

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

Щирба Зоряна Володимирівна

**VHDL-модель сенсора температури / VHDL-
model of temperature sensor**

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

Випускна кваліфікаційна робота

Виконав: студент групи КСМ-42/1
Щирба Зоряна Володимирівна

Науковий керівник:
ктн, Дубчак Л.О.

Випускну кваліфікаційну роботу
допущено до захисту:

" ___ " _____ 20__ р.

Завідувач кафедри
О. М. Березький

ТЕРНОПІЛЬ - 2019

РЕЗЮМЕ

Бакалаврська робота містить 63 сторінки пояснювальної записки, 8 цифр, 7 таблиць, 3 заявки. Обсяг графічного матеріалу - 2 аркуші формату А3.

Метою бакалаврської роботи є розробка моделі VHDL датчика температури. Ця бакалаврська робота присвячена питанням сучасного стану застосування датчиків температури. Розглянуто опис основних характеристик датчика температури. В даній роботі проведено аналіз основних вимог до розробки проекту, розроблено етапи проектування та обрано поточне середовище проектування, проаналізовано алгоритм і обрано основні компоненти датчика температури AD7814, вхідний були проаналізовані вихідні змінні, проведено моделювання та моделювання проекту в середовищі Active-HDL, проект.

Розроблена модель температурного датчика AD7814, яка має практичне застосування в сучасних комп'ютерних системах.

Ключові слова: ДАТЧИК, СЕНСОР ТЕМПЕРАТУРИ, AD7814, VHDL, ACTIVE-HDL.

RESUME

Bachelor work contains 63 pages of explanatory note, 8 figures, 7 tables, 3 applications. The volume of graphic material is 2 sheets of a3 format.

The aim of the bachelor work is to develop a temperature sensor VHDL model.

This baccalaureate work is devoted to the issues of the current state of application of temperature sensors. The description of the main characteristics of the temperature sensor was considered. In this work the analysis of the main requirements for the development of the project was carried out, the design stages were elaborated and the current design environment was selected, the algorithm was analyzed and the main components of the AD7814 temperature sensor were selected, the input and output variables were analyzed, the simulation and simulation of the project in the Active-HDL environment were carried out, project.

The AD7814 temperature sensor HDL model has been developed, which has practical application in modern computer systems.

Keywords: SENSOR, SENSOR TEMPERATURE, AD7814, VHDL, ACTIVE-HDL.

ЗМІСТ

Вступ.....	8
1 Аналіз сучасних сенсорів та їх використання	9
1.1 Сучасні сенсори та їх застосування.....	9
1.2 Формування вимог дипломного проектування.....	14
1.3 Постановка задачі та аналіз дерева рішень	17
2 Алгоритми роботи та компоненти сенсора температури AD7814	19
2.1 Загальний опис сенсора температури AD7814	19
2.2 Робота датчика та алгоритми роботи	20
2.3 Складові сенсора температури AD7814.....	23
3 Реалізація проекту в середовищі Active-HDL.....	29
3.1 Інтерфейсний опис проекту	29
3.2 Моделювання сенсора температури в середовищі Active-HDL.....	33
3.3 Симуляція та верифікація проекту	36
4 Техніко-економічне обґрунтування розробки програмного засобу.....	40
4.1 Розрахунок витрат на розробку програмного забезпечення	41
4.2 Визначення експлуатаційних витрат.....	47
4.3 Розрахунок ціни споживання програмного продукту	49
4.4 Визначення показників економічної ефективності.....	51
4.5 Висновки	52
<u>Висновки</u>	53
Список використаних джерел.....	54
Додаток А VHDL-код моделі	56
Додаток Б Довідка про впровадження	57
Додаток В Світокопія тез доповіді.....	62

					БР.КСМ. 07129/15.00.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	VHDL-модель сенсора температури	Літ.	Арк.	Акрушів
Розроб.		Щирба З.В.					8	63
Перевір.		Дубчак Л.О.						
Конс.		Паздрій І.Р.						
Н. Контр.		Гураль І.В.						
Затверд.		Березький О.М.				ТНЕУ. ФКІТ. КСМ-42/1		

ВСТУП

У повсякденному житті люди звикли часто використовувати різні типи датчиків у різних енергетичних системах, таких як електричні та електронні прилади, системи контролю навантаження, домашня автоматизація або промислова автоматизація тощо.

Сенсор температури – це пристрій, що використовується для вимірювання кількості теплової енергії, визначає зміну температури від конкретного джерела та перетворює дані для пристроїв чи користувачів.

Датчик температури відіграє важливу роль у багатьох застосуваннях. Наприклад, підтримка певної температури необхідна для обладнання, що використовується для виготовлення медичних препаратів, нагрівання рідин або очищення іншого обладнання. Для таких застосувань, якість і точність схеми виявлення можуть бути критичними для контролю якості.

Метою бакалаврської роботи є розробка VHDL-моделі сенсора температури.

Для цього необхідно розв'язати наступні задачі:

- проаналізувати сучасні сенсори температури та визначити їх основні характеристики;
- створити дерево рішень та вибрати середовище проектування;
- виділити основні складові сенсора температури;
- створити HDL-модель сенсора температури AD7814 засобами програмного забезпечення Active-HDL та провести верифікацію проекту.

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ СУЧАСНИХ СЕНСОРІВ ТА ЇХ ВИКОРИСТАННЯ

1.1 Сучасні сенсори та їх застосування

У широкому розумінні сенсор – це пристрій, модуль або підсистема, метою якого є виявлення подій або змін у його оточенні та передача інформації іншій електроніці, часто процесору комп'ютера.

Датчиками називають частини технічних систем, призначених для вимірювання, сповіщення, регулювання, управління приладами або процесами. Вони перетворюють будь яку величину, яка вимірюється (тиск, температура, витрати, частота, швидкість, напруга, струм і т.д.) у електричний, оптичний або пневматичний сигнал, зручний для вимірювання, передавання, перетворення та зберігання інформації про стан об'єкту вимірювання [1].

Історично та логічно датчики пов'язані з технікою вимірювання та вимірювальними приладами, такими як: термометри, витратоміри, прилади для вимірювання тиску та інше. Узагальнювальний термін «давач (датчик)» закріпився завдяки розвитку автоматичних систем керування, як елемент узагальненої логічної концепції «давач – система керування – виконавчий пристрій – об'єкт керування». Спеціальним випадком є використання давачів у автоматичних системах реєстрації параметрів, наприклад, у системах наукових досліджень [2].

Датчики – це складні пристрої, які часто використовуються для виявлення та реагування на електричні або оптичні сигнали. Датчик перетворює фізичний параметр (наприклад, температуру, кров'яний тиск, вологість, швидкість і т.д.) в сигнал, який може бути визначений як електричний.

Останнім часом стосовно давачів застосовуються терміни: «багатофункціональний давач» чи «інтелектуальний давач», що відображає напрямки розвитку сучасних давачів. Під цими термінами, крім функцій первинного вимірювального перетворення, мають на увазі додаткові можливості вимірювання декількох фізичних величин та використання вбудованих аналого-

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

цифрових перетворювачів з мікроконтролерами, що суттєво розширює функціональний діапазон давачів, а саме [3]:

- попередня обробка сигналів (лінеаризація, фільтрування, корекція похибок);
- самодіагностування;
- дистанційне конфігурування (діапазону вимірювань, одиниць вимірювань, узгодження частотних характеристик);
- окремі елементи керування;
- передавання інформації з використанням протоколів промислових мереж Profibus чи Foundation Fieldbus.

Класифікують сенсори за багатома характеристиками. За принципом перетворення енергії розрізняють активні та пасивні давачі, що відрізняються способами формування сигналу та схемами підключення [1]:

- активні (генераторні) – давачі, у яких здійснюється перетворення видів енергії від входу до виходу;
- пасивні (параметричні) – давачі, у котрих вхідна енергія змінює параметри визначених елементів первинних вимірювальних перетворювачів.

Всі типи датчиків можна класифікувати в основному на аналогові та цифрові сенсори. Але є кілька типів датчиків, таких як датчики температури, датчики інфрачервоного випромінювання, ультразвукові датчики, датчики тиску, датчики наближення та сенсори дотику, які часто використовуються в більшості прикладних програм [3].

Пристрій, який виявляє наявність або відсутність об'єкта, що знаходиться поруч, або властивості цього об'єкта, і перетворює його в сигнал, який може бути легко прочитаний користувачем або простим електронним інструментом без контакту з ними, називається датчиком наближення.

Датчики близькості в основному використовуються в роздрібній промисловості, оскільки вони можуть виявляти рух і співвідношення між клієнтом і продуктом, який вони можуть зацікавити. Вони також використовуються для паркування автомобілей.

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Датчик тиску – це пристрій, який відчуває тиск і перетворює його в електричний сигнал. Тут кількість залежить від рівня тиску. Є багато пристроїв, які покладаються на рідину або інші форми тиску. Ці датчики дозволяють створювати системи, які контролюють системи та пристрої, що працюють під тиском. При будь-якому відхиленні від стандартного діапазону тиску пристрій сповіщає системного адміністратора про будь-які проблеми, які необхідно виправити [2].

Датчики якості води використовуються для виявлення якості води та моніторингу іонів, головним чином, у системах розподілу води. Вода практично використовується скрізь. Ці датчики відіграють важливу роль, оскільки вони контролюють якість води для різних цілей. Вони використовуються в різних галузях промисловості.

