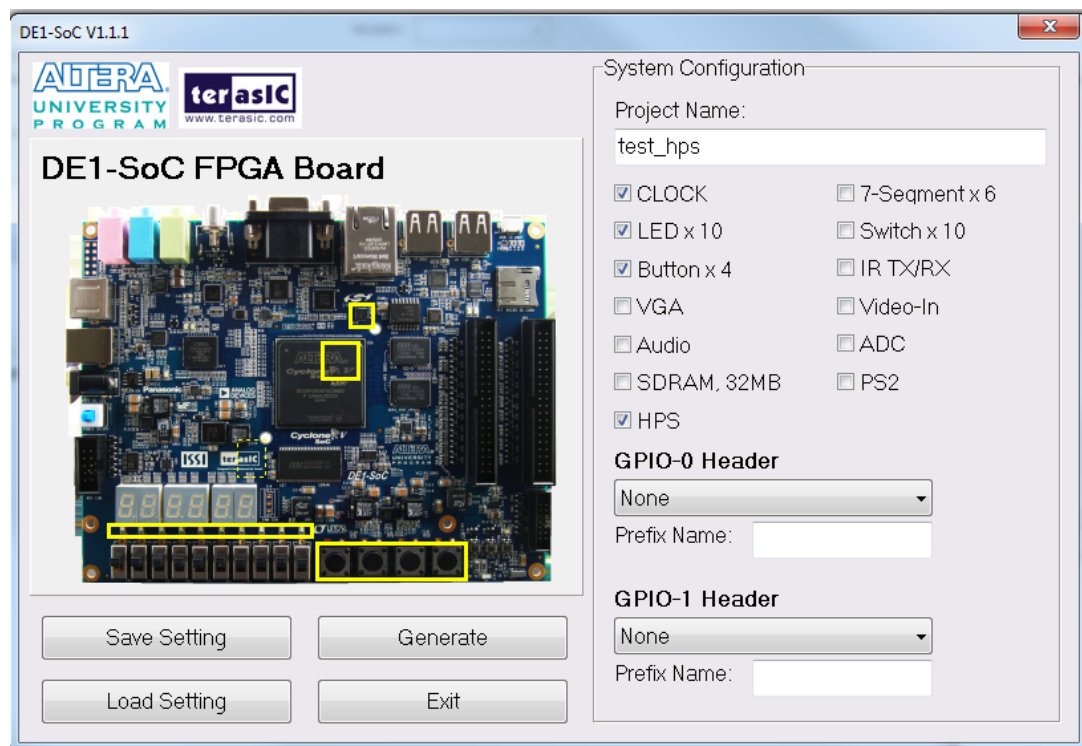


Рисунок 3.1 – Quartus project wizard

Ця утиліта генерує top-level файл написаний Verilog з оголошеннями вибраних елементів.

Було залучено компоненти CLOCK, HPS (SoC), кнопки, світлодіоди та інші. Натискаючи Generate отримуємо проект Quartus.

Головна перевага створення проекту цим способом, це те, що не доведеться призначати піни FPGA в Pin Plannerю

Таким чином, утиліта заощадила багато часу (рисунок 3.2).

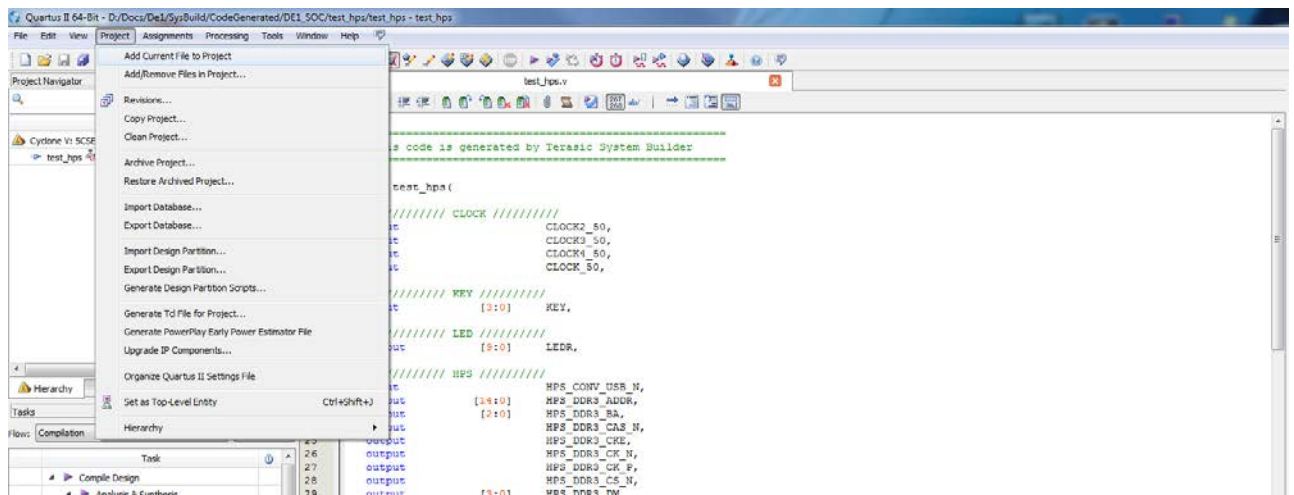


Рисунок 3.2 – Інтерфейс проекту

Наступним кроком було створено систему QSYS.

Cyclone V SoC не просто FPGA, всередині її структури знаходиться двоядерний процесор Cortex-A9 із модулями:

- 1) USB порти;
- 2) Ethernet;
- 3) SPI;
- 4) SD/MMC;

тощо.

По-простому, це як мікроконтролер всередині FPGA.

Створюючи систему QSYS було пов'язано HPS (hard processor system) систему з елементами, синтезованими в FPGAю

При цьому було використано готові ядра (IP cores).

QSYS дозволило без зайвих турбот з'єднати різні ядра через шини Avalon-MM або AMBA AXIю

Не було необхідності вручну писати код, QSYS генерує його автоматично.

Заходимо в QSYS і вибираємо у вкладці IP cores систему HPS.

У налаштуваннях системи знадобиться лише Lightweight H2F Bridge у вкладці FPGA interfaces (рисунок 3.3).

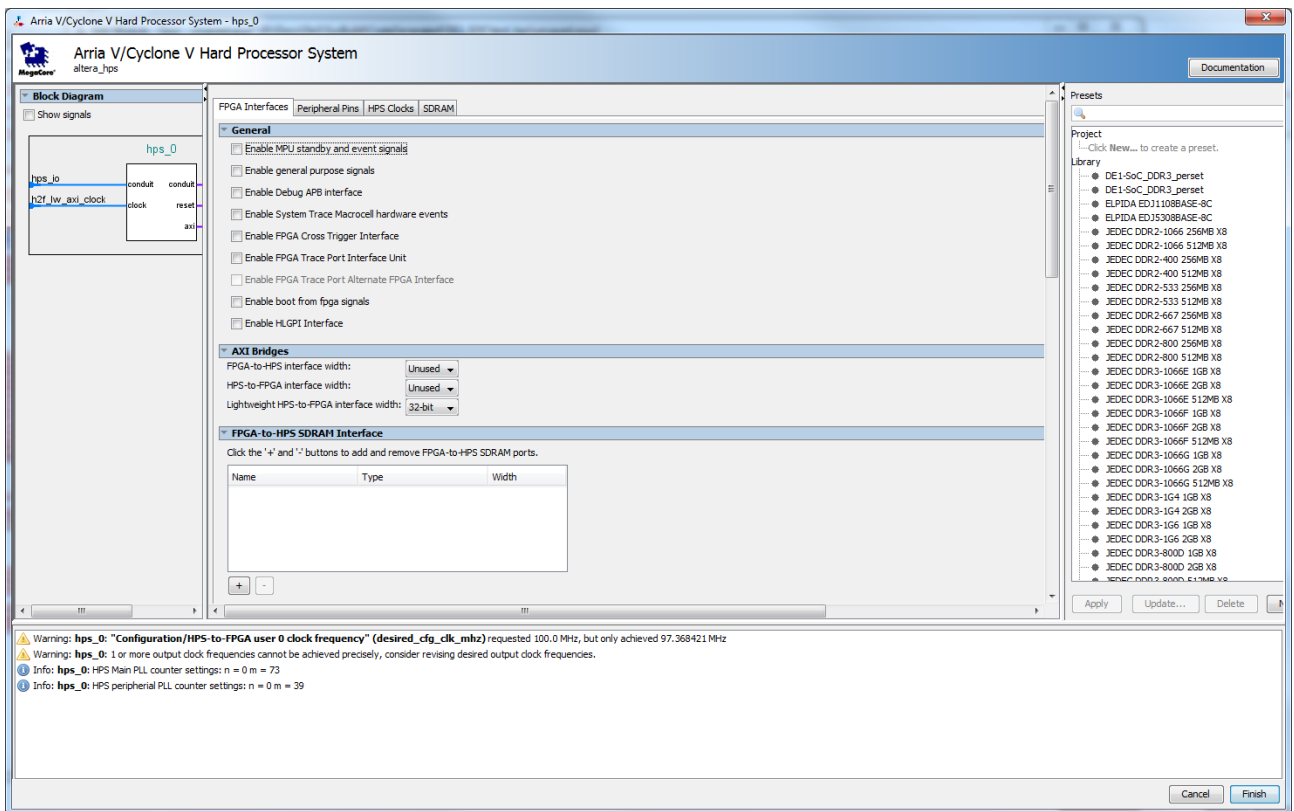


Рисунок 3.3 – Інтерфейс системи

У вкладці Peripheral Pins оберемо SD/MMC (з картки завантажується програма) і UART (рисунок 3.4).

У вкладці HPS clocks залишається все за замовчуванням.

У вкладці SDRAM необхідно заповнити всі необхідні поля значеннями skew тощо.

Вони повністю залежать від трасування ліній від залучених мікросхеми до SDRAM.

В проєкті було використано дані для DE1-SoC, тому було взято їх з готового проєкту для моєї плати (це був SOC-Computer у прикладах Altera University Program).

Далі було збережено установки як персет, щоб більше не заповнювати їх.

Тепер можна бачити його в правій колонці.

Далі в IP cores знайдемо PIO, це будуть світлодіоди та кнопки (рисунок 3.5).