Хімічні сенсори застосовуються в ряді різних галузей промисловості. Їх мета полягає в тому, щоб вказати зміни в рідині або виявити хімічні зміни повітря. Вони відіграють важливу роль у великих містах, де необхідно відстежувати зміни і захищати населення.

Основні випадки використання хімічних сенсорів можна знайти в промисловому моніторингу навколишнього середовища та контролю технологічних процесів, навмисно або випадково випустили шкідливе хімічне виявлення, вибухонебезпечне та радіоактивне виявлення, процеси утилізації на космічній станції, фармацевтичній промисловості та лабораторії тощо.

Газові сенсори подібні до хімічних, але спеціально використовуються для моніторингу зміни якості повітря та виявлення наявності різних газів. Як і хімічні сенсори, вони використовуються в багатьох галузях, таких як виробництво, сільське господарство та охорона здоров'я, і використовуються для моніторингу якості повітря, виявлення токсичних або горючих газів, моніторинг небезпечних газів у вугільних шахтах, нафтогазова промисловість, хімічні лабораторні дослідження, виробництво – фарби, пластмаси, гумові, фармацевтичні та нафтохімічні [2].

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Датчик диму – це пристрій, який відчуває дим (повітряні частинки та газу) і його рівень. Вони використовуються протягом тривалого часу. Однак, з розвитком технологій, вони тепер ще більш ефективні, оскільки вони підключені до системи, яка негайно сповіщає користувача про будь-яку проблему, що виникає в різних галузях.

Датчики диму широко використовуються обробною промисловістю, будинками та інфраструктурою розміщення для виявлення пожежних та газових випадків. Це слугує для захисту людей, які працюють у небезпечних умовах, оскільки вся система набагато ефективніша в порівнянні зі старими.

Інфрачервоний датчик – це датчик, який використовується для визначення певних характеристик його оточення шляхом випромінювання або виявлення інфрачервоного випромінювання. Він також здатний вимірювати тепло, що випромінюється об'єктами [4].

В даний час вони використовуються в різних проектах, особливо в сфері охорони здоров'я, оскільки вони роблять моніторинг кровотоку і артеріального тиску простим. Вони навіть використовуються в широкому спектрі звичайних смарт-пристроїв, таких як смарт-годинники та смартфони. Інше поширене використання включає побутову техніку та дистанційне керування, аналіз дихання, інфрачервоне бачення (тобто візуалізацію витоків тепла в електроніці, моніторинг кровотоку, історичні мистецтва, щоб побачити під шарами фарби), носну електроніку, оптичне сполучення, безконтактні вимірювання температури, виявлення автомобільного сліпого кута.

Датчики зображення є інструментами, які використовуються для перетворення оптичних зображень в електронні сигнали для відображення або зберігання файлів в електронному вигляді.

Найбільше використання датчика зображення можна знайти в цифрових камерах і модулях, медичному обладнанні для зображень і обладнанні для нічного бачення, тепловізійних пристроях, радарх, гідролокаторах, засобах медіа, біометричних і IRIS пристроях.

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

Детектор руху – це електронний пристрій, який використовується для виявлення фізичного руху (руху) в заданій області і перетворює рух у електричний сигнал; руху будь-якого об'єкта або руху людських істот [5].

Виявлення руху відіграє важливу роль у галузі безпеки. Підприємства використовують ці датчики в тих місцях, де не повинно бути виявлено жодного руху в будь-який час, і легко помітити присутність тих, хто має ці датчики. Вони використовуються в основному для систем виявлення вторгнень, автоматика для управління дверима, смарт-камери (тобто відеозапису на основі руху), автоматичних систем паркування, автоматичних мийок, сушарок для рук, систем управління енергією (тобто освітлення, змінного струму, вентилятора, керування приладами).

З іншого боку, ці датчики також можуть розшифровувати різні типи рухів, роблячи їх корисними в деяких галузях, де клієнт може спілкуватися з системою, розмахуючи рукою або виконуючи подібну дію. Наприклад, хтось може помахати датчиком у роздрібному магазині, щоб отримати допомогу у прийнятті правильного рішення про покупку.

Незважаючи на те, що їх первинне використання співвідноситься з індустрією безпеки, оскільки технологія прогресує, кількість можливих застосувань цих датчиків тільки зростатиме.

Акселерометр – датчик, який використовується для вимірювання фізичного або вимірюваного прискорення, яке відчуває об'єкт через інерційні сили і перетворює механічний рух в електричний вихід. Вона визначається як швидкість зміни швидкості по часу [6].

Ці датчики тепер присутні в мільйонах пристроїв, таких як смартфони. Їх використання передбачає виявлення вібрацій, нахилу і прискорення в цілому. Це відмінно підходить для моніторингу вашого автопарку або за допомогою смарт-крокоміра. У деяких випадках він використовується як форма захисту від крадіжки, оскільки датчик може відправляти оповіщення через систему, якщо переміщується об'єкт, який повинен залишатися нерухомим.

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Вологість визначається як кількість водяної пари в атмосфері повітря або інших газів. Найбільш часто використовувані терміни – відносна вологість (RH).

Ці датчики зазвичай використовують датчики температури, оскільки багато виробничих процесів вимагають досконалих умов роботи. За допомогою вимірювання вологості можна переконатися, що весь процес проходить безперебійно, а коли відбувається будь-яка раптова зміна, дія може бути прийнята негайно, оскільки датчики виявляють зміну майже миттєво. Їх застосування та використання можуть бути знайдені в промисловому та житловому обладнанні для управління системами опалення, вентиляції та кондиціонування повітря. Їх можна також знайти в автомобілях, музеях, промислових приміщеннях і теплицях, станціях метеорології, промисловості фарб і покриттів, лікарнях та фармацевтичних галузях для захисту лікарських засобів [7].

1.2 Формування вимог дипломного проектування

За визначенням, пристрій, що використовується для вимірювання кількості теплової енергії, що дозволяє виявити фізичну зміну температури від конкретного джерела і перетворює дані для пристрою або користувача, відомий як датчик температури.

Ці датчики були присутні протягом тривалого часу в різних пристроях. Лише кілька років тому їх використовували, в основному, у кондиціонерах, холодильниках та аналогічних пристроях, що використовуються для контролю навколишнього середовища.

Однак з приходом у світ новіших технологій, вони знайшли свою роль у виробничих процесах, сільському господарстві та галузі охорони здоров'я. У процесі виробництва багато машин вимагають певної температури

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

навколишнього середовища, а також температури пристрою. При такому вимірі виробничий процес завжди може залишатися оптимальним.

З іншого боку, в сільському господарстві температура ґрунту має вирішальне значення для зростання врожаю. Це допомагає при вирощуванні рослин, максимізації виходу і т.д.

Є декілька підкатегорій датчиків температури. Датчики – це пристрої напруги, які вказують на вимірювання температури зі зміною напруги. Зі збільшенням температури вихідна напруга термопар підвищується [6].

Датчики температури резисторів (RTD): працюють на принципі, коли опір пристрою збільшується, якщо температура підвищується.

Термістори – це термочутливі резистори, які змінюють свій фізичний опір зі зміною температури.

ІС (напівпровідник) є лінійним пристроєм, де провідність напівпровідника лінійно зростає і використовує переваги властивостей змінного опору напівпровідникових матеріалів. Вона може забезпечити пряме читання температури в цифровій формі, особливо при низьких температурах.

Правильний і належний датчик повинен відповідати таким характеристикам – чутливий до вимірюваної властивості, нечутливий до будь-якого іншого властивості. Чутливість – це співвідношення між вихідним сигналом і вимірюваною властивістю. Наприклад, якщо датчик вимірює температуру і має вихідну напругу, чутливість є постійною з одиницями V/K. Чутливість є нахилом передавальної функції. Трансформація електричного виходу датчика (наприклад, V) в одиницю вимірювання (наприклад, K) вимагає поділу електричного виходу на нахил (або множення на його зворотний) [8].

Для того, щоб аналоговий сигнал датчика був оброблений або використаний в цифровому обладнанні, він повинен бути перетворений в цифровий сигнал за допомогою аналого-цифрового перетворювача. Оскільки датчики не можуть повторювати ідеальну функцію передачі, можуть бути кілька типів відхилень, які обмежують точність датчика. Оскільки діапазон вихідного сигналу завжди обмежений, вихідний сигнал в кінцевому рахунку досягне

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

мінімального або максимального, коли виміряне властивість перевищує межу. Діапазон повної шкали визначає максимальне та мінімальне значення вимірюваної властивості [8].

Чутливість на практиці може відрізнятися від заданої величини. Це називається помилкою чутливості. Це помилка в нахилі лінійної передавальної функції. Якщо вихідний сигнал відрізняється від правильного постійного значення, датчик має помилку зсуву. Це помилка у-перехоплення лінійної передатної функції [8].

Нелінійність – це відхилення функції передавача датчика від функції передачі прямої лінії. Це зазвичай визначається вихідним значенням, яке відрізняється від ідеальної поведінки в діапазоні датчика, часто відзначається як відсоток від повного діапазону. Відхилення, викликане швидкими змінами вимірюваної властивості в часі, є динамічною помилкою. Часто така поведінка описується графічним індикатором, який показує похибку чутливості і фазовий зсув як функцію частоти періодичного вхідного сигналу.

Якщо вихідний сигнал змінюється повільно незалежно від вимірюваної властивості, він визначається як дрейф. Довгостроковий дрейф протягом місяців або років викликаний фізичними змінами в датчику.

Шум – це випадкова зміна сигналу, який змінюється в часі. У результаті помилки гістерезису початкове значення змінюється залежно від попередніх вхідних значень. Якщо вихід датчика відрізняється залежно від того, чи було досягнуто певне вхідне значення за рахунок збільшення або зменшення вхідного сигналу, то датчик має помилку гістерезису [8].

Якщо датчик має цифровий вихід, вихідний сигнал, по суті, є наближенням вимірюваної властивості. Ця помилка також називається помилкою квантування.

Якщо сигнал керується цифровим способом, частота дискретизації може викликати динамічну помилку або, якщо вхідна змінна або додана зміна шуму при частоті декількох частот дискретизації, можуть виникнути помилки накладання.

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Датчик може певною мірою бути чутливим до властивостей, відмінних від вимірюваного властивості. Наприклад, на більшість датчиків впливають температури навколишнього середовища.

Всі ці відхилення можна класифікувати як систематичні або випадкові помилки. Систематичні помилки іноді можуть бути компенсовані стратегією калібрування. Шум є випадковою помилкою, яка може бути зменшена шляхом обробки сигналу, такого як фільтрація, зазвичай внаслідок динамічної поведінки датчика [8].

1.3 Постановка задачі та аналіз дерева рішень

В даному дипломному проекті поставлена задача розробки HDL-моделі сенсора температури AD7814. Процес проектування складається з таких етапів, котрі зображені на рисунку 1.1.

Дана компонента розробляється в середовищі Active-HDL 9.1. Програма дозволяє проектувати пристрій за допомогою мов опису апаратури, а також за допомогою структурних схем. Спочатку програма підтримувала тільки мову VHDL, але з часом додалася підтримка мов Verilog і SystemC. За допомогою програми можна графічно проектувати кінцеві автомати, а також конвертувати HDL опис в графічні структурні схеми і навпаки [9].

Спочатку для більшого розуміння опишемо подане дерево рішень дипломного проектування зображеного на рисунку 1.1. Першим кроком є – аналіз сучасних сенсорів температури. Наступним кроком формуємо вимоги та визначаємо основні характеристики сенсора. Третім кроком дипломного проектування є вибір датчика температури, в даному випадку це AD7814. Після цього вибираємо відповідну систему автоматизованого проектування. П'ятим кроком є створення VHDL– моделі сенсора температури AD7814. Заключним

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

етапом, зображеним на дереві рішень дипломного проектування є перевірка правильності роботи поданої VHDL-моделі.

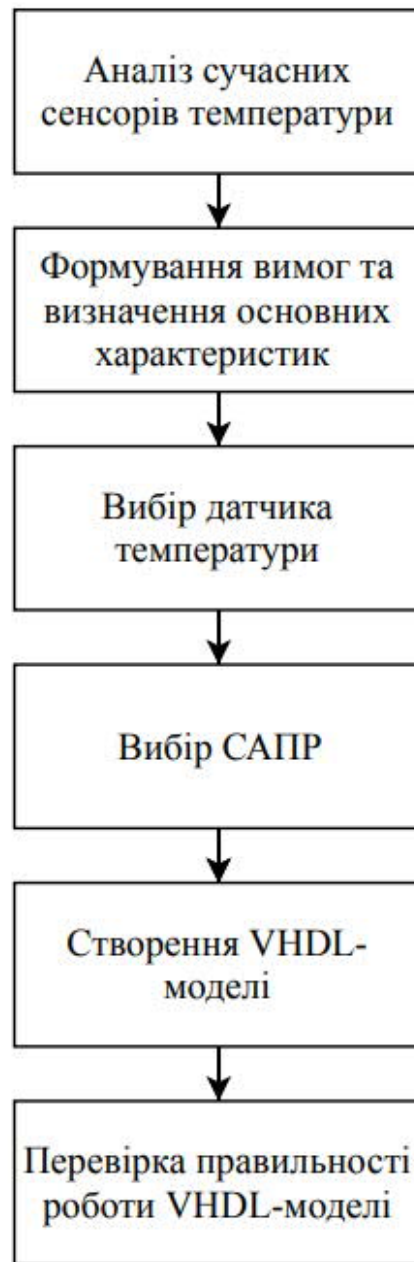


Рисунок 1.1 – Дерево рішень дипломного проектування

2 АЛГОРИТМИ РОБОТИ ТА КОМПОНЕНТИ СЕНСОРА ТЕМПЕРАТУРИ AD7814

2.1 Загальний опис сенсора температури AD7814

AD7814 є повноцінною системою моніторингу температури в пакеті SOT-23 або MSOP з 8 ліній. Він містить датчик температури зазору та 10-бітний АЦП для моніторингу та оцифрування зчитування температури до роздільної здатності $0,25^{\circ}\text{C}$.

Датчик AD7814 має гнучкий послідовний інтерфейс, що дозволяє легко взаємодіяти з більшістю мікроконтролерів. Інтерфейс сумісний з протоколом SPI, QSPI і MICROWIRE, а також сумісний з DSP. Частина має режим очікування, керований через послідовний інтерфейс [10].

Широкий діапазон напруги живлення AD7814, низький струм живлення та сумісний із SPI інтерфейс роблять сенсор ідеальним для різних застосувань, включаючи персональні комп'ютери, офісне обладнання та побутову техніку.

Існує п'ять основних характеристик датчика:

- AD7814 має вбудований датчик температури, який дозволяє точно вимірювати температуру навколишнього середовища;
- повний теоретичний діапазон АЦП від -128°C до $+127^{\circ}\text{C}$;
- вимірюваний діапазон температур становить від -55°C до $+125^{\circ}\text{C}$, з точністю до температури $\pm 2^{\circ}\text{C}$;
- напруга живлення від 2,7 В до 5,5 В;
- пакет SOT-23, що заощаджує простір, і пакет MSOP з 8 ліній;
- 10-бітове відтворення температури до $0,25^{\circ}\text{C}$;
- перетворення відбувається приблизно кожні 350 мкс;
- режим очікування, який зменшує споживання струму до 1 мкА.

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

2.2 Робота датчика та алгоритми роботи

AD7814 – це 10-бітний цифровий датчик температури. У корпусі розміщено датчик температури на мікросхемі, 10-розрядний A/D-конвертер, еталонні і послідовні логічні функції в пакеті SOT-23.

Секція АЦП складається зі звичайного перетворювача послідовного наближення, що базується на конденсаторі ЦАП. Частини здатні працювати від джерела живлення напругою від 2,7 В до 5,5 В.

Датчик температури на кристалі дозволяє проводити точне вимірювання температури навколишнього пристрою. Робочий діапазон вимірювання AD7814 становить від -55°C до $+125^{\circ}\text{C}$.

Годинники перетворення генеруються всередині, тому не потрібно ніяких зовнішніх годин, крім як при читанні і запису в послідовний порт. У звичайному режимі внутрішній тактовий генератор запускає послідовність автоматичного перетворення. Перетворення починається приблизно кожні 350 мкс. В цей час деталь прокидається і виконує перетворення температури. Це перетворення температури зазвичай займає 25 мкс, після чого деталь автоматично відключається. Результат самого останнього перетворення температури доступний в регістрі послідовного виходу в будь-який час.

Як тільки перетворення завершено, внутрішній генератор починає відлік і розрахований на тайм-аут кожні 350 мкс. Потім AD7814 включається і виконує перетворення. Якщо значення CS знижується кожні 350 мкс ($\pm 30\%$), то одне і те ж значення температури буде виводитися на лінію DOUT кожен раз без зміни. Рекомендується, щоб лінія CS не знижувалася кожні 350 мкс ($\pm 30\%$). $\pm 30\%$ охоплює зміна процесу. CS повинен стати активним (від високого до низького) за межами цього діапазону [10].

Пристрій призначений для автоматичного перетворення кожні 350 мкс. Якщо до AD7814 звертаються під час процесу перетворення, генерується внутрішній сигнал, щоб запобігти будь-яке оновлення регістра значення

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

температури під час перетворення. В іншому випадку користувач може прочитати неправдиві дані.

Конструкція цієї функції призводить до того, що цей сигнал внутрішнього блокування скидається тільки на початку наступної автоконверсії. Отже, якщо лінія CS активується до того, як сигнал внутрішньої блокування був скинутий в свій неактивний режим, сигнал внутрішньої блокування не буде скинутий.

Щоб переконатися, що сигнал блокування не встановлено, встановіть низький рівень CS на більший чи менший час, ніж 350 мкс ($\pm 30\%$). Це буде означати, що AD7814 не переривається під час процесу перетворення.

AD7814 може бути переведений в режим вимкнення через регістр управління, і в цьому випадку генератор на кристалі відключається, і ніякі подальші перетворення не ініціюються, поки AD7814 не буде виведений з режиму вимкнення.

Результат перетворення, отриманий при останньому перетворенні перед вимиканням, все ще може зчитуватися з AD7814, навіть коли він знаходиться в режимі виключення.

У режимі автоматичного перетворення кожного разу, коли відбувається операція читання або запису, внутрішній тактовий генератор перезавантажується в кінці операції читання або запису. Результат перетворення зазвичай доступний через 25 мкс. Аналогічним чином, коли деталь виводиться з режиму вимкнення, внутрішній тактовий генератор перезавантажується, і результат перетворення зазвичай доступний через 25 мкс. Читання з пристрою до завершення перетворення надасть той же набір даних.

Регістр значення температури є регістром тільки для читання, в якому зберігаються показання температури з АЦП в форматі з додаванням 10 бітів. Формат даних про температуру показано у таблиці I. Він показує повний теоретичний діапазон АЦП від -128°C до $+127^{\circ}\text{C}$, але на практиці діапазон виміру температури обмежений діапазоном робочих температур пристрої (від мінус 55°C до плюс 125°C).