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

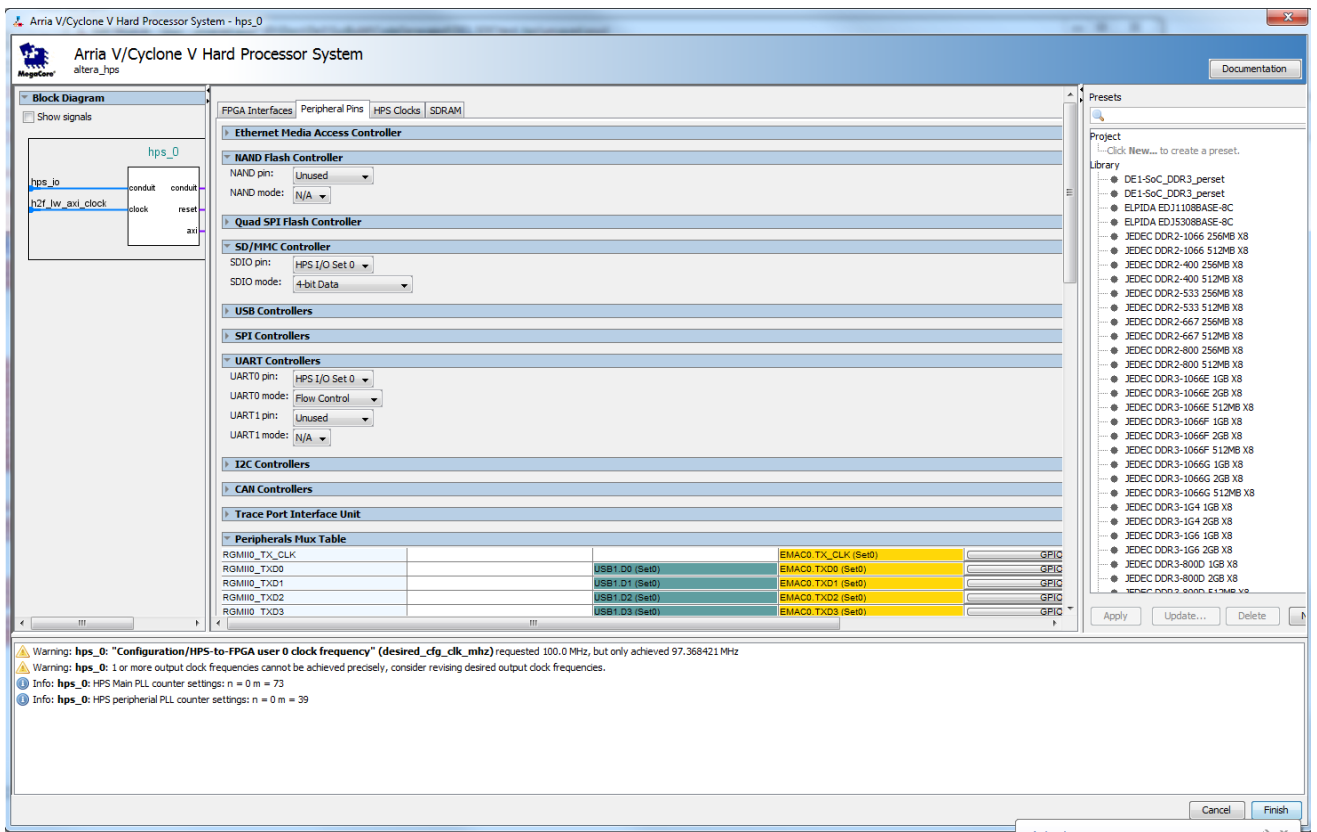


Рисунок 3.4 - Рисунок 3.3 – Інтерфейс налагодженої системи

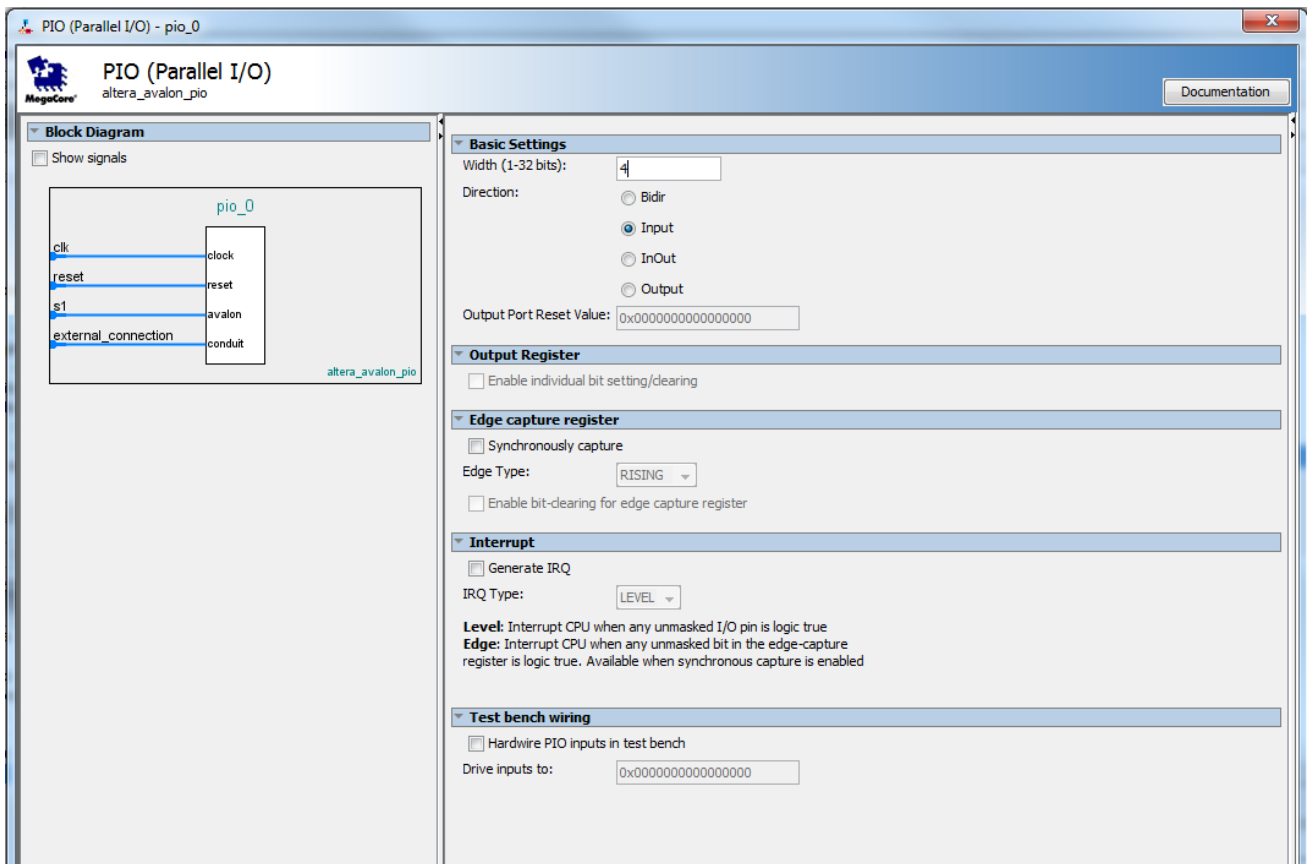


Рисунок 3.5 – Використання світлодіодів та кнопок

Для налаштувань кнопок було обрано виберемо Input і ширину 4 біти так як 4 кнопки (рисунок 3.6).

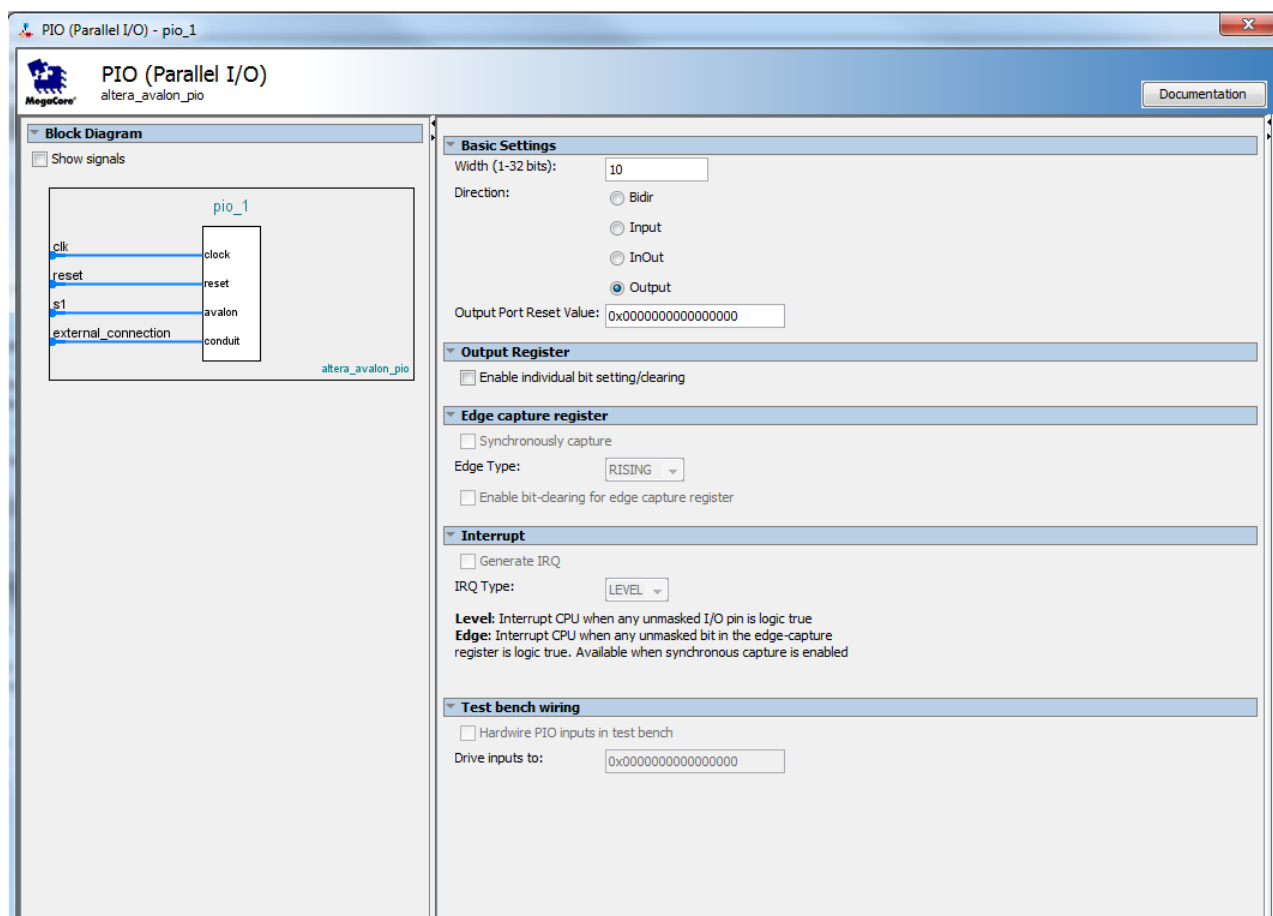


Рисунок 3.6 – Налаштування кнопок Input

Для світлодіодів відповідно Output та ширина 10.

З'єднуємо отриману систему, як показано на рисунку 3.7)

Тепер можемо змінити імена PIO, щоб вигляд був зрозумілішим і нагляднішим.

LWH2F bridge є сполучною ланкою між HPS і FPGA (так само призначає майстром HPS).

PIO – це елементи, що виконуються в «тканині» FPGA.

Якщо клікнути двічі на напис «Double click for export» навпроти external connection, щоб вивести з системи висновки PIO світлодіодів і кнопок.

Пізніше стає видно, що у коді top-level файлу генерованої системи ці ВИВОДИ.

Зм..	Арк.	№докум.	Підпис	Дата

Оскільки доступ до PIO, та й до всіх інших ядер, HPS здійснюється за адресами, необхідно призначити їх.

Це можна зробити в системі автоматично, якщо натиснути Assign base addresses.

Процес роботи з assign base addresses подано на рисунку 3.7.

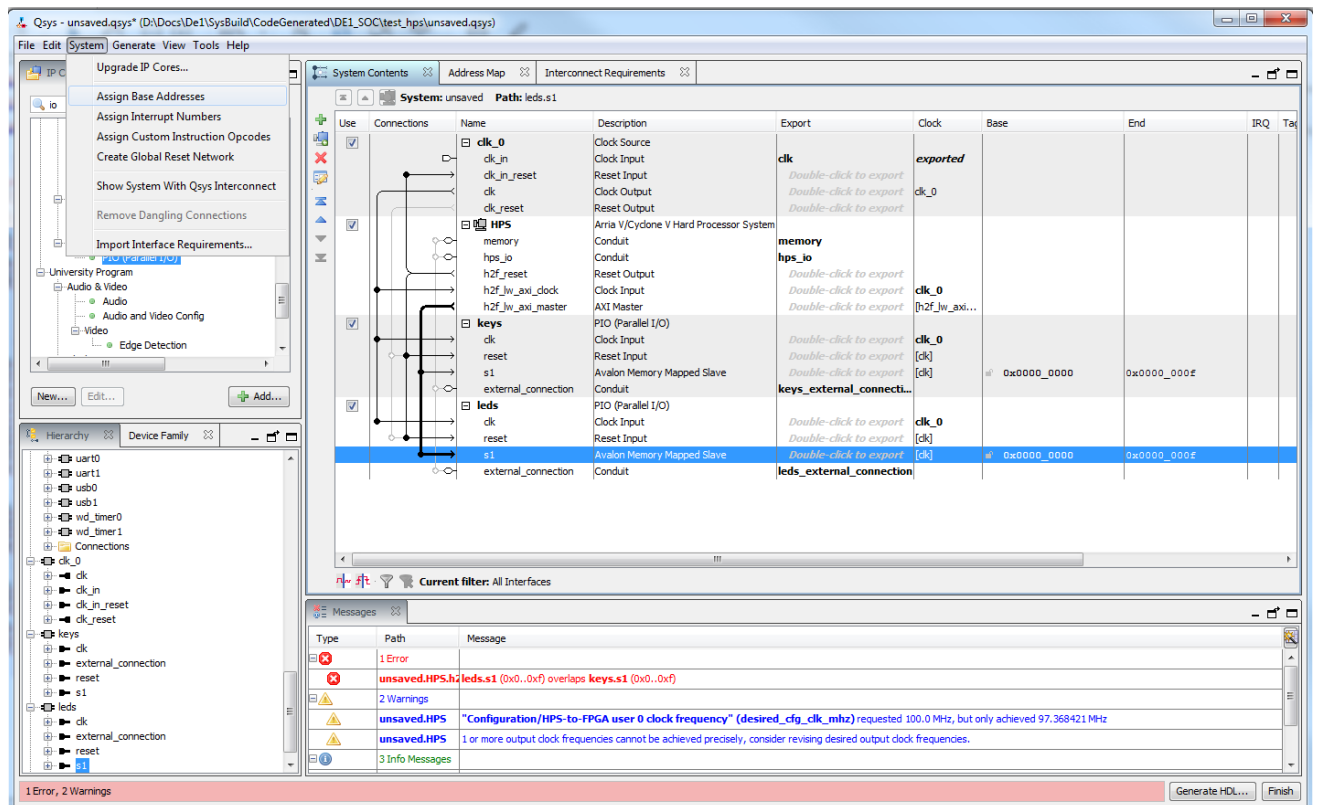


Рисунок 3.6 – Налаштування кнопок Output

Після цього можна генерувати код системи (рисунок 3.8).

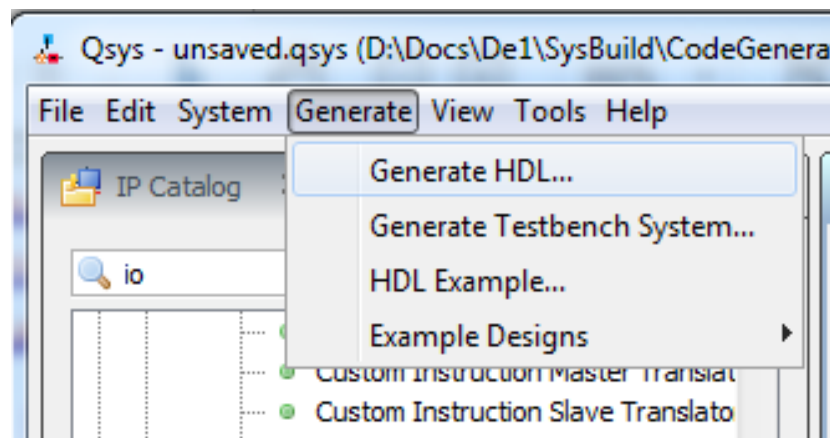


Рисунок 3.7 – Процес роботи з assign base addresses

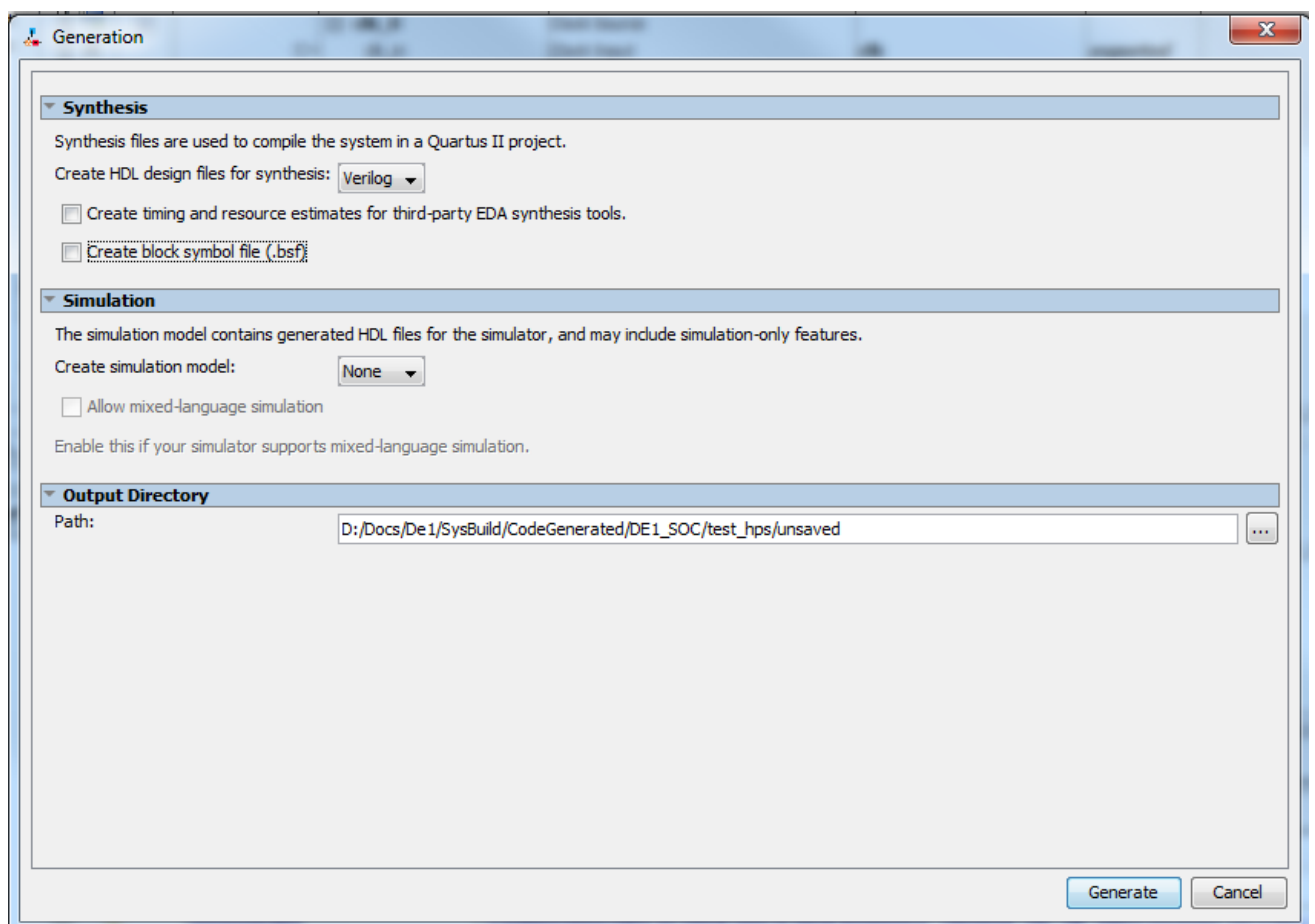


Рисунок 3.8 – Процес генерації коду

Вказуємо шлях та бажану мову коду Verilog.

Після цього QSYS можна закрити. У вікні Quartus з'являється таке повідомлення, як показано на рисунку 3.9.

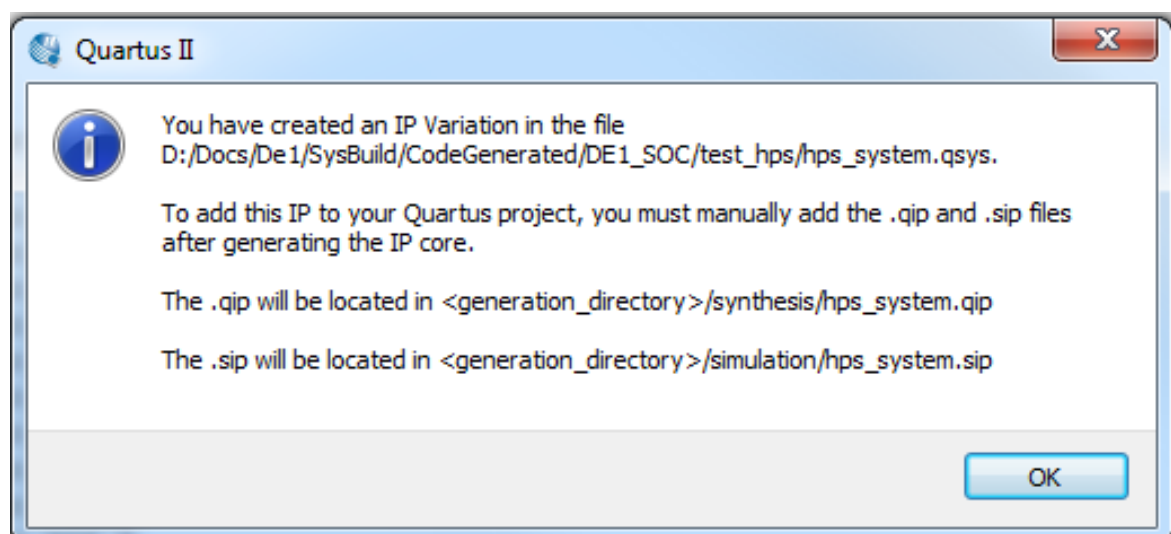


Рисунок 3.9 – Повідомлення у вікні Quartus

Після погодження запиту, можна перейти до корегування налаштувань системи.

Інтерфейс налаштування системи подано на рисунку 3.10.

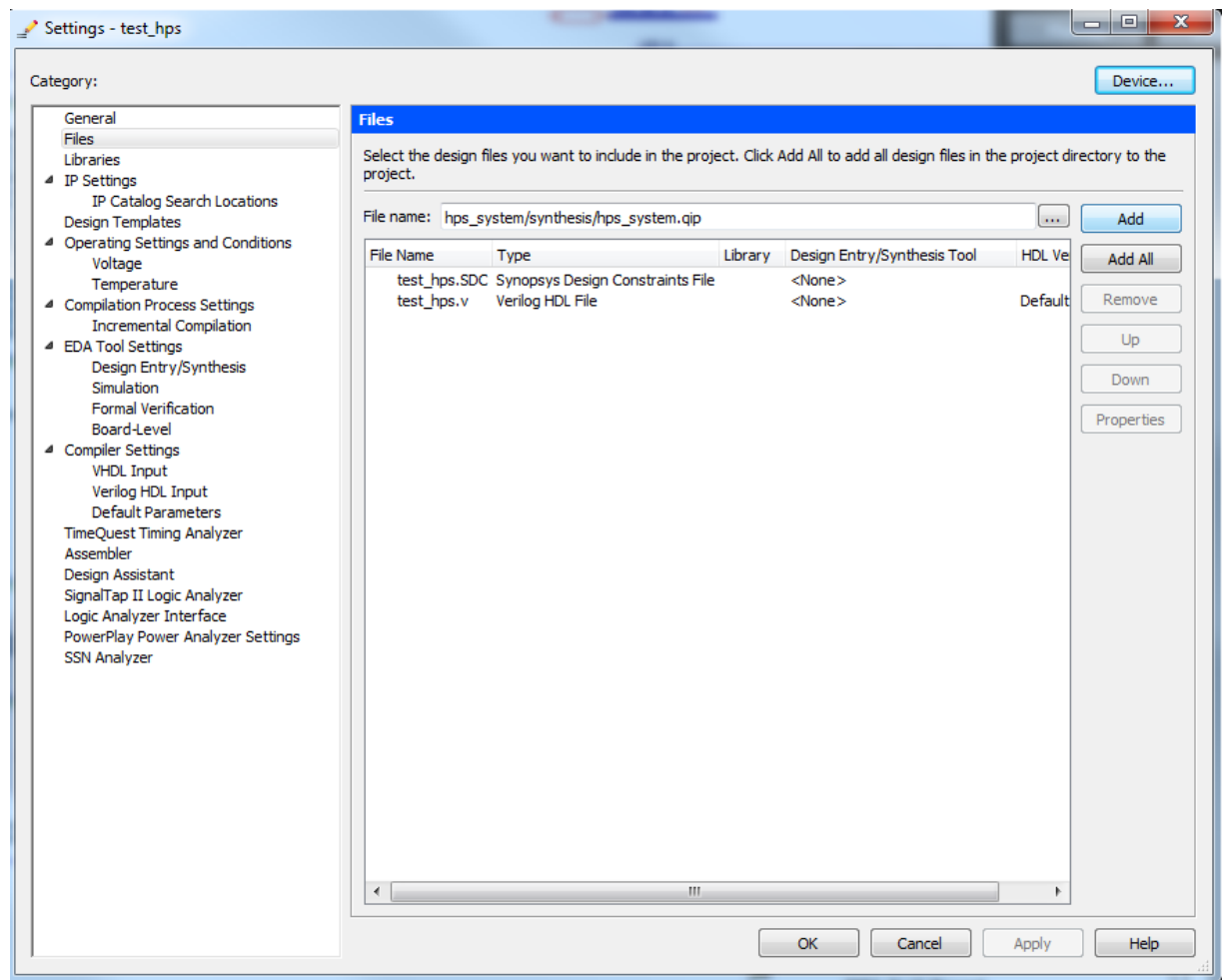


Рисунок 3.10 – Інтерфейс налаштування системи

У вікні Files бачимо hps_system.qip.

Якщо його розкрити, то побачимо top-level файл системи (рисунок 3.11).

top-level файл системи подано на рисунку 3.11.

Все, що було обрано в системі QSYS тепер у цьому файлі.

Тепер потрібно вставити цей модуль у початковий файл.

Це буде top-level файлом прошивки FPGA.