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

Послідовний інтерфейс AD7814 складається з чотирьох проводів CS, SCLK, DIN і DOUT. Інтерфейс може працювати в 3-х дротовому режимі з DIN, прив'язаним до заземлення, в цьому випадку інтерфейс має можливість тільки для читання, при цьому дані зчитуються з регістра даних через лінію DOUT.

Лінія DIN використовується для переключення деталі в режим очікування, якщо потрібно. Лінія CS використовується для вибору пристрою, коли до послідовним годинах і лініях даних підключено більше одного пристрою [11].

Частина працює в підлеглому режимі і вимагає, щоб зовні застосовані послідовні годинник надходили на вхід SCLK для доступу до даних з регістра даних. Послідовний інтерфейс AD7814 призначений для забезпечення можливості сполучення деталі з системами, які забезпечують синхронізацію послідовних сигналів з послідовними даними, такими як мікроконтролери 80C51, 87C51, 68HC11, 68HC05 і PIC16Cxx, а також процесори DSP.

Операція читання з AD7814 здійснює доступ до даних з регістра значень температури, в той час як операція запису в деталь записує дані в регістр управління.

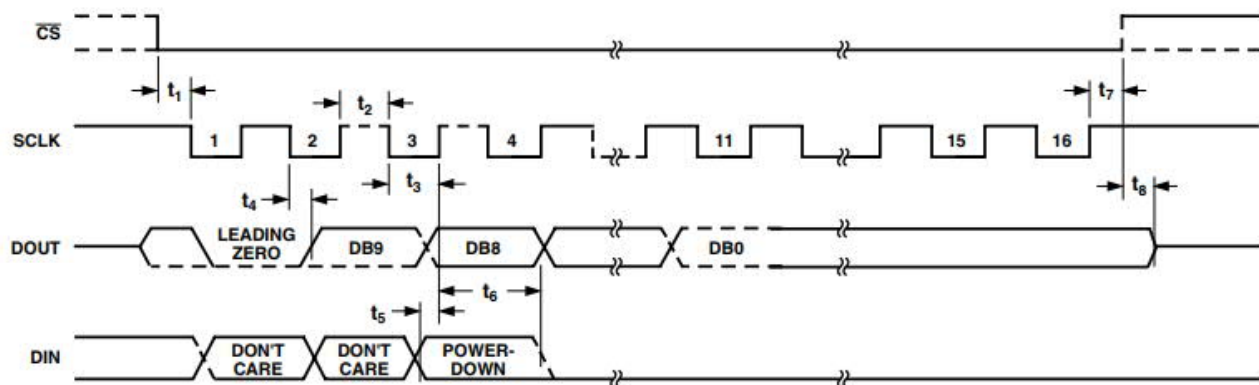


Рисунок 2.1 – Схема послідовного інтерфейсу

На рисунку 2.1 показана тимчасова діаграма для послідовного читання з AD7814. Лінія CS включає вхід SCLK. Десять біт даних і ведучий нуль передаються під час операції читання. Операції читання відбуваються під час

потоків по 16 тактових імпульсів. До послідовним даними звертаються в кількості байтів, якщо зчитуються 10 біт даних.

В кінці операції читання лінія DOUT залишається в стані останнього біта даних, синхронізованих з AD7814, до тих пір, поки CS не досягне високого рівня, після чого лінія DOUT перейде до третього стану.

На рисунку 2.1 також показана тимчасова діаграма для послідовного запису в AD7814. Операція запису відбувається одночасно з операцією читання. Дані синхронізуються з контрольним регістром на частини на передньому фронті SCLK. Тільки третій біт в потоці даних забезпечує керувану користувачем функцію. Цей третій біт є бітом відключення живлення, який при значенні 1 переводить AD7814 в режим вимкнення.

Перші 2 біта потоку даних не враховуються, в той час як всі інші біти в потоці даних, крім біта відключення живлення, повинні бути 0, щоб забезпечити правильну роботу AD7814.

Дані завантажуються в керуючий регістр на п'ятнадцятому падаючому фронті SCLK, і дані вступають в силу в цей час (тобто, якщо деталь запрограмована на відключення, вона робить це в цей момент).

2.3 Складові сенсора температури AD7814

Послідовний інтерфейс AD7814 забезпечує легкий інтерфейс для більшості мікрокомп'ютерів і мікропроцесорів. На рисунках 3-6 показані деякі типові інтерфейсні схеми. Послідовний інтерфейс AD7814 складається з чотирьох проводів: CS, DIN, DOUT і SCLK. Всі показані інтерфейсні схеми використовують всі чотири інтерфейсні лінії. Проте, інтерфейс може працювати з трьома проводами.

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

Якщо для застосування не потрібно функція відключення живлення, пропонована AD7814, лінія DIN може бути постійно прив'язана до низького рівня.

Таким чином, інтерфейс може управлятися лише трьома проводами: SCLK, CS і DOUT. Послідовна передача даних в та із AD7814 вимагає 16-бітної операції читання. Багато 8-розрядні мікроконтролери мають 8-розрядні послідовні порти, і ця 16-розрядна передача даних обробляється як дві 8-розрядні передачі. Інші мікроконтролери і процесори DSP передають 16 біт даних в операції послідовних даних [10].

На рисунку 2.2 зображений інтерфейс між AD7814 і мікро контролером MC68HC11. MC68HC11 налаштований в режимі майстра з бітом CPOL, встановленим на логічну одиницю, і бітом CPHA, встановленим на логічну одиницю. Коли MC68HC11 налаштований таким чином, його лінія SCLK не діє між передачами даних.

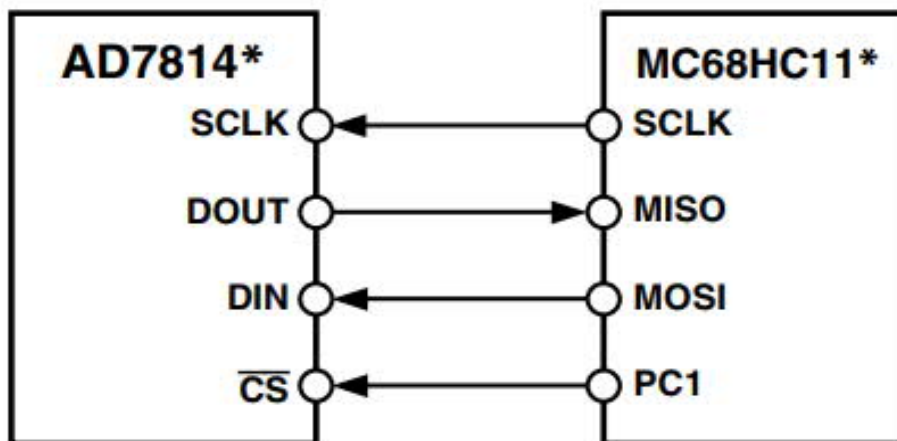


Рисунок 2.2 – Інтерфейс AD7814 - MC68HC11

Дані передаються в та із AD7814 в двох 8-бітних послідовних операціях даних. На схемі представлений повний (4-х провідний) інтерфейс. PC1 MC68HC11 налаштований як вихід і використовується для управління входом CS.

Схема інтерфейсу між AD7814 і мікро контролером 8051 зображена на рисунку 2.3. 8xC51 налаштований в режимі послідовного інтерфейсу в режимі 0. Послідовна тактова лінія 8xC51 (на P3.1) простоює між передачами даних. Дані передаються в і з AD7814 в двох 8-бітних послідовних операціях даних. AD7814 виводить MSB свого потоку даних в якості першого дійсного біта, в той час як 8xC51 очікує LSB першим.

Таким чином, дані, що передаються в послідовний буфер, повинні бути змінені таким чином, перш ніж правильне слово даних з AD7814 буде доступно в акумуляторі. У показаному прикладі AD7814 підключений до послідовного порту 8051.

Оскільки послідовний інтерфейс 8xC51 містить тільки одну лінію даних, лінія DIN AD7814 має низьку прив'язку в прикладі інтерфейсу, показаному на рисунку 2.3. Для додатків, якщо для AD7814 необхідно використовувати функцію відключення живлення, послідовний інтерфейс повинен бути реалізований з використанням ліній порту даних на 8051.

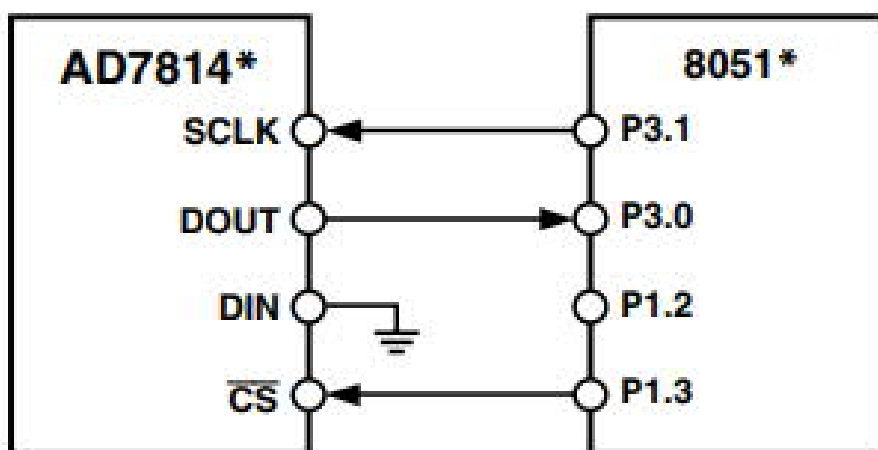


Рисунок 2.3 – Інтерфейс AD7814 – 8051

Це дозволяє реалізувати повнодуплексний послідовний інтерфейс. Цей метод включає в себе «бітову розрядку» лінії порту для генерації послідовного тактового сигналу, в той час як дві інші лінії порту використовуються для переміщення даних в четверту лінію порту, підключену до CS. Лінії портів з 1.0

по 1.3 (з P1.1, налаштованим в якості входу) можуть використовуватися для підключення до SCLK, DOUT, DIN і CS, відповідно, для реалізації цієї схеми.