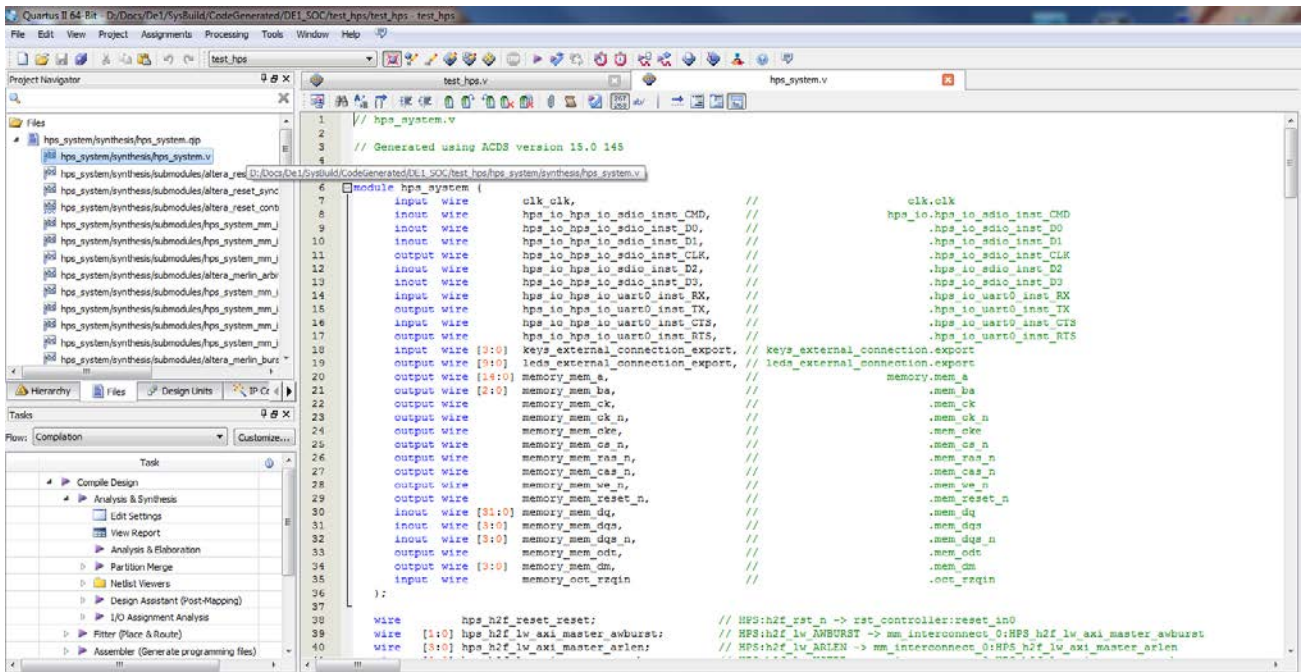


Рисунок 3.11 – top-level файл системи

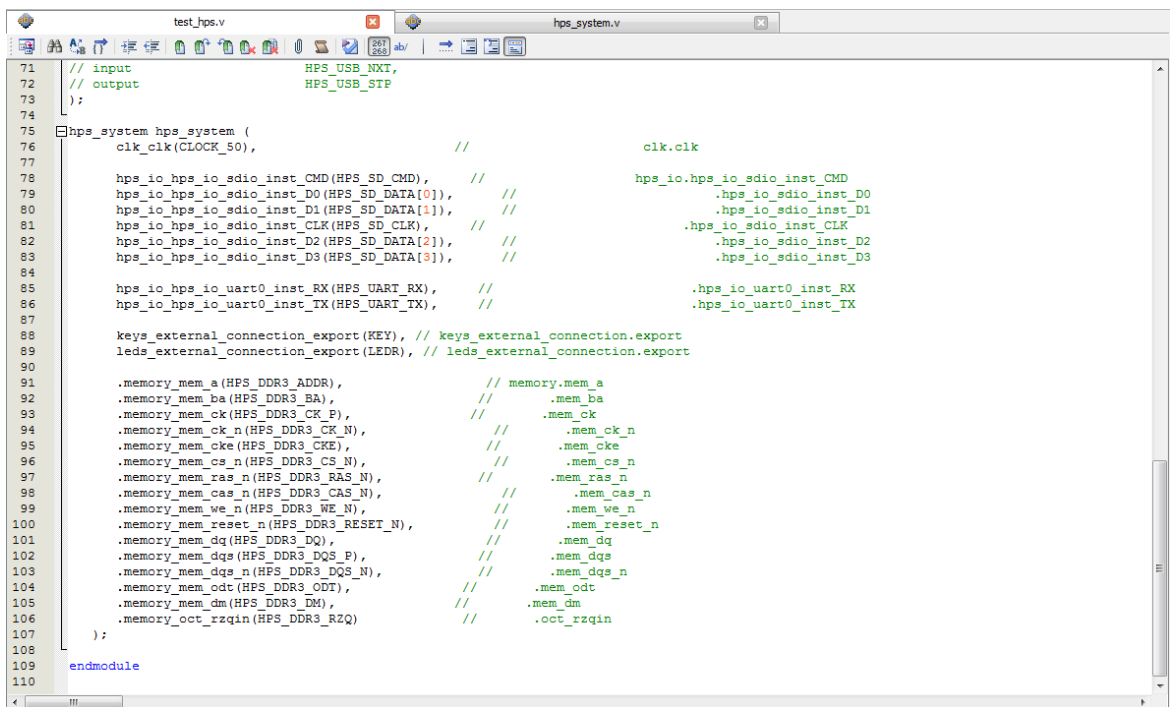


Рисунок 3.11 – top-level файл налаштованої системи

У налаштованому top-level файл приписуємо виводів HPS системи початкові I/O піни цього файлу.

Важливо вказати, що нічого не потрібно призначати в Pin Planner, все вже зроблено під час створення проекту утилітою.

Але необхідно призначити піни HPS, це робиться так, як показано на рисунках 3.12, 3.13.

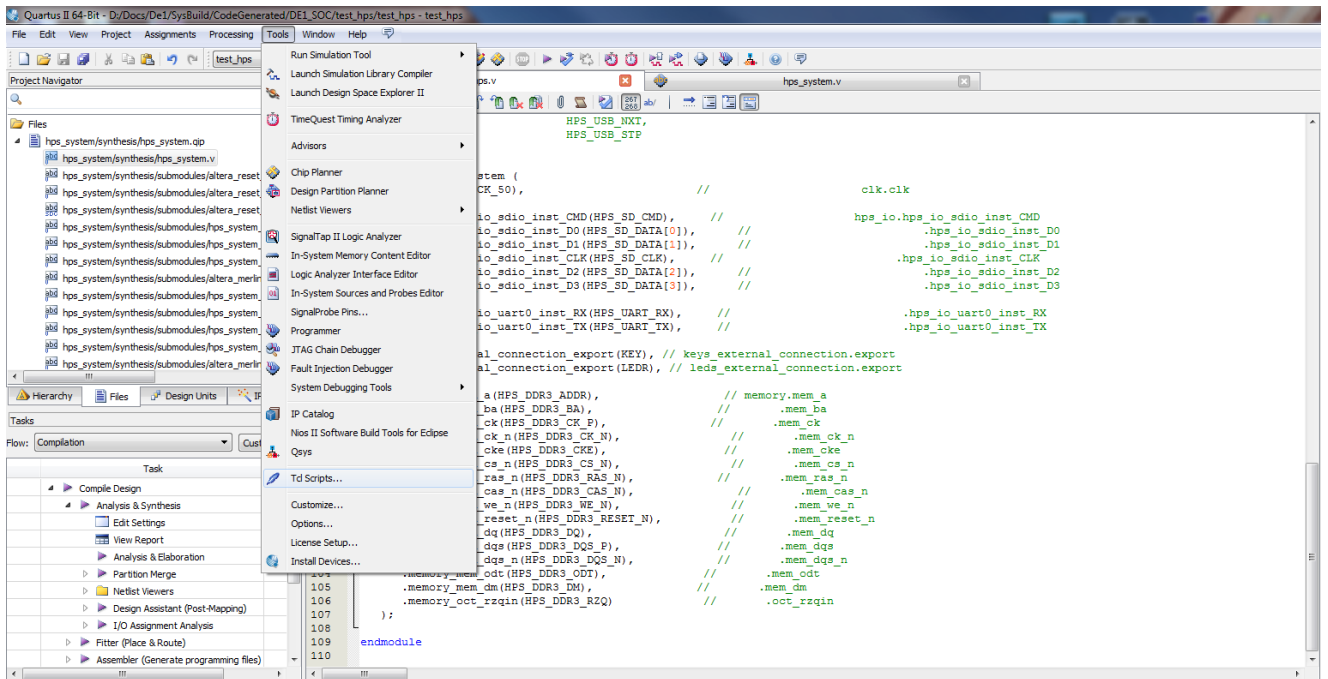


Рисунок 3.12 – Призначення пінів HPS

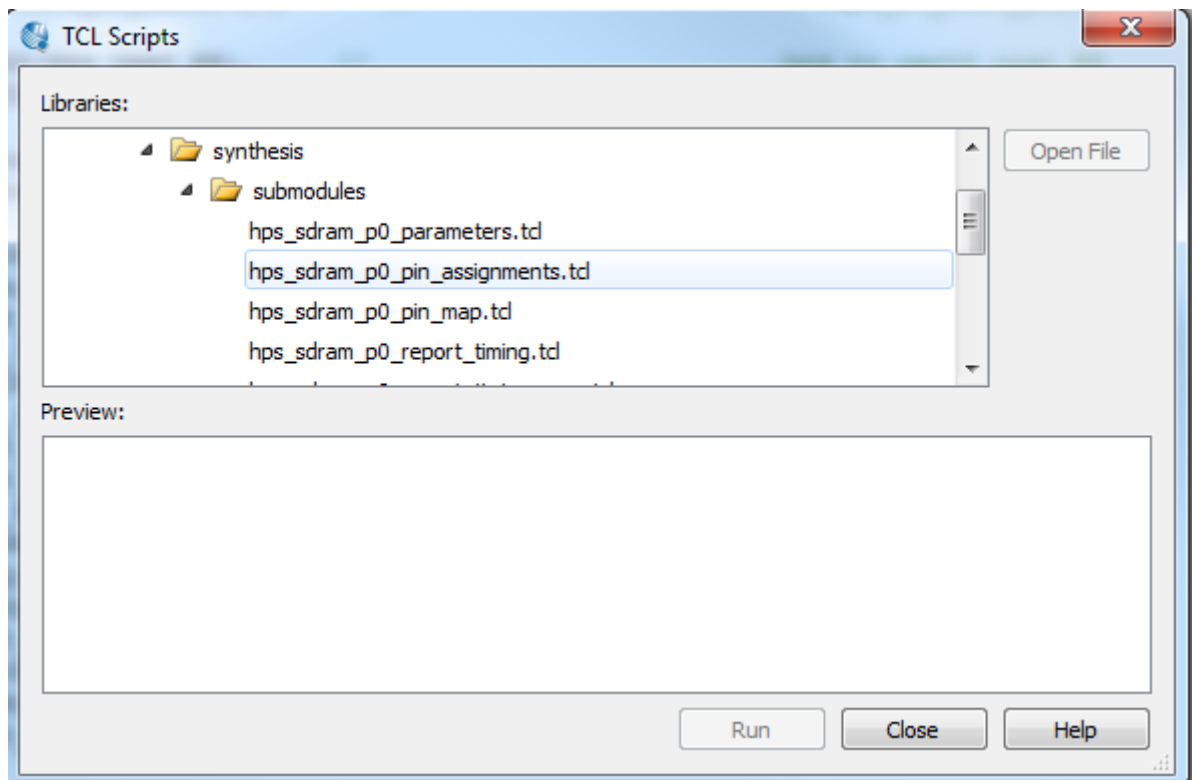


Рисунок 3.13 – Призначення пінів HPS засобами бібліотеки

Як тільки це зроблено, можна компілювати прошивку.

Важливо зауважити, що якщо були допущені будь-які помилки (наприклад, можливим є встановлення крапку в оголошенні hps_system біля clk_clk та hps_io_hps_io...).

В такій ситуації необхідно заново виконувати tcl скрипти.

Навіть якщо все написано правильно, і ви запустили скрипти tcl, але після запуску компіляції ви отримуєте помилку, варто спробувати ще раз запустити скрипти, нічого не змінюючи в коді.

Налаштування завдання подано на рисунку 3.14.

	Task	Time
✓	▶ Compile Design	00:04:52
✓	▶ Analysis & Synthesis	00:02:59
	Edit Settings	
	View Report	
✓	▶ Analysis & Elaboration	
	▶ Partition Merge	
	▶ Netlist Viewers	
	▶ Design Assistant (Post-Mapping)	
	▶ I/O Assignment Analysis	
✓	▶ Fitter (Place & Route)	00:01:06
✓	▶ Assembler (Generate programming files)	00:00:16
✓	▶ TimeQuest Timing Analysis	00:00:31
	▶ EDA Netlist Writer	
	Edit Settings	
	View Report	
	Program Device (Open Programmer)	

Рисунок 3.14 – Налаштування завдання

3.2 Створення Preloader

Процес завантаження HPS включає кілька етапів.

Першим кроком є врахування архітектури ПЛІС.

Варто зауважити, що Cortex-A9 є процесором для додатків, літера «А» в назві означає Applicationю

В першу чергу призначений для роботи з ОС, наприклад Linux.

Тому ідея запуску Bare-Metal програм може є аргументованою.

Відразу після увімкнення виконується код розташований прямо на Flash пам'яті Cortex-A9 званий BootRom.

Тепер можна змінити його або переглянути його зміст.

Він служить для первинної ініціалізації і на наступному етапі передає увесь процес завантаження в SSBL (Second Stage Boot Loader, який називається Preloader).

Код BootRom вибирає джерело завантаження Preloader, орієнтуюсь на зовнішні фізичні піни BSEL.

У DE1-SoC спочатку обрана конфігурація пінів для завантаження з картки SD.

Важливо також розуміти, що зміна на, наприклад, на QSPI або NAND flash, не припаюючи додатково перемикача і пари резисторів, неможливо.

Тому QSYS було обрано у вкладці Peripheral Pins піни SD карти.

Є також варіант завантаження не з зовнішніх джерел, а з пам'яті, створеної в FPGA, з попередньо завантаженим туди кодом.

В роботі було обрано варіант, коли після виконання коду BootRom починає завантажуватися Preloader, необхідний для налаштування Clock, SDRAM та іншого.

Після цього починає виконуватися програма.

Для створення Preloader потрібно SoC EDS, який необхідно завантажити його із сайту Intel FPGA.

Інтерфейс командного рядка подано на ринку 3.15.

					КвРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		51

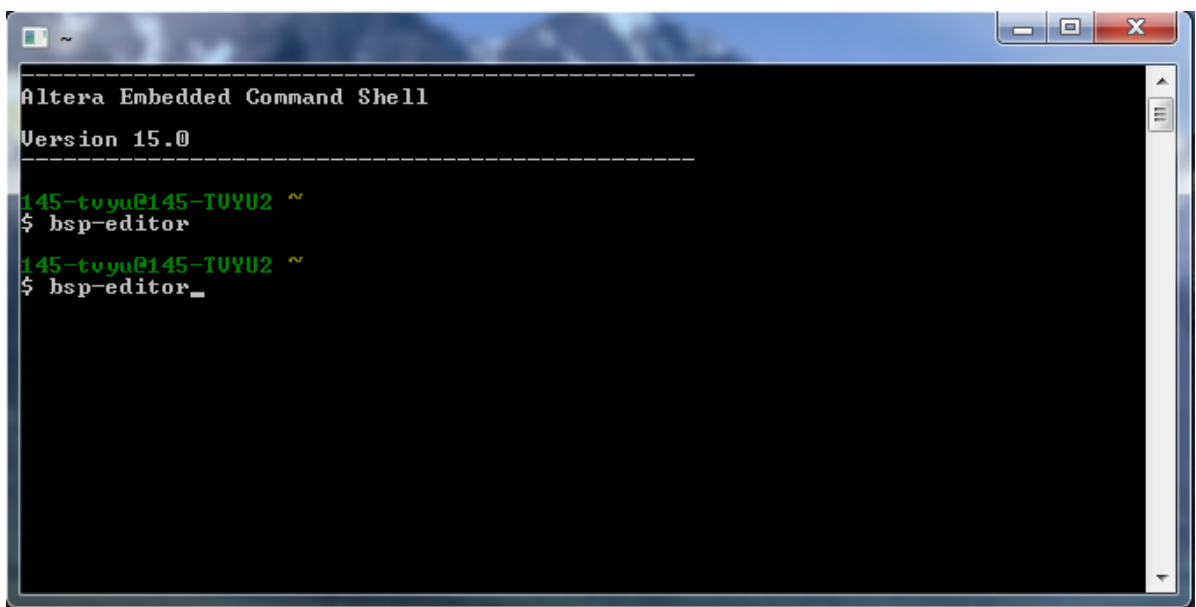


Рисунок 3.15 – Інтерфейс командного рядка

Програма працює із командного рядка.

Для початку було створено BSP. Для цього було написано відповідну команду «bsp-editor».

У цьому вікні треба було натиснути New HPS BSP.

Інтерфейс створення BSP подано на рисунках 3.16, 3.17.

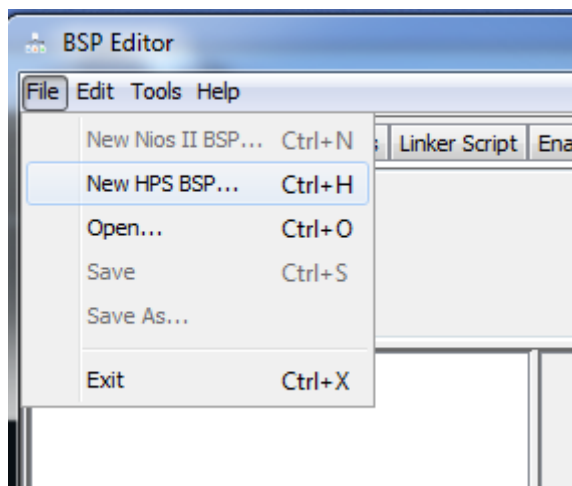


Рисунок 3.16 – Інтерфейс створення BSP

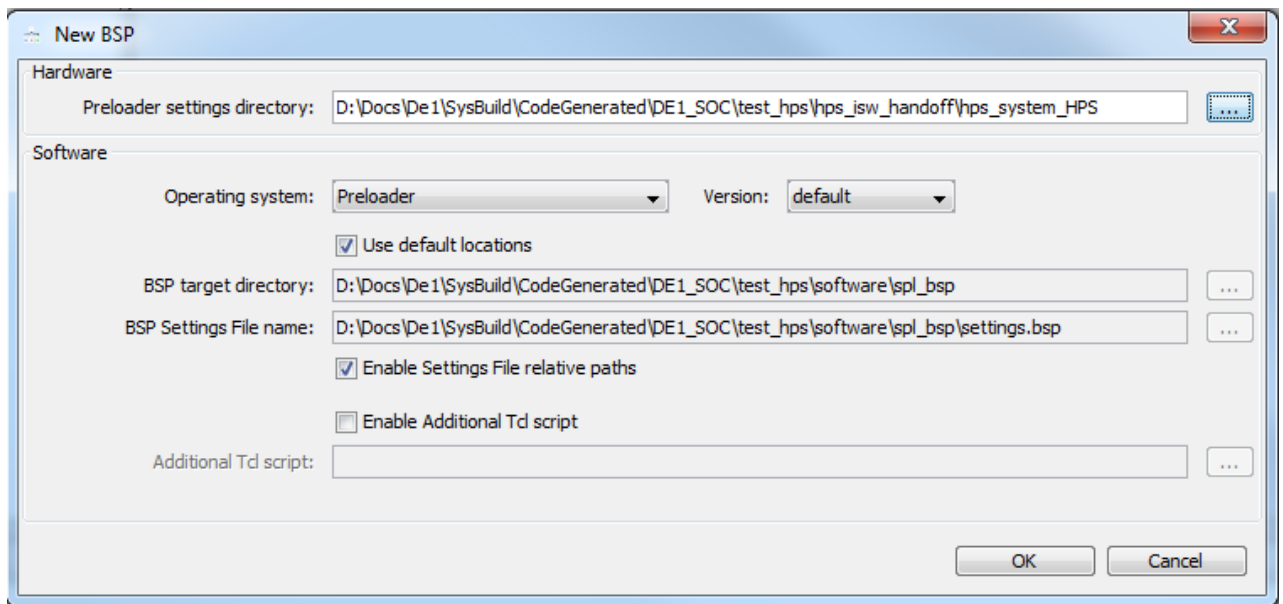


Рисунок 3.17 – Інтерфейс створення BSP через візард

Далі необхідно вказати шлях до поточного проєкту розробки системи BSP {Project directory}/hps_isw_handoff/.

Після натискання ОК, інші параметри в розроблюваній системі змінювати не потрібно.

Налаштування параметрів подано на рисунку 3.18.

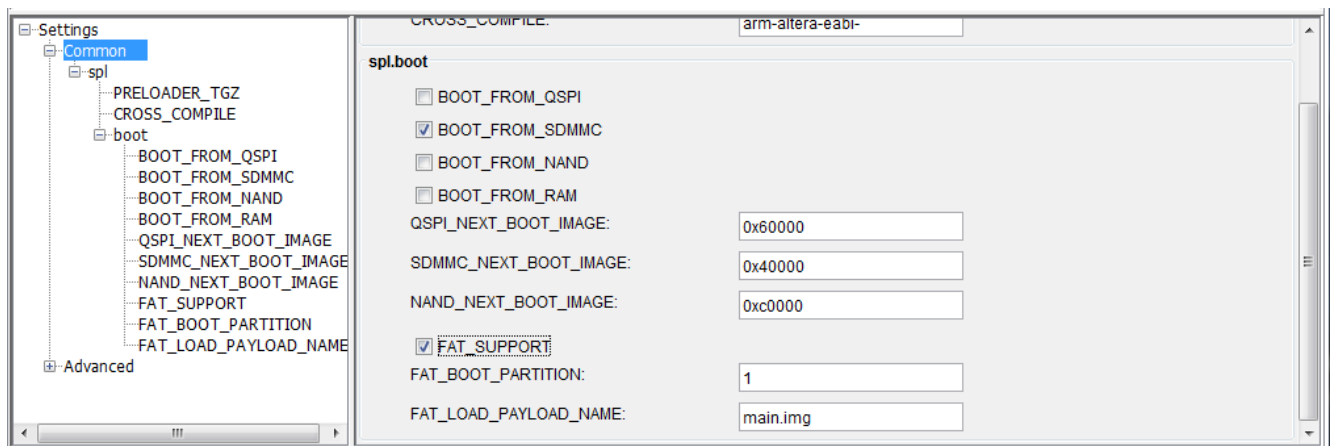


Рисунок 3.18 – Налаштування параметрів подано

Вибираємо налаштування spl.boot в якому вказуємо джерело.

Звідти буде завантажуватися Preloader.

У кваліфікаційній роботі в проєкті було обрано картку SD.

Далі було обрано параметр BOOT_FROM_SDMMC.

Було використано варіант завантаження Preloader та розробленої програми.

При цьому на флеш-накопичувачі має бути як мінімум 2 розділи:

- 1) розділ не відформатований з id=A2, для Preloader;
- 2) розділ із системою FAT32, для програми.

Існує інший варіант розмітки, який не передбачає поділу флеш-накопичувачі на розділи.

Мова йде так званий формат RAW, але в проєкті такої необхідно не було.

Відформатувати таким чином флеш-накопичувач можна будь-якою призначеним для цього програмним забезпеченням.

Також можна використовувати вже зібраний образ, який розміщено у системному каталозі
...\\embedded\\embeddedsw\\socfpga\\prebuilt_images\\sd_card_linux_boot_image.tar.gz і записати його на флешку через Win32DiskImager.

Далі було обрано параметри:

- 1) FAT_SUPPORT;
- 2) FAT_BOOT_PARTITION 1;
- 3) FAT_LOAD_PAYLOAD_NAME .img.

Також важливо не використовувати параметри:

- 1) WATCHDOG_ENABLE;
- 2) EXE_ON_FPGA.

Це пояснюється тим, що в проєкті завантаження наявного Preloader відбувається не з FPGA.

Налаштування параметрів подано на рисунку 3.19.

Serial Support означає, що Preloader буде використовувати UART модуль для виведення діагностичних повідомлень прямо під час завантаження.

Semihosting означає, що під час виведення цих діагностичних повідомлень, вони будуть автоматично виводитись у вікні debugger при системному налагодженні.

Використання в самій програмі цієї функції дуже зручне, вона дозволяє виводити у вікно debugger все, що написано у функції printf без додаткового написання коду.

Якщо в налаштуваннях QSYS у HPS не вказати використання UART, але поставити галочку Serial Support у BSP – такий використовуваний Preloader працювати не буде.

Якщо прибрати Serial Support, залишивши Semihosting, також система працювати не буде.

Далі натискаємо Generate, а потім Exit.

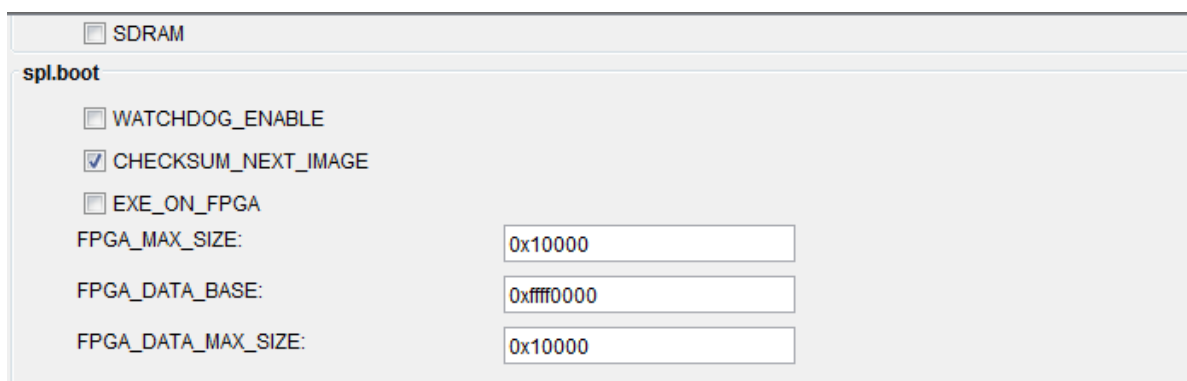


Рисунок 3.19 – Налаштування параметрів подано

Налаштування spl.debug подано на рисунку 3.20.