На рисунку 2.4 зображена схема інтерфейсу між AD7814 і мікроконтролером PIC16C6x / 7x. Синхронний послідовний порт (SSP) PIC16C6x / 7x налаштований як майстер SPI з бітом полярності тактового сигналу, встановленим в логічний біт.

В цьому режимі послідовна лінія синхронізації PIC16C6x / 7x простоє між передачами даних. Дані передаються в і з AD7814 в двох 8-бітних послідовних операціях даних. У показаному прикладі лінія порту RA1 використовується для генерації CS для AD7814.

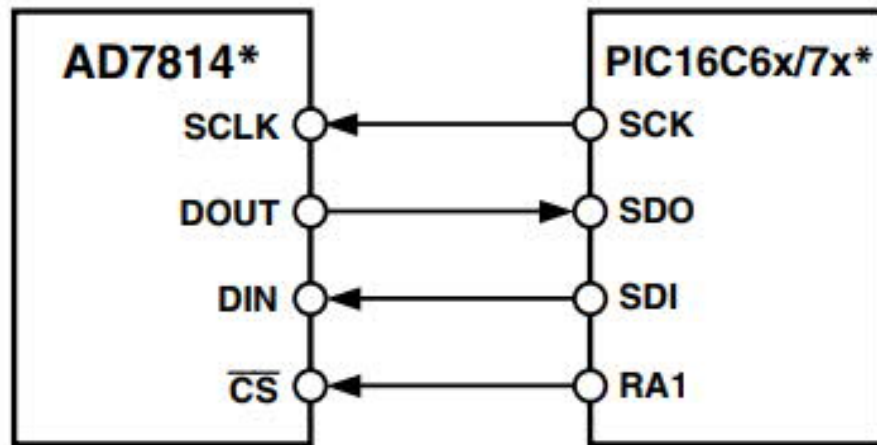


Рисунок 2.4 – Інтерфейс AD7814 - PIC16C6x / 7x

На рисунку 2.5 зображений інтерфейс між сенсором температури AD7814 і процесором DSP ADSP-21xx.

Для забезпечення правильного та коректного функціонування інтерфейсу, регістр управління SPORT повинен бути налаштований таким чином:

- TFSW = RFSW = 1 – альтернативний кадр;
- INVRFS = INVTFS = 1 – активний сигнал низького кадрування;
- DTYPE = 00 – вирівнювання праворуч;
- SLEN = 1111, 16 – бітові слова даних;
- ISCLK = 1 – внутрішній серійний годинник;

- TFSR = RFS = 1 – кадр кожного слова;
- IRFS = 0, RFS – налаштований як вхідний;
- ITFS = 1, TFS – налаштований як вихідний.

Інтерфейс вимагає інвертора між SCLK лінією ADSP-21xx і входом SCLK AD7814. ADSP21xx має TFS і RFS SPORT, пов'язані з TFS, встановленими як вихідні сигнали, і RFS, встановлені як вхідні дані. DSP працює в режимі альтернативного кадрівання, а регістр керування SPORT встановлюється, як описано раніше [11].

AD7814 можна використовувати для вимірювання температури поверхні або повітря. Якщо пристрій приклеєний до поверхні за допомогою теплопровідного клею, температура матриці буде в межах приблизно 0,1 ° C від температури поверхні завдяки низькому енергоспоживанню пристрою.

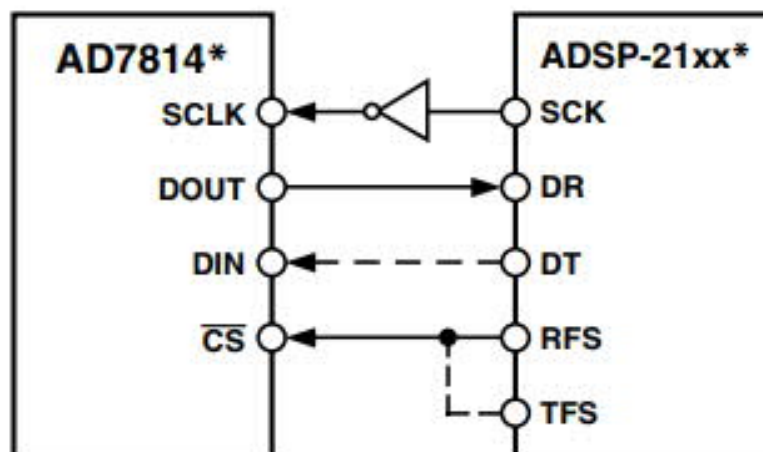


Рисунок 2.5 – AD7814 до інтерфейсу ADSP-21xx

Необхідно дбати про ізоляцію задньої стінки та провідників пристрою від повітря, якщо температура навколишнього повітря відрізняється від температури поверхні, що вимірюється.

Заземлювальний штифт забезпечує кращий тепловий шлях до фільєри, так що температура фільєри буде близька до температури друкованого ланцюга. Необхідно дбати про те, щоб це було в хорошому тепловому контакті з вимірюваною поверхнею.

Електростатичні заряди до 4000 В легко накопичуються на тілі людини і випробовують обладнання та можуть розряджатися без виявлення. Незважаючи на те, що AD7814 має власну схему захисту ESD (електростатичні заряди), можуть викликати пошкодження пристроїв, які піддаються електростатичним розрядам високої енергії. Тому, щоб уникнути погіршення продуктивності або втрати функціональних можливостей, рекомендується вживати належних заходів безпеки.

Як і будь-які інші мікросхеми, AD7814 та пов'язані з ним електропроводки та схеми повинні бути захищеними від вологи, щоб запобігти витокам і корозії, особливо в холодних умовах, де конденсація більш ймовірна.

Для захисту можуть використовуватися водостійкі лаки та конформні покриття. Малий розмір пакету AD7814 дозволяє встановлювати його в герметичні металеві зонди, які забезпечують безпечне середовище для пристрою.

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ ПРОЕКТУ В СЕРЕДОВИЩІ ACTIVE-HDL

3.1 Інтерфейсний опис проекту

VHDL – це мова опису апаратних засобів, що використовується в автоматизації проектування для опису цифрових і змішаних сигналів, таких як програмовані логічні матриці та інтегральні схеми. VHDL також може використовуватися як мова паралельного програмування загального призначення [12].

Початкова версія VHDL, призначена для IEEE стандарту IEEE 1976-1987, включала в себе широкий спектр типів даних, в тому числі числового (цілого числа і дробового), логічного (бітового і булевого), часу, а також масиви.

Основна перевага VHDL, коли використовується для проектування систем, полягає в тому, що він дозволяє описати (змоделювати) поведінку необхідної системи і перевірити (симулювати) як інструменти синтезу переведуть конструкцію в реальне обладнання.

Іншою перевагою є те, що VHDL дозволяє описувати паралельну систему. VHDL є мовою потоку даних, на відміну від процедурних обчислювальних мов, таких як BASIC і C, який виконується послідовно, одна інструкція одночасно.

Проект VHDL є багатоцільовим. Створюваний один раз, блок обчислення може використовуватися в багатьох інших проектах. Однак можна налаштовувати багато параметрів формаційного та функціонального блоків (параметри ємності, розмір пам'яті, елементна база, структура блоків та структура з'єднань) [13].

Проект VHDL є портативним. Створений для однієї елементної бази, проект обчислювального пристрою може бути перенесений на іншу елементну базу, наприклад VLSI з різними технологіями.

Великою перевагою VHDL в порівнянні з оригінальним Verilog є те, що VHDL має повний тип системи. Дизайнери можуть використовувати систему

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

типу для написання набагато більш структурованого коду (особливо шляхом оголошення типів записів).

Active-HDL є інтегрованим рішенням для проектування та моделювання FPGA на Windows для командних середовищ. Інтегрований дизайн середовища Active-HDL (IDE) включає в себе повний набір інструментів HDL і графічного дизайну і симулятор змішаної мови на рівні RTL / gate для швидкого розгортання і перевірки конструкцій FPGA.

Вікно потоку менеджерів викликає більше 120 EDA і FPGA інструментів, під час проектування введення, моделювання, синтезу та реалізації потоків і дозволяє командам залишатися в рамках однієї спільної платформи протягом всього процесу розробки FPGA. Active-HDL підтримує провідні FPGA-пристрої від Altera, Atmel, Lattice, Microsemi (Actel), Quicklogic, Xilinx та багато інших [14].