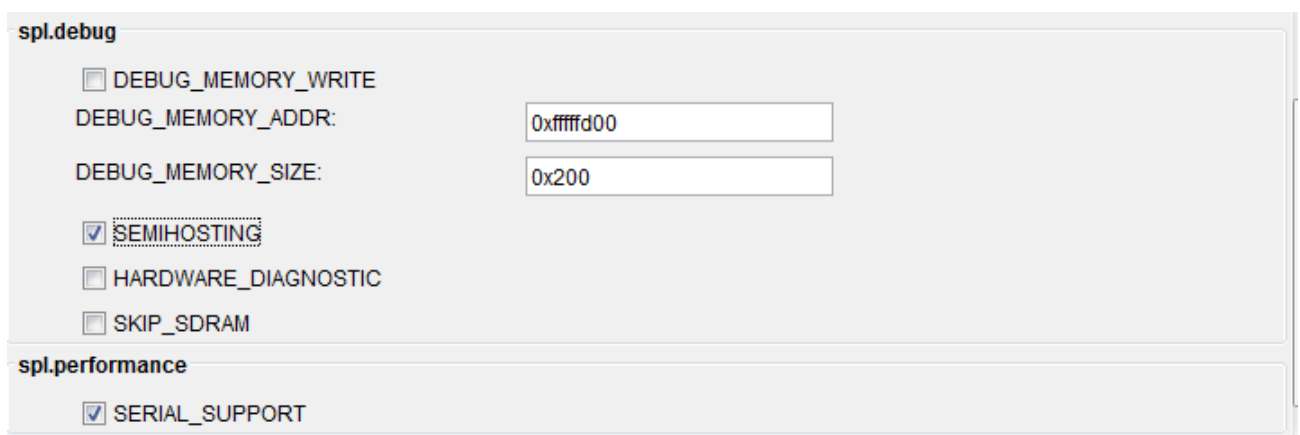


Рисунок 3.20 – Налаштування spl.debug

Тепер необхідно поміняти робочу папку SoC EDS засобами використання команди:

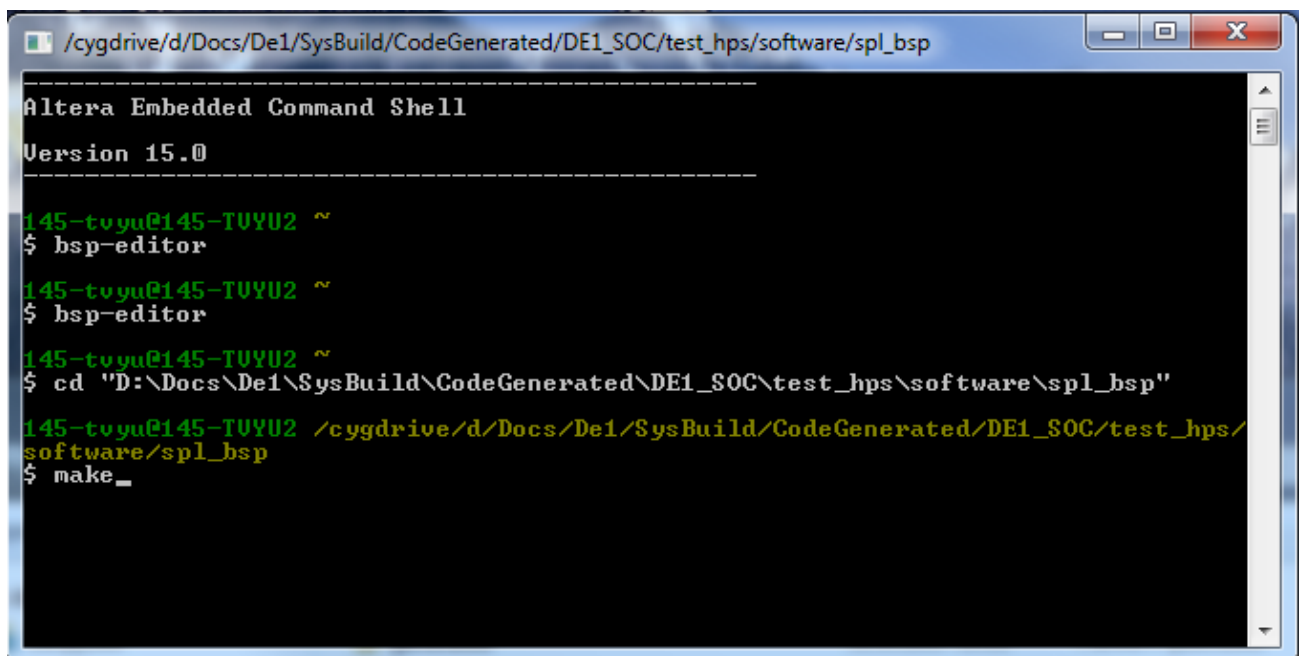
```
cd "<вказується повний шлях до генерованих файлів BSP (за дефолтом це ...software/spl-bsp)>"
```

Для збірки Preloader виконається відповідна команда:

```
make
```

Наприкінці процесу в пакеті було отримано необхідний файл preloader-mkprimage.bin.

Процес виконання процедури збірки Preloader представлено на рисунку 3.21.



```
Altera Embedded Command Shell
Version 15.0
-----
145-tvyu@145-TUYU2 ~
$ hsp-editor
145-tvyu@145-TUYU2 ~
$ hsp-editor
145-tvyu@145-TUYU2 ~
$ cd "D:\Docs\De1\SysBuild\CodeGenerated\DE1_SOC\test_hps\software\spl_bsp"
145-tvyu@145-TUYU2 /cygdrive/d/Docs/De1/SysBuild/CodeGenerated/DE1_SOC/test_hps/software/spl_bsp
$ make_
```

Рисунок 3.21 – Процес виконання процедури збірки Preloader

Це зібраний системний файл, в якому знаходяться однакові чотири образи Preloader.

Щоб виконати в SoC EDS розбір цього файлу на складові (окремі образи), а потім зібрати з інших конфігурацій (з різними образами Preloader) необхідно виконати команду:

```
mkprimage
```

Інші конфігурації системи можуть бути абсолютно різними (різні системи QSYS).

Тобто в такому набору було отримано окремі файли preloader-mkprimage.bin з 4 однаковими образами (по 64кб кожен).

Після розбору їх на складові було зібрано новий preloader-mkprimage.bin з різними образами (наприклад, із різними джерелами завантаження).

Це було здійснено для надійності проєктованої системи.

Наприклад, можливою є ситуація, коли завантаження системи з першої спроби з першого образу було невдалим.

Тоді автоматично починається завантаження другого образу, а він, наприклад, обов'язково має відмінності для аварійної ситуації.

Якщо цей не вдалося завантажити, то переходимо до третього і так далі.

Наступним кроком необхідним є встановлення флеш-накопичувача з підготовленим розділом A2.

Далі було записано на флеш-накопичувач зібраний preloader-mkprimage.bin за допомогою команди:

```
«alt-boot-disk-util -p -a write -d »
```

SoC EDS повинен вказувати шлях до папки де розташовано зібраний файл Preloader.

Підготовчий процес для написання власного програмного забезпечення стає завершеним.

Для цього необхідним було створення header файлів з визначеннями елементів QSYS.

Тоді було замінено шлях SoC EDS, вказавши файл прошивки, і виконано команду:

```
sopc-create-header-files .sopcinfo.
```

в результаті було отримано файли, що використовуються для написання власного ПЗ.

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		57

3.3 Розроблення програмного забезпечення

Для написання та налагодження програми було використано середовище DS-5 Eclipse.

Таким чином, було запущено Eclipse засобами SoC EDS системної команди "eclipse&".

Знак "&" в кінці команди ставиться для того, щоб вікно SoC EDS було активним після відкриття Eclipse.

Далі було створено порожній "C" проект.

Далі було обрано використовуваний компілятор.

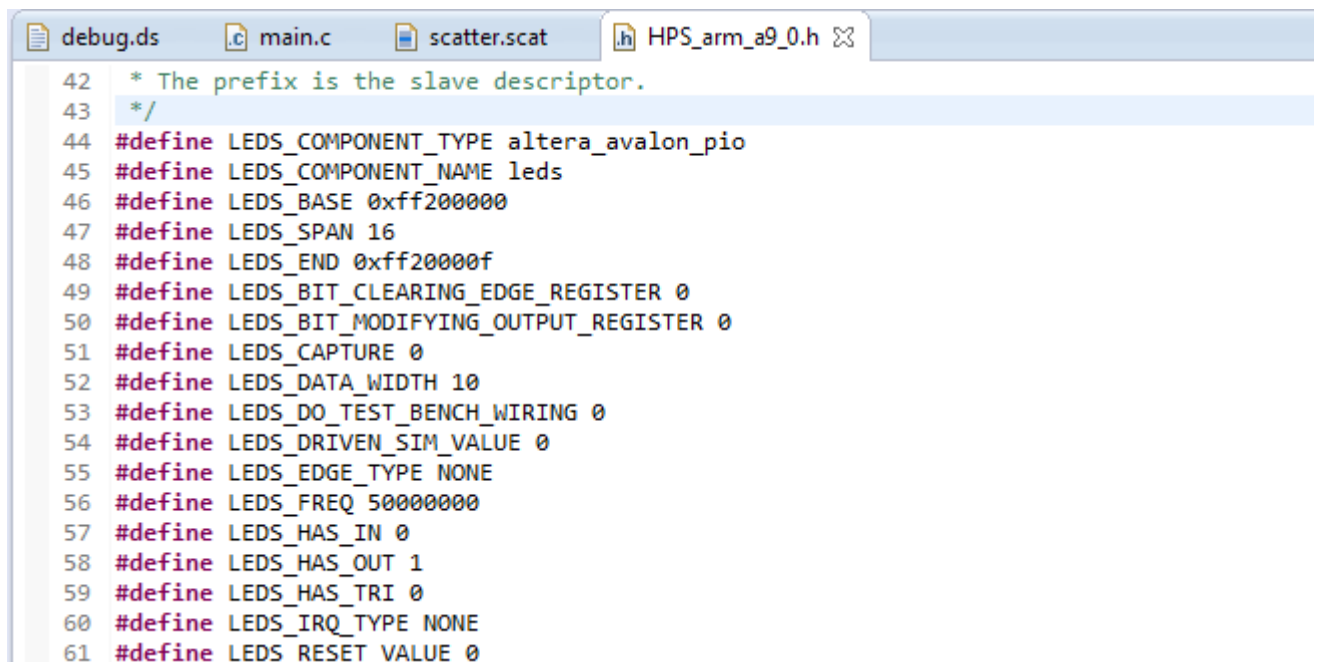
В проєкті було застосовано Arm Compiler 5.

Він вирізняється простотою синтаксису scatter файлів.

Компілятор використовує для розміщення написаної програми в різних частинах програми.

Також компілятор базується на використанні linker script GCC.

HPS_arm файл подано на рисунку 3.22.



```
debug.ds  main.c  scatter.scats  HPS_arm_a9_0.h  X
42  * The prefix is the slave descriptor.
43  */
44  #define LEDS_COMPONENT_TYPE altera_avalon_pio
45  #define LEDS_COMPONENT_NAME leds
46  #define LEDS_BASE 0xff200000
47  #define LEDS_SPAN 16
48  #define LEDS_END 0xff20000f
49  #define LEDS_BIT_CLEARING_EDGE_REGISTER 0
50  #define LEDS_BIT_MODIFYING_OUTPUT_REGISTER 0
51  #define LEDS_CAPTURE 0
52  #define LEDS_DATA_WIDTH 10
53  #define LEDS_DO_TEST_BENCH_WIRING 0
54  #define LEDS_DRIVEN_SIM_VALUE 0
55  #define LEDS_EDGE_TYPE NONE
56  #define LEDS_FREQ 50000000
57  #define LEDS_HAS_IN 0
58  #define LEDS_HAS_OUT 1
59  #define LEDS_HAS_TRI 0
60  #define LEDS_IRQ_TYPE NONE
61  #define LEDS_RESET_VALUE 0
```

a)

```

62
63 /*
64  * Macros for device 'keys', class 'altera_avalon_pio'
65  * The macros are prefixed with 'KEYS_'.
66  * The prefix is the slave descriptor.
67  */
68 #define KEYS_COMPONENT_TYPE altera_avalon_pio
69 #define KEYS_COMPONENT_NAME keys
70 #define KEYS_BASE 0xff200010
71 #define KEYS_SPAN 16
72 #define KEYS_END 0xff20001f
73 #define KEYS_BIT_CLEARING_EDGE_REGISTER 0
74 #define KEYS_BIT_MODIFYING_OUTPUT_REGISTER 0
75 #define KEYS_CAPTURE 0
76 #define KEYS_DATA_WIDTH 4
77 #define KEYS_DO_TEST_BENCH_WIRING 0
78 #define KEYS_DRIVEN_SIM_VALUE 0
79 #define KEYS_EDGE_TYPE NONE
80 #define KEYS_FREQ 50000000
81 #define KEYS_HAS_IN 1
82 #define KEYS_HAS_OUT 0
83 #define KEYS_HAS_TRI 0
84 #define KEYS_IRQ_TYPE NONE
85 #define KEYS_RESET_VALUE 0
86

```

б)

Рисунок 3.22 – HPS_arm файл

Вказання властивостей проекту подано на рисунку 3.23.