Переваги:

- управління дизайном на основі уніфікованої команди підтримує однорідність між локальними або віддаленими командами;
- налаштовувані інтерфейси FPGA / EDA Flow Manager з інструментами більше 120 постачальників дозволяють командам залишатися на одній платформі протягом розробки FPGA;
- потужний спільний симулятор ядра змішаної мови, який підтримує VHDL, Verilog, SystemVerilog (Design) і SystemC;
- забезпечення якості та надійності коду за допомогою графічно інтерактивних інструментів налагодження та якості коду;
- виконання перевірки на основі метрик, щоб визначити невикористані частини коду, використовуючи інструменти аналізу коду;
- покращення якості перевірки та знаходження помилок за допомогою ABV - підтвердження на основі тверджень (SVA, PSL, OVA);
- анотація програмного інтелекту представляють їх у легко зрозумілій графічній формі з використанням HDL перетворювача;

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

– розробка конструкції за допомогою автоматичної генерації документації коду в HTML і PDF.

Active-HDL забезпечує потужну функцію, яка дозволяє здійснювати зв'язок і співпрацю з низкою провідних систем керування версіями джерел. Вбудований інтерфейс дозволяє працювати на наступних версіях і переглядах вихідних файлів дизайну безпосередньо з середовища Active-HDL. У такому середовищі можна відстежувати зміни, внесені в конструкцію, і переглядати відмінності між наступними версіями вихідних файлів.

Система управління вихідним переглядом також полегшує роботу команди, оскільки вона дозволяє групі людей працювати над одним проектом. Як тільки вихідні файли архівуються у сховищі, вони доступні іншим членам команди. Крім того, всі зміни, внесені до будь-якого файлу, зберігаються з повною історією, тому можна відновити будь-яку версію будь-якого файлу в будь-який час.

Active-HDL підтримує наступні методи стимулювання або форсування вхідних сигналів під час моделювання:

- вручну вибрані стимулятори з ресурсів Active-HDL;
- файли VHDL або Verilog TestBench, створені майстром TestBench;
- створення користувачем VHDL або Verilog TestBench файлів;
- файли TestBench з VHDL WAVES (згідно з специфікацією IEEE WAVES 1029.1);
- порівняння результатів Verilog TestBench;
- команди моделювання вводяться з вікна консолі;
- файли, що містять макрокоманди моделювання;
- введення моделювання на основі сигналів, що редагуються користувачем.

Користувач може вибрати метод, який найкраще підходить для конкретних проектних потреб і бути впевненими, що існує відповідність між часом, необхідним для створення вхідних даних моделювання та складністю перевірки конструкції [15].

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

Тип `std_logic` був введений відповідно до стандарту IEEE-1164 в якості точного представника одного проводу або біта. Сама мова VHDL не забезпечує однобітний тип, який є достатньо надійним для представлення "реальної" логіки. Тобто представляти всі можливі стани модельованої і синтезованої логіки в сучасному програмованому логічному пристрої.

На початку історії VHDL, різні розробники по суті створювали свої власні версії `std_logic` поза необхідністю представляти сигнали реального світу. IEEE-1164 представив цей стандартизований тип логіки з метою підвищення сумісності коду, написаного різними розробниками для різних архітектур.

`Std_logic` визначається як підтип `std_ulogic`, тому всі оператори і функції, визначені для `std_ulogic`, можуть бути застосовані до `std_logic`. Тип `std_ulogic` є розширенням стандартного типу `Bit`. Він визначає дев'ять значень, які дозволяють визначити логічні системи [16]. Сигнал або змінна цього типу можуть приймати наступні значення:

- 'U' – неініціалізований, цей сигнал ще не встановлено;
- 'X' – невідомий, неможливо визначити це значення чи результат;
- '0' – логіка 0;
- '1' – логіка 1;
- 'Z' – високий опір;
- 'W' – слабкий сигнал, не може визначити, чи має він бути 0 або 1;
- 'L' – слабкий сигнал, який, ймовірно, повинен перейти до 0;
- 'H' – слабкий сигнал, який, ймовірно, повинен перейти до 1.

Основні логічні операції VHDL визначені на цьому типі: `and`, `nand`, `or`, `nor`, `xor`, `xnor`, `not`. Вони можуть використовуватися як вбудовані операції на бітах.

Для того, щоб полегшити специфікацію багатовимірних сигналів (наприклад, шини даних), пакет `std_logic_1164` визначає функцію дозволу для `std_ulogic`, яка в свою чергу служить основою для декларування типу `std_logic` [17].

Пакет `std_logic_1164` визначає перевантажені логічні оператори ("`i`", "`nand`", "`або`", "`nor`", "`xor`" і "`not`") для операндів типу `std_ulogic`. Крім того,

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

визначено дві функції перетворення: `std_ulogic` до `Bit` (функція `To_Bit`) і `Bit` до `std_ulogic` (функція `To_StdULogic`) [18].

Для реалізації проекту використані наступні бібліотеки `ieee.std_logic_1164.all`, `ieee.std_logic_arith.all`, `ieee.std_logic_unsigned.all`, які дозволяють працювати з логічними даними, здійснювати арифметичні операції та вводити/виводити дані. Ці бібліотеки є стандартними в середовищі `Active-HDL` [19].

Вхідними змінними проекту є:

- `sysclk` – системний час, який дозволяє здійснювати симуляцію проекту та в подальшому реалізовувати даний проект на ПЛІС чи ПЛМ;
- `read_temp` – тригер, який зчитує дані з температурного сенсора;
- `temp_data` – значення температури, яке подають на компоненту `mux`;
- `temp_din` – поточне значення входу сенсора;
- `temp_dout` – поточне значення виходу сенсора;
- `temp_sclk` – поточне значення часу сенсора.

В загальному інтерфейсний опис проекту має вигляд:

```
entity TempSensor is
  Port ( SYSCLK      : in  std_logic;
        READ_TEMP   : in  std_logic;
        TEMP_DATA    : out std_logic_vector(9 downto 0);
        TEMP_DIN     : out std_logic;
        TEMP_DOUT    : in  std_logic;
        TEMP_SCLK    : out std_logic);
end TempSensor;
```

3.2 Моделювання сенсора температури в середовищі `Active-HDL`

Для зручної роботи з проектами `Active-HDL` має доступний інтерфейс, дерево проекту та зрозумілий консоль (рисунк 3.1).

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		


```

SerClkProc : process (SYSCLK)
  Begin if (rising_edge(SYSCLK))
    then
      if (SerClkEn = '1')
        then
          if (SerClkCntr < SERCLKRATE) then
            SerClkCntr <= SerClkCntr + 1;
          else SerClkCntr <= (others => '0');
            SerClk <= not SerClk;
          end if;
        else SerClkCntr <= (others => '1');
          Serclk <= '1';
        end if;
      end if;
    end process;

```

Компонента SerReadProc має наступний опис:

```

SerReadProc : process(SerClk, SerClkEn)
  begin
    if (SerClkEn = '1')
      then
        if (rising_edge(SerClk))
          then
            SerDataIn <= SerDataIn(14 downto 0) & TEMP_DOUT;
          end if;
        else
          DataInLatch <= SerDataIn(14 downto 5);
        end if;
      end process;

```

Компонента SerWrtProc має наступний опис:

```

SerWrtProc : process(SerClk, SerClkEn, SerPwrDown)
  begin
    if (SerClkEn = '0')
      then
        SerDataOut <= "00" & SerPwrDown & "00000000000000";
      elsif (falling_edge(SerClk))
        then
          TEMP_DIN <= SerDataOut(15);
          SerDataOut <= SerDataOut(14 downto 0) & '0';
        end
      if;
    end process;

```

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

Вказані компоненти описані в проекті як процеси для зручності оформлення. Варто зазначити, що в моделі використано додаткові сигнали:

- constant SERCLKRATE = "010" – період тактової частоти в даний час встановлений на 3 періоди clk, тобто 75ns на фазу;
- constant SERFRAME = "10000" – кількість бітів в послідовному кадрі;
- signal SerClk – фактичний синхронізуючий сигнал;
- signal SerClkEn – запуск серійного годинника;
- signal SerDataIn – послідовне зчитування сигналів;
- signal SerDataOut – паралельне зчитування сигналів;
- signal DataInLatch – запис значення SerDataIn для подальшого використання;
- constant TEMPMEAS = "1111101000" – опис 1000 тактових циклів;
- signal TempTimer – лічильник таймера очікування відповіді датчика;
- signal SerPwrDown – вказівник підключення датчика.

Загальна схема датчика температури подана у графічній частині БР.КСМ.07129/15.00.00.000 С1.

3.3 Симуляція та верифікація проекту

Редактор часових діаграм – це графічний редактор, призначений для перегляду і редагування часових діаграм. Зокрема, у цьому редакторі можна створювати форми сигналів, що подаються на входи схем при моделюванні, а також переглядати результати моделювання.

При редагуванні часових діаграм можна застосовувати переміщення, копіювання, склеювання, тощо. Редактор часових діаграм дозволяє також порівняти між собою два сигнали і виділити кольором їх відмінності.

Редактор часових діаграм є основним засобом для створення стимулюючих сигналів і перегляду результатів моделювання. Файли часових діаграм мають

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

імена з розширенням *.awf і зберігаються у папці Waveforms. Щоб відкрити існуючий файл часової діаграми можна клацнути мишкою на закладці файла у вікні редакторів, або двічі клацнути на імені файла у вікні Design Browser, або вибрати у контекстному меню команду open. Для створення нового файла часової діаграми можна скористатися кнопкою New Waveform.

В лівій частині вікна редактора часових діаграм у колонках Name, Value, Stimulator відображаються відповідно імена сигналів, їх значення у момент, що відповідає положенню курсора, а також стимулятори, які застосовуються для формування вхідних сигналів. Натискаючи на кнопки з заголовками цих трьох колонок, можна розташувати сигнали за алфавітним порядком їх імен чи стимуляторів, у порядку зростання значень або у порядку протилежному вказаним. Якщо сигнали не впорядковані за жодною з цих ознак, то їх розташування можна обирати довільно, перетягуючи сигнали за допомогою мишки [20].