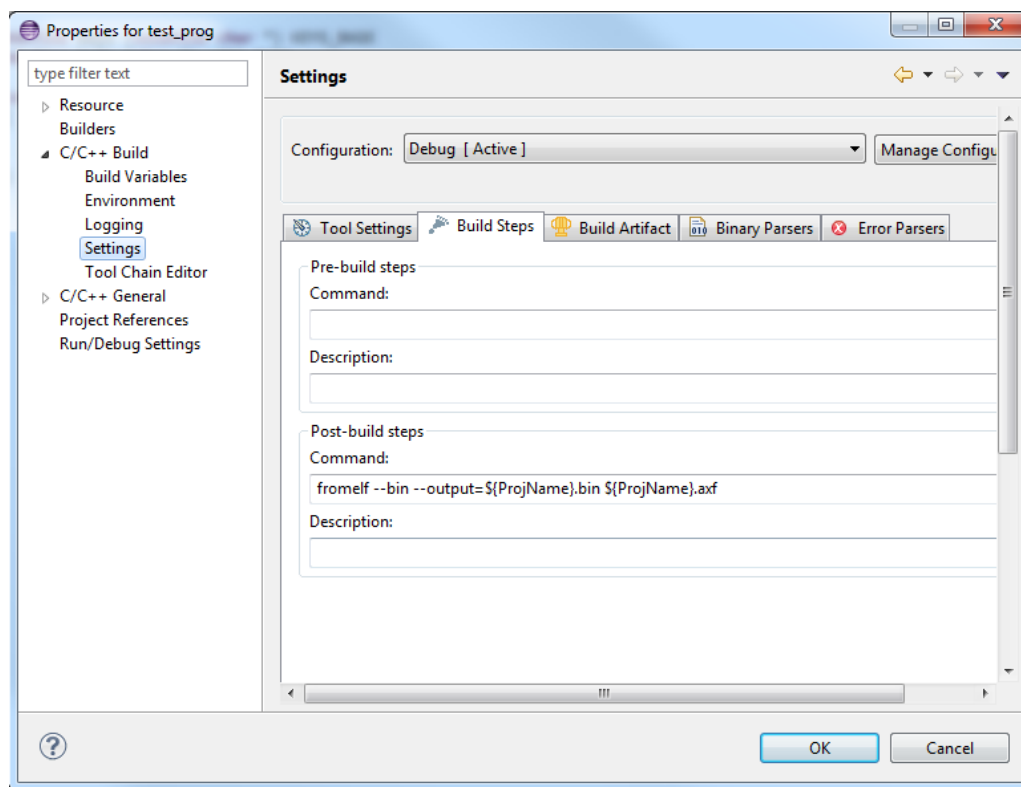


Рисунок 3.23 – Вказання властивостей проекту

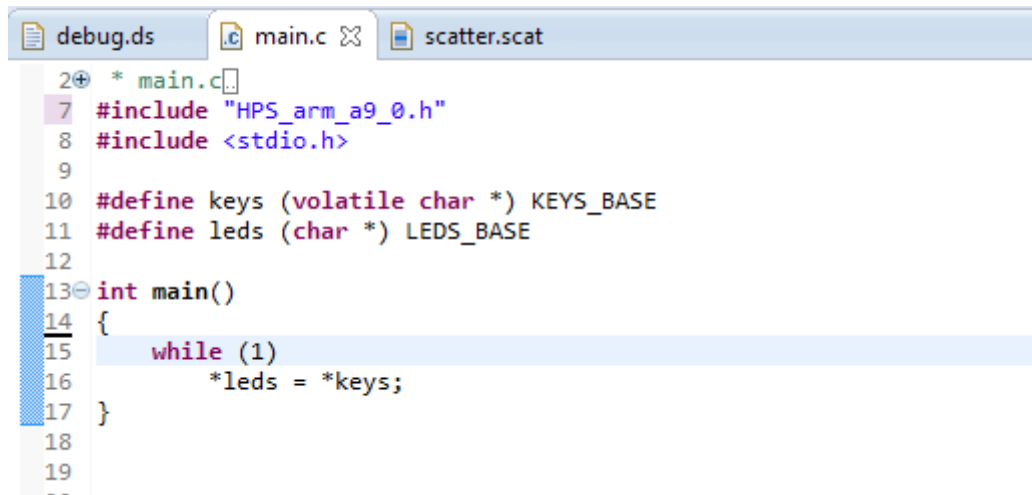
Зм.	Арк.	№докум.	Підпис	Дата

Наступним кроком є необхідність написання коду для забезпечення доступу до пам'яті.

З цією метою було описано код, представлений на рисунку 3.24.

Ця програма надає вміст у пам'яті за адресою KEYS_BASE для адреси LEADS_BASE.

Якщо натиснути кнопку на платі ПЛІС, то світлодіод згасне.



```
debug.ds  main.c  scatter.scad
2+ * main.c
7 #include "HPS_arm_a9_0.h"
8 #include <stdio.h>
9
10 #define keys (volatile char *) KEYS_BASE
11 #define leds (char *) LEADS_BASE
12
13 int main()
14 {
15     while (1)
16         *leds = *keys;
17 }
18
19
20
```

Рисунок 3.24 - Коду для забезпечення доступу до пам'яті за адресою KEYS_BASE для адреси LEADS_BASE

Розглянемо вміст файлу scatter.

Для налагодження в debugger необхідно написати відповідний системний скрипт.

Вадливо пам'ятати, що процес завантаження здійснюється запуском скрипту, який зупиняє виконання Preloader на етапі завантаження програми з джерела та передає це завдання комп'ютерній системі.

Також вадливо врахувати необхідність здійснення прошивки плати перед завантаженням програми та налагодженням.

Вміст файлу scatter подано на рисунку 3.25.

Процес відлагодження подано на рисунку 3.27.

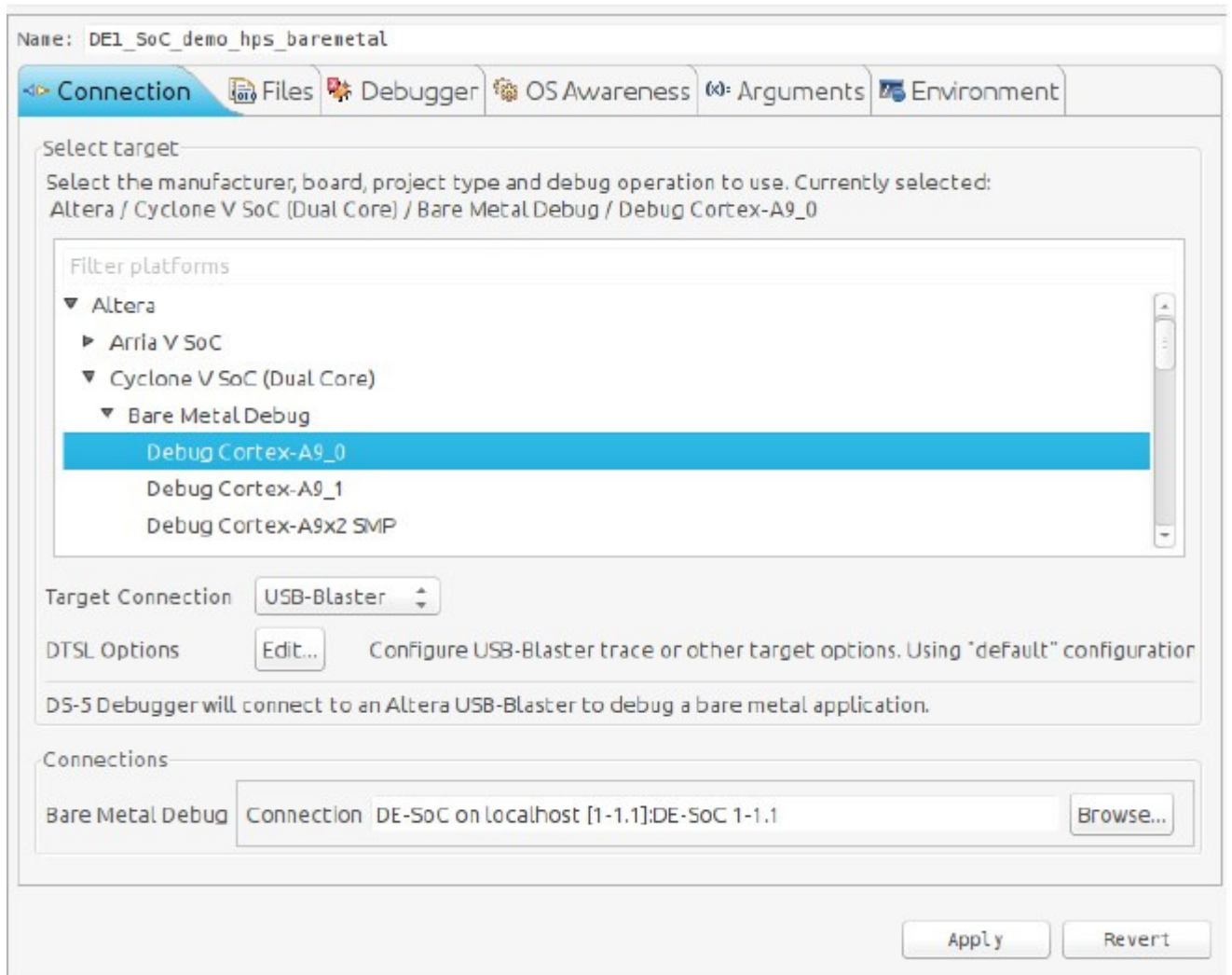


Рисунок 3.27 – Процес відлагодження

Процес налагодження зв'язку подано на рисунку 3.28.

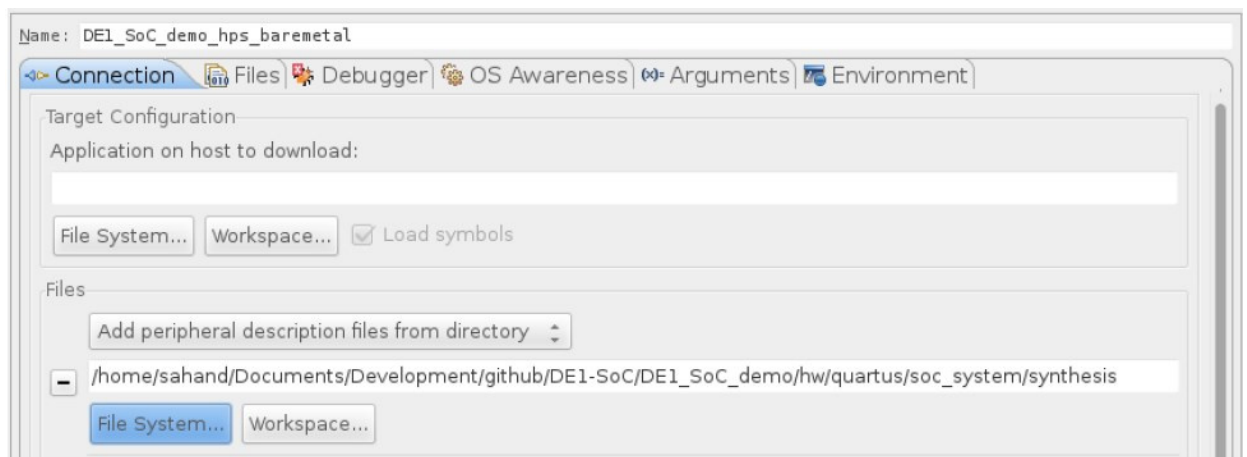


Рисунок 3.28 – Процес налагодження зв'язку

Додаткові налаштування відлагодження подано на рисунку 3.29.

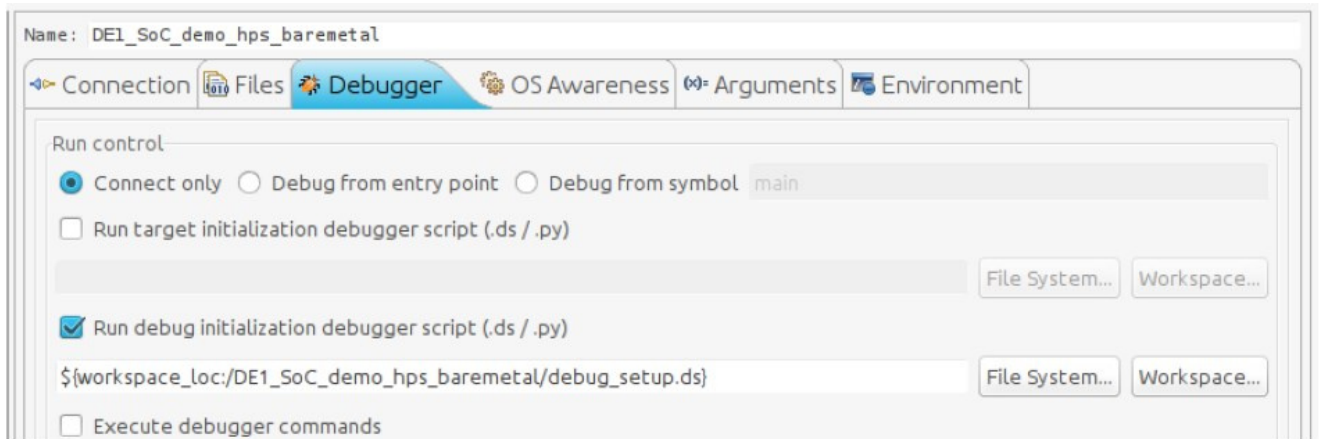


Рисунок 3.29 – Додаткові налаштування відлагодження

На наступному кроці стало можливим здійснення компіляції, налагодження програми, а також перегляд вмісту регістрів.

Вікно компіляції, налагодження програми, а також перегляд вмісту регістрів подано на рисунку 3.30.

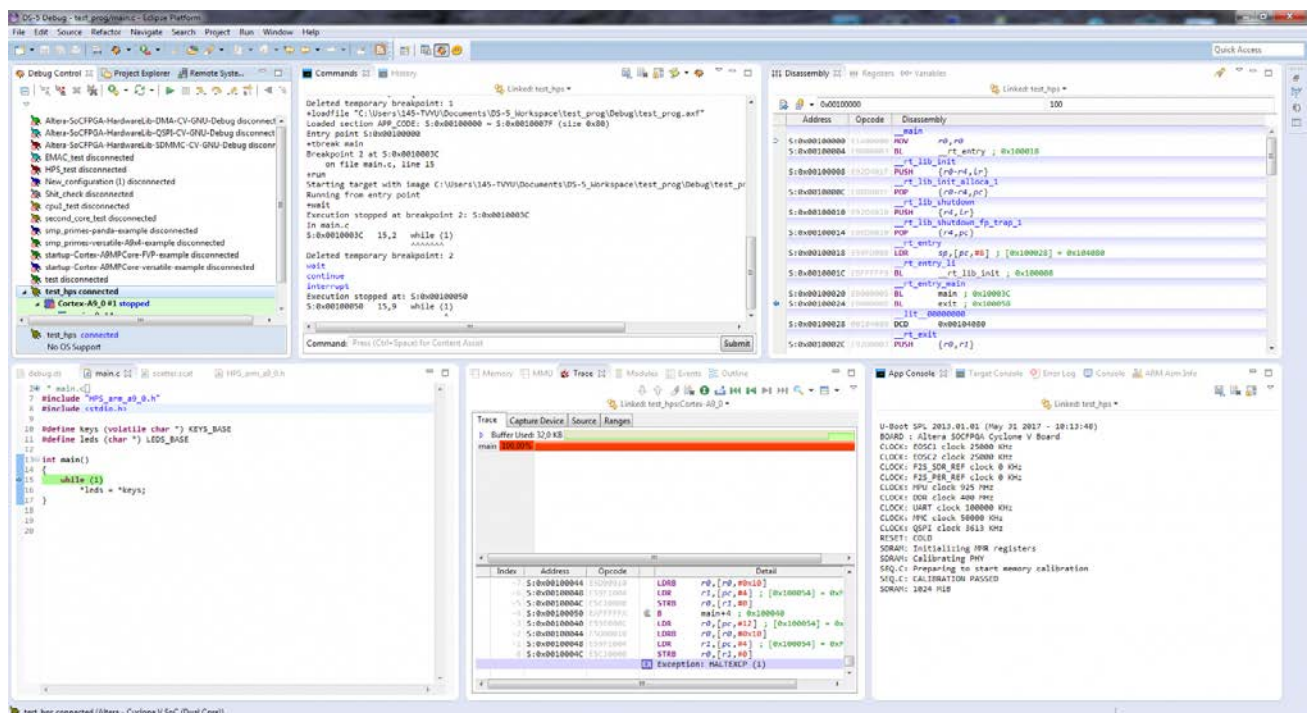


Рисунок 3.30 – Вікно компіляції, налагодження програми, а також перегляд вмісту регістрів

Після того, що програма працює коректно, можна створити образ програми самостійного завантаження.

Для цього з папки Debug з проектом DS-5 було взято файл <prj_name>.bin і через SoC EDS конвертовано його в формат .img.

Це робиться командою:

```
"mkimage -A arm -O u-boot -T standalone -C none -a 0x00100000 -e 0x00100000 -n "baremetal image" -d .bin .img."
```

де "-a" адреса куди завантажувати, а "-e" точка входу програми.

Точка входу також може ставитись у самому проекті, якщо використовуються вектори переривання.

В робот вони не використовуються, тому не це було задано.

Важливо орієнтуватися на те, які адреси було задавано у системному scatter файлі.

Перед виконанням важливо не забути вказати шлях до bin файлу SoC EDS командою

```
cd "<папка з файлом>".
```

Назва img файлу повинна збігатися з тим, що було вказано в editor BSP в FAT_LOAD_PAYLOAD_NAME.

Як результат було скопійовано файл на флеш-накопичувач у fat розділ, як звичайний файл.

Вставивши флеш-накопичувач в DE1-SoC можна побачити виконання розробленої програми.

3.5 Висновки

В розділі представлено процес програмно-апаратна реалізації та тестування програмно-технічного засобу відео-декодера на базі fpga altera cyclone v.

З цією метою було створено прошивку FPGA. Для цього було створено проект Quartus.

Також реалізація передбачала створення Preloader, який складався з процесу завантаження HPS.

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		64

Описано код для забезпечення доступу до пам'яті FPGA.

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		65

ВИСНОВКИ

В результаті роботи реалізовано програмно-апаратний засіб відео-декодера на базі FPGA altera cyclone v.

В першому розділі подано основні аспекти поняття FPGA. Наведено основні переваги та недоліки застосування для вирішення задач. Таким чином, зроблено висновок щодо можливості використання FPGA для розроблення програмно-апаратної реалізації відео-декодера. В якості засобу створення пропонується залучення FPGA ALTERA CYCLONE V.

В другому розділі подано аналіз функційних можливостей плати розробника DE1-SoC. В розділі наведено комплект поставки, технічні характеристики та основана конфігурація.

В третьому розділі представлено процес програмно-апаратна реалізації та тестування програмно-технічного засобу відео-декодера на базі FPGA altera cyclone v. З цією метою було створено прошивку FPGA. Для цього було створено проект Quartus. Також реалізація передбачала створення Preloader, який складався з процесу завантаження HPS. Описано код для забезпечення доступу до пам'яті FPGA.

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		66

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Матвієнко М.П., Розен В.П., Закладний О.М. Архітектура комп'ютерів: Навчальний посібник . К.: Ліра-К, 2019. 264 с.
2. С.Є. Бантюков, О.В. Чаленко, В.С. Меркулов. Архітектура комп'ютерів та периферійні пристрої: Навч. посібник. Харків: УкрДУЗТ, 2018.Ч. 1. 116 с., рис. 35, табл. 2.
3. А.Я. Чураков, С.В. Шаров, О.В.Строкань. Архітектура ЕОМ. Мелітополь: РВЦ МДПУ, 2012. 195 с.
4. Blanchet G, Dupouy B. Computer Architecture . London: ISTE, 2013. 364 p.
5. Comer D. Essentials of Computer Architecture. Boca Raton, UNITED STATES: Chapman and Hall - CRC, 2017. 536 p.
6. Joseph D. Computer architecture: fundamentals and principles of computer design, University of Tennessee at Chattanooga, Chattanooga, TN, USA - CRC Press, 2017.462 p.
7. J.L. Hennessy, D.A. Patterson. Computer architecture: a quantitative approach. Amsterdam: Morgan Kaufmann, 2018. 856 p.
8. L. Chen, D. Penney, D. Jiménez. AI for computer architecture : principles, practice, and prospects. Morgan & Claypool Publishers, 2021. 142 p.
9. D.M. Harris, S.L. Harris Digital design and computer architecture. Morgan Kaufmann, 2021. - 720 p.
10. R.F. Hodson, A. Kande. Real-time expert systems computer architecture / 1 - CRC Press, 2018. - 300 p.
11. Bindal A. Fundamentals of computer architecture and design. Cham, Switzerland: Springer, 2019. - 606 p.
12. Ledin J. Modern Computer Architecture and Organization: Learn X86, ARM, and RISC-V Architectures and the Design of Smartphones, PCs, and Cloud Servers. 560 p.
13. G. Arroz, J. Monteiro, A. Oliveira, Instituto Superior Técnico, Portugal - Hackensack. Computer architecture : digital circuits to microprocessors. NJ: World Scientific, 2019. 750 p.

					КВРКІ 190188. 14. 14 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		67

14. Микитишин А. Г., Митник М. М., Стухляк П. Д. Комп'ютерні мережі: Львів: «Магнолія 2006», 2021. 256 с. ISBN 978-617-574-087-3.
15. Тарнавський Ю.А., Кузьменко І.М.. Організація комп'ютерних мереж: підручник. КПІ ім. Ігоря Сікорського. Київ: КПІ ім. Ігоря Сікорського, 2018. – 259с.
16. Stallings W. Data and Computer Communications 10th - Pearson, 2013. 912 p.
17. Кулаков Ю.О., Луцький Г.М. Комп'ютерні мережі. Видавництво: Київ, "Юніор" 2005. С.397.
18. T. Qian, L. Chen, X. Li, H. Sun and J. Ni, A 1.25Gbps Programmable FPGA I/O Buffer with Multi-Standard Support, *2018 IEEE 3rd International Conference on Integrated Circuits and Microsystems (ICICM)*, 2018, pp. 362-365, doi: 10.1109/ICAM.2018.8596436.
19. Z. Zhang, Q. Yu, L. Njilla and C. Kamhoua, FPGA-oriented moving target defense against security threats from malicious FPGA tools, *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2018, pp. 163-166, doi: 10.1109/HST.2018.8383907.
20. N. Rezzak, J. -J. Wang, A. Cai, F. Hawley and E. Hamdy, Total Ionizing Dose Characterization of 28 nm PolarFire SONOS-based FPGA, *2019 19th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*, 2019, pp. 1-6, doi: 10.1109/RADECS47380.2019.9745684.
21. U. Farooq, I. Baig and B. A. Alzahrani, An Efficient Inter-FPGA Routing Exploration Environment for Multi-FPGA Systems, *IEEE Access*, vol. 6, pp. 56301-56310, 2018, doi: 10.1109/ACCESS.2018.2873041.
22. J. Yang, H. Guo, L. Chen, Y. Zhang and X. Li, SEU sensitivity evaluation of JTAG circuit used for SRAM-based FPGA, *2017 IEEE 24th International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, 2017, pp. 1-5, doi: 10.1109/IPFA.2017.8060186.
23. Z. Zhang, L. Njilla, C. A. Kamhoua and Q. Yu, Thwarting Security Threats From Malicious FPGA Tools With Novel FPGA-Oriented Moving Target Defense,

Wed May 25 16:01:11 EEST 2022, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 1.0%

 Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 18%**

ID: 103977 Название: Програмно-технічний засіб відео-декодера на базі FPGA Altera Cyclone V Добавлено в БД: 2022-05-25 Авторы: О.О. Кухар Руководители: С.М. Лисенко Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	57344	503	413 (1%)	6 (1%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

Ім'я користувача:
Кафедра КІ

ID перевірки:
1011337362

Дата перевірки:
25.05.2022 16:49:29 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
25.05.2022 16:50:22 EEST

ID користувача:
100005591

Назва документа: Кухар_Програмно-технічний засіб відео-декодера на базі FPGA Altera Cyclone V

Кількість сторінок: 64 Кількість слів: 7887 Кількість символів: 55900 Розмір файлу: 6.55 MB ID файлу: 1011223021

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

1.93%
Схожість

Найбільша схожість: 1.15% з джерелом з Бібліотеки (ID файлу: 1011223018)

0.86% Джерела з Інтернету

14

Сторінка 66

1.33% Джерела з Бібліотеки

106

Сторінка 66

0% Цитат

Не знайдено жодних цитат

Не знайдено жодних посилань

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

3

Підозріле форматування

24
сторінки

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Кухар Олександр Олегович

Тема: Програмно-технічний засіб відео-декодера на базі FPGA Altera Cyclone V

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 65

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є розробка програмно-технічного засобу керування конфігурацією та інфраструктурою локального центру обробки даних.
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: В першому розділі подано основні аспекти поняття FPGA. Наведено основні переваги та недоліки застосування для вирішення задач. Таким чином, зроблено висновок щодо можливості використання FPGA для розроблення програмно-апаратної реалізації відео-декодера. В якості засобу створення пропонується залучення FPGA ALTERA CYCLONE V. В другому розділі подано аналіз функційних можливостей плати розробника DE1-SoC. В розділі наведено комплект поставки, технічні характеристики та основана конфігурація. В третьому розділі представлено процес програмно-апаратна реалізації та тестування програмно-технічного засобу відео-декодера на базі FPGA altera cyclone v. З цією метою було створено прошивку FPGA. Для цього було створено проект Quartus. Також реалізація передбачала створення Preloader, який складався з процесу завантаження HPS. Описано код для забезпечення доступу до пам'яті FPGA.
4. Позитивні сторони роботи: висока практична цінність роботи. В результаті роботи реалізовано програмно-апаратний засіб відео-декодера на базі FPGA altera cyclone V.

4. Позитивні сторони роботи: висока практична цінність роботи. В результаті роботи реалізовано програмно-апаратний засіб відео-декодера на базі FPGA altera cyclone V.

5. Негативні сторони роботи: недостатня увага моделюванню схеми в середовищі Quartus.

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 належному науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільно (3.00/ E)

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

Джулій В.М., к.т.н., доцент, кафедри кібербезпеки Хмельницького національного університету

“ ___ ” _____ 2022 р.

 (підпис)

Завідувачу кафедри КІСП
д-ру техн.наук, проф. Говорушенко Т. О.

Кухар О.

ІІІ, здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2с-19-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність плагіату ознайомлений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

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

дата

підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОНУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмно-технічний засіб відео-декодера на базі FPGA Altera Cyclone V

Автор: Кухар Олександр Олегович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Лисенко Сергій Миколайович, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та дорпрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

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

- 1) запозичення розміщені в розділі аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформлені посилання;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів щодо використаних програмних скриптів, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 1,93% і адресується до 14 періоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру практичної роботи і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОН

Завідувач кафедри КІСП

С.М. Лисенко

С. М. Лисенко

Т. О. Говорушенко