Масштаб зображення регулюється кнопками Zoom In (Num +), Zoom Out (Num –), Display Selection Area (Home), Zoom To Fit (End), які дозволяють відповідно збільшити масштаб вдвічі; зменшити масштаб вдвічі; вивести усю діаграму на один екран; задати у вікні діалогу часові межі зображення на екрані.

Кнопка Zoom Mode включає режим Zoom, у якому можна, виділивши мишкою частину зображення, показати його на весь екран. Щоб збільшити масштаб вдвічі, тут достатньо просто клацнувши мишкою по діаграмі, а для зменшення – клацнути мишкою при натиснутій клавіші Ctrl. Крім того, в будь-якому режимі зручно збільшувати і зменшувати масштаб за допомогою клавіш "+" і "-" на додатковій клавіатурі.

Векторні сигнали справа від імені мають кнопку, що дозволяє переключатись між згорнутим (єдина діаграма для всього сигналу) і розгорнутим (окрема діаграма для кож-ного біту) способом їх відображення. Опція Properties з контекстного меню сигналу дозволяє також обрати його колір, систему числення, в якій вказується значення вектор-ного сигналу, тощо.

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

Щоб розмістити сигнал на часовій діаграмі, треба відкрити закладку Structure у вікні Design Browser, виділити в структурі проекту елемент, сигнали якого ви хотіли б відобразити на часовій діаграмі та методом drag-&-drop перетягти потрібні сигнали з нижньої частини закладки Structure у ліву частину вікна Редактора часових діаграм.

Інший спосіб розміщення сигналу на часовій діаграмі – це використання опції Add Signals з контекстного меню лівої частини вікна Редактора часових діаграм. Користуючись ним, треба виділити потрібні сигнали і натиснути кнопку Add.

Сигнали можна також додавати за допомогою команди Wave <ім'я_сигналу>, що вводиться з клавіатури у вікно Console.

Щоб вилучити сигнал зі списку сигналів часової діаграми треба, встановивши на ньому курсор, натиснути клавішу Delete чи одноіменну опцію у контекстному меню.

Будь-якому сигналові, присутньому на часовій діаграмі, можна надати бажаних значень, скориставшись вбудованими у програму стимуляторами. Для цього треба встановити курсор на потрібному сигналі і вибрати опцію Stimulators у контекстному меню або одноіменну кнопку на панелі інструментів. Вікно, що відкривається при цьому, дає можливість вибору таких стимуляторів:

- value – дозволяє вибрати значення, що буде надане виділеному сигналу чи групі сигналів, об'єднаних у шину;
- formula – дозволяє задати часову діаграму за допомогою формули;
- hotkey – дозволяє призначити сигналові гарячу клавішу, натискання якої призводитиме до зміни значення сигналу;
- clock – дозволяє задати періодичний сигнал з визначеними параметрами (частотою, періодом тощо), користуючись зручним графічним редактором;
- counter – дозволяє моделювати сигнали на виходах лічильника, з заданими значеннями періоду, напрямку лічби і кодування даних у лічильнику: двійковий код, код Грея та ін.;

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

– custom – дозволяє використати стимул користувача, створений у редакторі часових діаграм;

– predefined – дозволяє вибрати наперед визначений стимулюючий сигнал з заданого переліку та поповнити цей перелік власними сигналами, що часто використовуються.

Часові діаграми можна редагувати та зберігати у файлах для подальшого використання. Щоб приступити до редагування діаграми, слід припинити моделювання командою End Simulation з меню Simulation та натиснути кнопку , що переводить редактор у режим редагування.

Зміна моменту переключення (фронту) сигналу виконується аналогічно регулюванню ширини колонок у таблицях Word чи Excel. Для цього треба підвести курсор до зображення фронту сигналу (вертикального відрізка на часовій діаграмі) так, щоб курсор набув вигляду двонаправленої стрілки, потім натиснути ліву клавішу мишки і відпустити її там, де має проходити фронт.

Аналогічним чином, підвівши курсор до горизонтального відрізка діаграми, можна підняти його вгору, або опустити вниз, змінивши тим самим значення сигналу 0 на 1 чи навпаки.

Для створення нового імпульсу, треба натиснути клавішу Shift (при цьому курсор набуде форми олівця) та клацнути мишкою по часовій діаграмі. В результаті на цьому місці буде створено імпульс невеликої ширини. Потім описаним вище способом можна відредагувати ширину та положення імпульсу на часовій осі.

Щоб виділити фрагмент часової діаграми, треба натиснути клавішу Ctrl (у результаті курсор набуде форми хрестика) натиснути ліву клавішу мишки на початку і відпустити її у кінці фрагмента, що виділяється. Фрагмент може включати як одну, так і декілька сусідніх часових діаграм.

Виділений фрагмент можна:

- вилучити натисканням клавіші;
- встановити у потрібне значення;

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

- змінити масштаб часу, вибравши у контекстному меню команду Stretch і вказавши у вікні Scale процент "розтягування" даного фрагмента у часі;
- скопіювати у буфер обміну натисканням кнопки або комбінації клавіш Ctrl+C;
- вирізати натисканням кнопки або комбінації клавіш Ctrl+X.

Запис результатів редагування часової діаграми у файл виконується за допомогою команд Save та Save as з меню File.

Для здійснення симуляції проєкту спочатку задаються вхідні стимули відповідно до типів вхідних змінних. Часова діаграма спроектованого сенсора температури подана на рисунку 3.2.

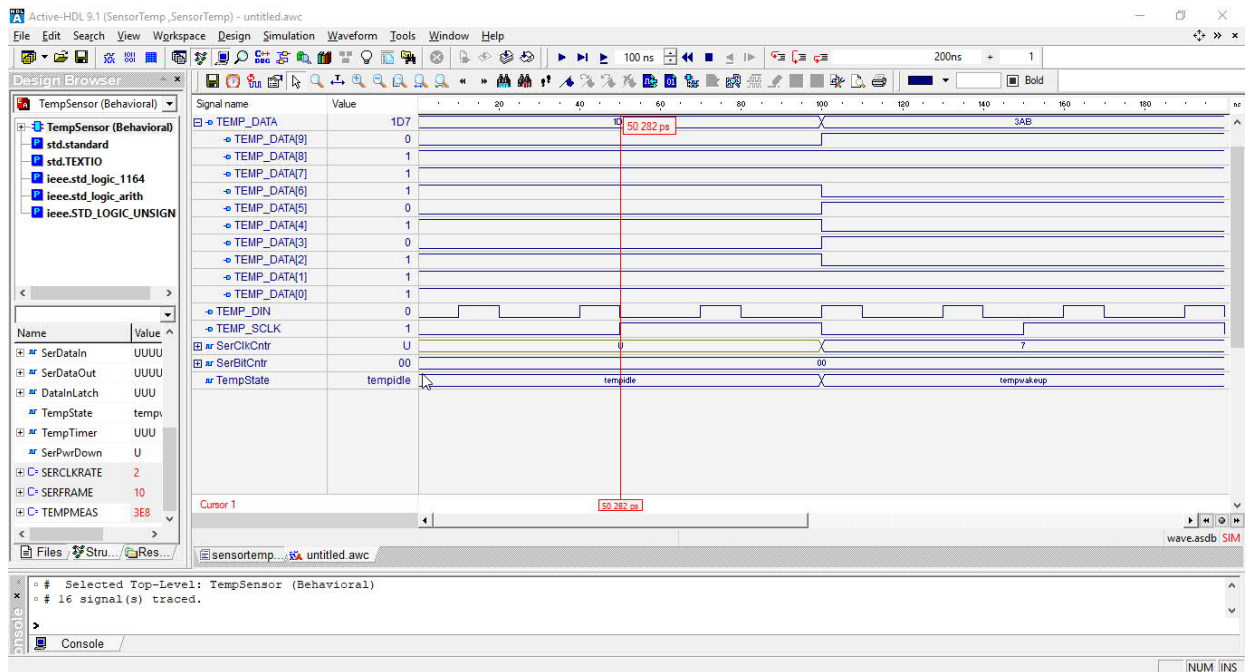


Рисунок 3.2 – Часова діаграма роботи HDL-моделі сенсора температури

Аналіз часової діаграми підтверджує її працездатність та правильність результатів що дозволяє застосовувати розроблений HDL-опис для програмування сучасних ПЛІС.

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ПРОГРАМНОГО ЗАСОБУ

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

Розроблене програмне забезпечення призначене для моделювання роботи сенсора температури AD7814 і характеризується підвищеною ефективністю виконання алгоритму, що призводить до зменшення часу візуального представлення об'єкту дослідження.

4.1 Розрахунок витрат на розробку програмного забезпечення

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

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

де K_1 – витрати на розробку програмних засобів, грн;

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

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробників ($B_{оп}$);
- витрати на відрахування у спеціальні державні фонди ($B_{ф}$);
- витрати на покупні вироби ($Пв$);

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

- витрати на придбання спецобладнання для проведення експериментальних робіт (*Об*);
- накладні витрати (*Н*);
- інші витрати (*Ів*).

4.1.1 Розрахунок витрат на оплату праці

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

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

Таблиця 4.1 – Вихідні дані для розрахунку витрат на оплату праці

Посада виконавців	Місячний оклад, грн
Керівник ВКР, доцент	6397
Консультант техніко-економічного розділу, доцент	7293
Студент	1400

Витрати на оплату праці розробників проекту визначаються за формулою:

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

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

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

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

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

Середньо годинна ставка працівника може бути розрахована за формулою:

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

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

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

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

Результати розрахунку записують до таблиці 4.2.

Таблиця 4.2 – Розрахунок витрат на оплату праці

Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год	Витрати на розробку, грн
Керівник ВКР, доцент	16	49,5	792
Консультант техніко-економічного розділу, доцент	2	63,8	127,6
Студент	100	8,3	830
Разом		1749	

4.1.2 Відрахування на соціальні заходи

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

$$B_{\phi} = \frac{20,5}{100} \cdot 1749 = 358,5 \text{ (грн)} \quad (4.4)$$

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

4.1.3 Розрахунок витрат на матеріали та комплектуючі

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

Таблиця 4.3 – Розрахунок витрат на матеріали та комплектуючі

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

4.1.4 Витрати на використання комп'ютерної техніки

Витрати на використання комп'ютерної техніки включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним забезпеченням, витрати на електроенергію, що споживається комп'ютером. За даними обчислювального центру ТНЕУ для комп'ютера типу ІВМ РС/АТХ вартість години роботи становить 5,2 грн. Середній щоденний час роботи на комп'ютері – 2 години. Розрахунок витрат на використання комп'ютерної техніки приведений в таблиці 4.4.

Таблиця 4.4 – Розрахунок витрат на використання комп'ютерної техніки

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

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

4.1.5 Накладні витрати

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

$$H = 1,5 \cdot 1749 = 2623,5 \text{ (грн)} \quad (4.5)$$

4.1.6 Інші витрати

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

$$I = 1749 \cdot 0,1 = 174,9 \text{ (грн)} \quad (4.6)$$

Витрати на розробку програмного забезпечення складають:

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

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

$$K_1 = 1749 + 358,5 + 187 + 2623,5 + 174,9 = 5092,9 \text{ (грн)} \quad (4.8)$$

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

$$K_2 = S_{м.г.} \cdot t_{від} \quad (4.9)$$

де $S_{м.г.}$ – вартість однієї машино-години роботи ПК, грн/год.

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

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

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

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

Таблиця 4.5 – Кошторис витрат на розробку програмного забезпечення

Найменування витрат	Сума витрат, грн
Витрати на оплату праці	1749
Відрахування у спеціальні державні фонди	358,5
Витрати на куповані вироби	187
Накладні витрати	2623,5
Інші витрати	174,9
Витрати на відлагодження і дослідну експлуатацію програмного продукту	416
Разом	5508,9

4.2 Визначення експлуатаційних витрат

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

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

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

де E_{Π} – одноразові експлуатаційні витрати на ПЗ (аналог), грн;

$E_{1\Pi}$ – вартість підготовки даних для експлуатації ПЗ (аналог), грн;

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

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

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

де N_{Π} – періодичність експлуатації ПЗ (аналог), раз/рік.

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

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

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

n_i – кількість працівників i -ої категорії, осіб;

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

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

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

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

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

b – коефіцієнт, який враховує додаткову заробітну плату;

m – кількість робочих годин у місяці, год.

Для роботи з даними як для поточного програмного забезпечення так і аналогу потрібен один працівник, основна місячна заробітна плата якого складає: $c^0 = 1400$ грн. Тоді:

$$c_1 = \frac{1400(1+0,57)}{22*8} = 12,5 \text{ (грн/год)} \quad (4.15)$$

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

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

Час роботи співробітників, год	Середньогодинна заробітна плата, грн/год	Витрати, грн
Проектне рішення		
1	12,5	12,5
Аналог		
1,5	12,5	18,75

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2П} = t * S_{МГ} \quad (4.16)$$

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

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

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

$$E_n = 12,5 + 5,2 = 17,7 \text{ (грн)}; E_a = 18,75 + 7,8 = 26,55 \text{ (грн)} \quad (4.18)$$

$$B_{en} = 17,7 \cdot 252 = 4460,4 \text{ (грн)}; B_{ea} = 26,55 \cdot 252 = 6690,6 \text{ (грн)} \quad (4.19)$$

4.3 Розрахунок ціни споживання програмного продукту

Ціна споживання – це витрати на придбання і експлуатацію го продукту за весь строк його служби:

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

де $Ц_{П}$ – ціна придбання програмного продукту, грн.

$$Ц_{П} = K \left(1 + \frac{П_p}{100}\right) + K_0 + K_k \quad (4.21)$$

де K – кошторисна вартість;

$П_p$ – рентабельність;

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

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

K_k – витрати на доукомплектування технічних засобів на об'єкті, грн.

$$Ц_d = 55089 \cdot (1 + 0,3) = 716157 \text{ (грн)}. \quad (4.22)$$

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

$$B_{\text{епрв}} = \sum_{t=0}^T \frac{B_{\text{еп}}}{(1 + R)^t} \quad (4.23)$$

де $B_{\text{еп}}$ – річні експлуатаційні витрати, грн;

T – термін служби програмного забезпечення, років;

R – річна ставка проценту банку.

$$B_{\text{епрв}} = \sum_{t=1}^5 \frac{4460,4}{(1 + 0,08)^t} = 17845,3 \text{ (грн)} \quad (4.24)$$

$$B_{\text{епрв}} = \sum_{t=1}^5 \frac{6690,6}{(1 + 0,08)^t} = 26767,9 \text{ (грн)} \quad (4.25)$$

Тоді ціна споживання програмного забезпечення дорівнюватиме:

$$Ц_{\text{сп}} = 7161,57 + 17845,3 = 25006,9 \text{ (грн)} \quad (4.26)$$

Аналогічно визначається ціна споживання для аналогу:

$$Ц_{\text{са}} = 6500,0 + 26767,9 = 33267,9 \text{ (грн)} \quad (4.27)$$

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

4.4 Визначення показників економічної ефективності

Економічний ефект в сфері розробки програмного продукту:

$$E_{PP} = C_{II} - C_A \quad (4.28)$$

$$E_{PP} = 7161,57 - 6500,0 = 661,57 \text{ (грн)} \quad (4.29)$$

Річний економічний ефект в сфері експлуатації:

$$E_{KC} = B_{EA} - B_{EP} \quad (4.30)$$

$$E_{KC} = 6690,6 - 4460,4 = 2229,6 \text{ (грн)} \quad (4.31)$$

Додатковий економічний ефект у сфері експлуатації:

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

$$\Delta E_{екс} = \sum_{t=1}^5 2229,6 * (1 + 0,08)^{5-t} = 13065,41 \text{ (грн)} \quad (4.33)$$

Сумарний ефект складає:

$$E = E_{pp} + \Delta E_{екс} = 661,57 + 13065,41 = 13726,98 \text{ (грн)} \quad (4.34)$$

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.7 – Показники економічної ефективності програмного забезпечення

Найменування	Одиниці вимірювання	Значення показників	
		Базовий варіант	Новий варіант
Капітальні вкладення	грн.	–	5508,9
Ціна придбання	грн.	6500,0	7161,57
Річні експлуатаційні витрати	грн.	17200,15	26767,9
Ціна споживання	грн.	24721,35	33267,9
Економічний ефект в сфері проектування	грн.	–	661,57
Економічний ефект в сфері експлуатації	грн.	–	2229,6
Додатковий ефект в сфері експлуатації	грн.	–	13065,41
Сумарний ефект	грн.	13726,98	

4.5 Висновки

В даному розділі проведено розрахунок витрат на розробку програмного забезпечення. Здійснено порівняння з існуючим аналогом, і цим показано, що вказане програмне забезпечення має переваги в порівнянні з аналогами, зокрема: надійність, простота використання, зручність. Згідно проведеного економічного обґрунтування зазначене програмне забезпечення є конкурентоздатним. Крім того, отримано економічний ефект у розмірі 13726,98 грн. і тому розробка і впровадження цього програмного забезпечення є економічно доцільними.

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

ВИСНОВКИ

В процесі виконання бакалаврської роботи здійснено:

- 1) аналіз сучасного стану застосування сенсорів температури, що дозволило виділити основні їх характеристики;
- 2) сформовано вимоги до проекту, що є основою для розробки HDL-моделі сенсора температури;
- 3) розроблено етапи проектування та вибрано сучасне середовище проектування засобів комп'ютерних систем Active-HDL;
- 4) проаналізовано алгоритм роботи та виділено основні складові сенсора температури AD7814, що дозволило розробити основні компоненти HDL-моделі сенсора;
- 5) здійснено аналіз вхідних та вихідних змінних сенсора температури, що дозволило розробити інтерфейсний опис проекту;
- 6) здійснено моделювання та симуляцію проекту в середовищі Active-HDL, аналіз якої підтвердив працездатність та правильність результатів роботи проекту;
- 7) проведено верифікацію проекту, в результаті якої сформовано технічну документацію, що може використовуватися для подальшої реалізації проекту на ПЛІС чи ПЛІМ;
- 8) здійснено техніко-економічне обґрунтування проекту.
- 9) Розроблена HDL-модель сенсора температури має практичне значення, що підтверджено довідкою про впровадження (додаток Б).

Результати дипломного проектування доповідалися на інтернет-конференції «Науково-практична конференція інтелектуальні комп'ютерні системи та мережі» (додаток В).

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

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

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

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

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

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

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

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

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

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

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

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

					БР. КСМ.07129/15.00.00.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		